Similarities and Differences between HL7 CDA® Release 2.0 and HTML5

Background

The HL7 Clinical Document Architecture (CDA®) format is an XML standard developed by Health Level Seven® (HL7®) International to support the exchange of clinical documents and notes between healthcare providers. This standard is used worldwide by healthcare organizations, Health IT systems and national programs to exchange clinical information. It has also been approved as an ANSI (2005 and 2010) and ISO (2010) standard.

The standard supports the exchange of human readable clinical information in an HTML like format, and optional machine readable semantics which can be connected to the human readable narrative. The machine readable semantics are based on the HL7 Reference Information Model, which has also been used as the basis of numerous other standards in the HL7 Version 3 family. Other structured document standards from HL7 support the expression of quality measures, order sets, and documents for labeling medications and medical devices.

Recently, the CDA standard gained prominence in the US because it has become part of federal programs and regulation supported the Health Information Technology for Economic and Clinical Health Act that was passed as part of the Recovery Act.

The author of this paper is developing a proposal to replace the CDA XML content in the next release of that standard with HTML5 and Microdata. The purpose of preparing this comparison is to understand the gaps that must be filled in the current W3C working drafts in order to meet the expectations of Healthcare provider organizations and Health IT vendors developing applications that would use the HTML5 and Microdata standards instead of the current format.

Similarities and Differences

The narrative content of CDA draws heavily upon the same models for document narrative that were used in HTML. This should be no great surprise because the developers of the CDA format came out of the SGML community. In fact the CDA table model is a simple restriction of the HTML table model. Like HTML, CDA supports paragraphs, lists, tables and inline multimedia.

The table below shows a mapping of CDA elements and attributes to those found in HTML5 today.

CDA	**HTML5**
‹content›	‹SPAN›
‹linkHTML›[1]	‹A›
‹link›	
‹sub›	‹SUB›
‹sup›	‹SUP›
‹br›	‹BR›

[1] attributes are identical

‹footnote›	(not available)
‹footnoteRef›	
‹renderMultimedia›	‹IMG›
‹paragraph›	‹P›
‹list listType="ordered"›	‹OL›
‹list listType="unordered"›	‹UL›
styleCode="className"	style=”…”
	class="className"
ID="value"	ID="value"
<component><section>	<section>
<article>	<ClinicalDocument>
<title>	<hgroup>

Like HTML, a CDA document is divided into two parts, the header, and the body. The header contains metadata about the subject of the document (the patient), the author, the holder of the document (the custodian), the signer (legal authenticator), and a number of other important information items needed to manage its content and set the context (e.g., the visit date, location of encounter, et cetera) for the document.

Unlike HTML, CDA does not support scripting or forms.

CDA does support some control over rendering, although it is much more limited that HTML with CSS. There are a limited number of “styleCode” values defined by the standard, many of which duplicate functionality of the style attribute of HTML and CSS. User defined “styleCode” values can be added, and these can be displayed using appropriate CSS classes in HTML rendering.

Few applications natively render CDA Release 2.0 today. Instead, they transform the human readable content (and sometimes the machine readable semantics) into HTML or XHTML for rendering in a browser control. One project by the author demonstrated that it is also possible to render the CDA document narrative using an embedded CSS stylesheet. However, non-conformances in many current browsers and limitations in CSS rendