

# On the Need for Assistance in HTML5 Web Authoring Systems

Julia C. Lee and Lawrence J. Henschen<sup>(✉)</sup>

Northwestern University, Evanston, IL 60208, USA

Julialee@agep.northwestern.edu, henschen@eecs.northwestern.edu

**Abstract.** HTML5 incorporates semantics, including among others the purpose and intention of the web author, as an integral part of the language and specification. The goal is to allow more sophisticated browsers to render the content in ways that are appropriate for both the platform and the abilities of the user, thereby achieving universal access. However, achieving that goal depends on web authors using the elements and attributes correctly. We illustrate why this will be difficult for most web authors. We propose that web editors be enhanced to provide guidance to web authors in the correct and proper usage of the HTML5 features and give some examples of how this might work.

**Keywords:** HTML5 · WAI-ARIA · Semantic web · Universal access

## 1 Introduction

HTML5 became an official recommendation on Oct. 28 2014 [1]. HTML5 inherits and improves on many features of the previous versions of HTML but also embraces many new ideas and philosophies. Among the new features that HTML5 offers are: homogenize HTML and XHTML into one specification; make DOM the “official” in-memory model of an HTML document; and hide the complexities of multithreading from web authors. But the feature that is perhaps the most important one for the HCI community is the giant step towards the separation of formatting and presentation from the content and semantics.

To clearly convey ones thoughts in a presentation and to correctly understand the thoughts conveyed by presentations from others are critical issues for successful communication between parties. As the internet becomes an increasingly indivisible part of our daily life, HTML becomes one of the most important languages used for communication among humans. Much research in the information technology field has been devoted to understanding the semantic meaning of human presentations and to providing tools that facilitate the correct presentation of peoples’ meaning. The Semantic Web and universal access are hot topics in the Web development/research area [2, 3, 8, 9]. HTML5 is a major step forward in supporting these trends. In particular, the separation of presentation from content and semantics, the introduction of new elements that explicitly give the intended purpose of the content, and the use of WAI-ARIA roles [5] greatly improves the ability to present content on arbitrary platforms and to users with arbitrary

abilities and needs in ways that facilitate human understanding of the content, that is, that greatly improves the chances for universal access to web content.

However, the degree to which HTML5 can lead to increased access depends heavily on web authors using the language properly. Languages, such as HTML5, are the tools used to present human thoughts. Good tools need users that have good understanding about the tools in order to fully utilize the good attributes that are built in. Many of the wonderful new features of HTML5 are difficult for web authors, particularly non-professional web authors who may produce the majority of web pages, to understand and use properly. We suggest in this article that, in order to achieve the promise of universal access inherent in the philosophy and design of HTML5, intelligent web authoring tools will be needed. We describe some of the subtleties of HTML5 and then suggest several ways in which web authoring tools and HTML5 editors can aid authors in properly using the language.

## 2 Related Work on the Semantic Web

W3C has contributed much in this area by developing a number of recommendations related to the semantic web and universal access. W3C published a specification of RDF (The **Resource Description Framework**) and its data model and an XML serialization as a recommendation in 1999 [4]. This is one of the earliest efforts focused on “semantic web”, and efforts by W3C and others have continued vigorously since then. There is now a suite of standards, recommendations, and tools related to RDF initiatives. One of the most important contributions by W3C is the recent WAI-ARIA specification [5].

“This specification provides an ontology of roles, states, and properties that define accessible user interface elements and can be used to improve the accessibility and interoperability of web content and applications”.

The role taxonomy defined by this specification makes direct connection between semantic meaning and HTML mark-up elements, as described in [6]. The role taxonomy is primarily defined for universal accessibility aimed for people with disabilities. HTML5 incorporated the role attribute into its element definition. The role taxonomy includes specific roles, such as “button”, “banner”, “checkbox”, etc., and abstract roles, such as “composite”, “command”, “range”, etc., that can express semantics of elements that are not directly represented by the element itself.

## 3 Semantic Aspects of HTML5

In this section we illustrate by example how HTML5 incorporates semantics. We point out in advance that web authors could express their content without using these HTML5 features, but this would make it difficult or impossible for renderers to understand the author’s meaning and thereby be able to render the content well on different platforms and to different kinds of users. We encourage the reader to think to him- or herself how much easier it is to understand content when the appropriate element or attribute is used.

### 3.1 New Elements with Semantic Meaning

HTML5 introduced many new elements that convey certain semantic meanings by the name itself – for example: video, address, legend, aside, blockquote, small, etc. The need and use of some of these is obvious; modern web pages often include video, and it is natural to have a special element for that purpose. The need for other elements, especially elements whose content is mainly text, is less obvious until one considers the rendering of text to, for example, a blind user or an application. A block of text in a document might be difficult to recognize as being address information or a side comment not part of the main subject, even if additional intelligent technology is applied to that text. And yet users with different abilities will need to have those texts presented in different ways, and applications would need signposts to identify the portions of the page they want to use. With the meaningful tag/element names an application program or a screen reader for blind people will know immediately the semantics of the text block. Here are some simple examples.

“data” elements mark up blocks of text made of machine readable data values. Application programs may be able to read the text directly, but blind users will likely need the content present in a special way.

“main” elements point out the main contents related to the central topic. Blind readers can be informed that this text contributes to the main idea of the page, and sighted readers will benefit by knowing the author’s intent for this portion of the page.

“blockquote” elements contain quoted material. Using the “blockquote” element allows the renderer to distinguish that content from a simple quoted word, such as at the beginning of these points. It also alerts the renderer to look for a “cite” element that contains the citation for the quoted material. Without the “cite” element it would be impossible in general for a renderer to find the citation information.

These are very much semantic markups rather than structural markups. Only the author of the document can correctly say in all cases which parts of the text are which; document processing systems and artificial intelligence have not yet evolved to a level that could match the author’s knowledge. If used correctly, these new semantic HTML5 elements will enhance the understanding of the semantics of documents and improve accessibility to those documents.

### 3.2 Element Attributes

Using attributes to note special issues, including semantic issues, for element usage is not unique to HTML5; attributes have been a feature of HTML from the beginning. However, prior to HTML5 there were no official guidelines for how to use attributes to specify semantics relating to the element in which it was used. For example, an author could assign a value to the “name” attribute of a “div” element, but there was nothing in earlier HTML versions indicating that this attribute should specify the element’s role in the context of the document.

However, unlike previous versions of HTML, HTML5 has placed great emphases on using element attributes correctly and intelligently. The “title” attribute was a required element in every HTML5 document as a child element of the “head” element.

Now it can also be an attribute of other elements such as “abbr”, “dfn”, and others. The HTML5 specification [1] specifically tells what kind of “advisory” information this attribute should provide for each element. For example, in an “abbr” element the “title” attribute should be used to provide an expansion of the abbreviation being defined. For the “dfn” element the value of the “title” attribute is the term being defined. As another example, consider the new keyword “cite”. When used as an element it should contain the title of the reference, but when used as an attribute in a “blockquote” or “q” element it must be a valid URL potentially surrounded by spaces. Authors writing HTML5 documents should not use attributes arbitrarily but rather use them for exactly those purposes prescribed in the HTML5 specification.

More importantly, HTML5 allows ARIA roles, states, and properties [5] as attributes on any HTML5 element. The role attribute allows the author to annotate markup languages with machine-extractable semantic information about the purpose of an element. The presence of the “role”, or purpose, information can lead to increased accessibility, as described in [5] or in other literature on universal access (for example, [10]), as well as better presentation on different platforms, increased processing flexibility for applications, etc. WAI-ARIA [5] defines a specific role taxonomy for web elements focusing on the widget elements. This specification also defines the supported states, representing more dynamic aspects of a role, and properties, representing more stable aspects of a role, for the roles in the taxonomy. Figure 1, taken from [7], gives an example of how this might be used. The ‘aria-flowto’ attribute allows the author to specify alternative navigation through the document that might be useful for, say, a blind user. The “aria-flowto” overrides the default ordering based on the occurrence of the text in the document and also serves to group related items that may be physically separated in the text by other elements.

```

...
                                <html>
<div role="main" title="Top News Stories" id="main" aria-
flowto="stock"></div>
<div role="complementary" title="television listings" id="tv"></div>
<div role="complementary" title="stock quotes" id="stock" aria-
flowto="messages"></div>
<div role="complementary" title="messages from friends" id="messages" aria-
flowto="tv"></div>

```

Fig. 1. Example of ARIA attributes

### 3.3 Subtleties in HTML5 Semantics

The HTML5 specification emphasizes “Elements, attributes, and attribute values in HTML are defined (by this specification) to have certain meanings (semantics)” [1]. The semantics of the elements allows different browsers, screen readers, search engines,

etc. to present and interpret the HTML document closely to the intention of the author. Unfortunately, there are many subtleties in the semantics of the various elements and attributes that are difficult for even a professional web developer to understand, much less a novice. Therefore, the goal of HTML5 to make the web more accessible will be difficult to achieve without tools that aid web authors in correctly using the HTML5 language. We illustrate the difficulty with a brief discussion of two sets of HTML5 elements - a set of section/grouping elements and a set of text-level elements.

The HTML5 specification includes several elements that group larger portions of content. These include, among others, “main”, “article”, and “section”. The specification gives brief descriptions of the semantics and proper usage of each of these. For example, an “article” is a portion of content that could stand on its own or be extracted from the web page and still be meaningful. But, how is that different from “main”; and often times a “section” in a major work has the same property. The situation is made more confusing by the fact that in many (perhaps even most) browsers the text content of these three elements will appear in identical font and style. Even experienced web authors will need guidance in deciding which of these to use at any given point in an HTML5 document. Another example in this category is the “aside” element, whose content should be related to but not part of the main subject of the content surrounding it [1]. Web authors may think this can be used for parenthetical remarks in the content itself because in ordinary writing tangentially related comments are often written as parenthetical text. However, the HTML5 specification explicitly states that “aside” is not to be used for that purpose. And to further confuse the issue, the text-level element “small” is to be used for side comments.

Several text-level elements will also lead to confusion. For example, “em”, “strong”, and “b” all indicate some kind of stress, but the HTML5 specification gives slightly different semantics to these.

- The em element represents stress emphasis of its contents.
- The strong element represents strong importance, seriousness, or urgency.
- The “b” element represents a span of text to which attention is being drawn.

It is difficult to fully understand the differences or to appreciate how the content would be presented differently to impaired users. Moreover, web authors experienced in using earlier versions of HTML would be tempted to use the “span” element with suitable attributes for most of these purposes, but this would not be in the spirit of HTML5. In addition to helping impaired users, proper usage of elements such as these will facilitate the development of more sophisticated applications, such as search engines and data miners.

## **4 How HTML5 Editors Can Improve the Quality and Universal Accessibility of Web Pages**

To overcome the difficulties mentioned in the preceding section, particularly Sect. 3.3, we propose that HTML5 editors provide help for web authors. It is not enough for editors to merely check for syntactical correctness or compliance or even to provide syntactically-oriented auto-correct features like automatically closing elements. Authors need

help in understanding the subtle semantic differences and connections between closely related elements and attributes so that the resulting web pages accurately reflect the HTML5 semantics and the author's intent. We present examples of how editors could give such help at several different levels of sophistication.

#### 4.1 Simple Guidance and Crosschecking

At the simplest level editors can provide guidance in the use of the various elements and attributes. A simple scheme would be to alert an author to alternate elements or attributes when the author used one of the HTML5 items. For example, the editor could pop up a window explaining the difference between “main”, “article”, “section”, “span”, etc. whenever a user selected one of those (either by typing it explicitly in a primitive editor or selecting it from a menu in one of the more sophisticated web authoring environments). Similarly for when the user types or selects one of “em”, “strong”, “u”, “i”, etc. The help might be several levels deep. For example, the first pop-up help window could remind the author of the differences and link to a second level help window with simple examples. Depending on the particular element or attribute group involved, there could be deeper levels explaining in great detail the differences and how those differences would impact impaired users.

Many of the semantic aspects of HTML5 elements have a closely related syntactical aspect that can easily be checked. For example, the presence of a “blockquote” should indicate the need for a “cite” element, and “li” elements should appear inside a list group element (“ol” or “ul”). HTML5 web authoring systems can easily check whether or not the document satisfies these and can suggest to the author how to fix errors.

As authors become better at writing semantically correct HTML5 documents, the need for such proactive help would reduce. Therefore, systems should allow users to specify the level of help and guidance and whether to present it during document preparation or only at certain times (for example, on request or at document save).

#### 4.2 Templates

A somewhat more sophisticated and proactive approach is for the web authoring system to know about templates for various kinds of web documents and to provide templates based on the kind of web page the author is developing. In this kind of system the user might be asked at the beginning to select a web page type from a drop-down menu or through a more complex dialogue. For example, the user might select “business” or “newsletter” or “scientific-article” from a list of basic web-page types. The system might then prompt for additional information, for example the kind of business page (simple advertisement, shopping, information only, etc.) Once the author has specified the type, the system proposes a structure and gives help and guidance aimed specifically at that type of web page. In some cases the choice of HTML5 elements and attributes will be quite clear, while in others there may be much more flexibility based on what the web author intends. We illustrate with two examples.

If the author indicates this is a moderate to extensive news page, the overall structure would be quite well defined. The bulk of the content would be organized as “article” elements, and there should be “head”, “title”, and possibly “link” and “meta” elements

for the document as a whole. In this situation the system would simply advise the author about the need for each of these and perhaps provide empty elements for the author to simply fill in. There would still be some issues involving choice. For example, while print news (e.g., printed newspapers) rarely have sections, web-based news may well benefit by the extra organization the “section” element provides. An authoring system should explain the benefits to the author but let the author make the final choice.

Other contexts would allow more flexibility, and the system would be more advisory in nature. For example, if the type was shopping page for a business, then the system might propose the author use either header elements (e.g., “h1”, “h2”, ...) or “article” for major areas of the web site (company overview, contact information, billing, shipping, product lines). Each of these choices has consequences in terms of presentation on different platforms and to different users as well as for applications that process the web page. Similarly, individual products might be presented in any of “p”, “ol”, “ul”, or “div” elements, optionally with “hr” elements. Again, each of these will have consequences for both presentation and processing by applications. The system should suggest these alternatives and explain the consequences so that the author not only achieves his or her purpose but also does it in a way that matches the semantics of HTML5.

Of course, the choices made by the author when the system presents alternatives, the specific content, and the actual look of the page (color schemes, logos, etc.) would still be left to the author. However, the actual HTML5 coding would comply with the semantics of HTML5.

### 4.3 Guidance for Universal Access

Very few web authors understand the difficulties encountered by web users with disabilities. So, while understanding the semantic of HTML5 is already a difficult matter, understanding the impact of the choices of HTML5 elements and attributes is even harder for most web authors. Moreover, web authors may not even be aware that there are special tools in HTML5 for use by assistive technology. Because they do exist in HTML5, this would be a good time for web authoring tools to take proactive measures to ensure authors use those features. Here are some samples of the kinds of help and guidance that could be provided.

- When the editor detects sequences of major sections or navigation points on a page it can suggest to the author that alternate navigations for impaired users should be provided. We have already illustrated how `aria-flowto` attributes can aid blind users.
- Authors should be prompted to use `aria` role types so that assistive technology can help impaired users understand web pages. For example, “img” elements being used in the role of “separator” should be handled differently than “img” elements with other roles. Similarly for “div” elements with role “alert” vs. role “banner” vs. role “definition”, etc.
- In [10] we proposed the development of new kinds of roles specifically for presentation to impaired users. Web authoring systems could add such roles to their libraries.

We also suggest that web developing systems incorporate the same assistive technologies that are available for web browsers so that authors can experience for themselves how impaired users will feel when using the documents being developed.

#### 4.4 Artificial Intelligence, Natural Language Processing, and the Future of Intelligent Web Authoring Tools

Artificial intelligence, and in particular natural language processing, has progressed to the point where useful tools that analyze text are now becoming available. For example, see [11] for a system that analyzes web content, aggregates information, and automatically generates news stories. Techniques from AI and NL could be applied to recognition of text within an HTML5 document, and the results could be used with rule systems and corresponding reasoning systems to recognize non-compliance with the semantics of HTML5. For example, text representing a citation that is not included in a “cite” element could be recognized and flagged. Such techniques could also be incorporated into HTML5 compliance checkers so that authors who do not use a sophisticated editing tool could still be helped to write semantically correct HTML5 documents.

## 5 Conclusion

We have shown by example how it will be difficult for web authors to understand the subtle semantics of HTML5 and therefore difficult for them to use HTML5 properly so that the goal of universal access can be achieved. We then proposed that web editors that can inform authors about the semantics and guide the authors as they develop web pages to correctly use the elements and attributes. The types of help from such editors can range from relatively simple explanation to sophisticated help using artificial intelligence and natural language understanding.

## References

1. World Wide Web Consortium. <http://www.w3.org/TR/2014/REC-html5-20141028/>
2. Berniers-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Sci. Am.* **284**, 29–37 (2001)
3. Hendler, J.: Agents and the semantic web. *IEEE Intell. Syst.* **16**, 30–37 (2001)
4. World Wide Web Consortium. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
5. World Wide Web Consortium. <http://www.w3.org/TR/wai-aria/>
6. World Wide Web Consortium. <http://www.w3.org/TR/2011/WD-role-attribute-20110113/>
7. World Wide Web Consortium. <http://www.w3.org/WAI/PF/aria-practices/>
8. Cameron, D., Smith, G., Daniulaityte, R., Sheth, A., Dave, D., Chen, L., Anand, G., Carlson, R., Watkins, K., Falck, R.: PREDOSE: a semantic web platform for drug abuse epidemiology using social media. *J. Biomed. Inform.* **46**, 985–997 (2013)
9. Dinh, D., Reis, J., Pruskia, C., Da Silveira, M., Reynaud-Delaitre, C.: Identifying relevant concept attributes to support mapping maintenance under ontology evolution. *Web Seman. Sci. Serv. Agents World Wide Web* **29**, 53–66 (2014)
10. Henschen, L., Lee, J.: Using semantic-level tags in HTML/XML documents. In: Proceedings of the 13th HCI International Conference, pp. 683–692 (2009)
11. Iacobelli, F., Birnbaum, L., Hammond, K.: Tell me more, not just more of the same. In: Proceedings of the 14th International Conference on Intelligent User Interfaces, pp. 81–90 (2010)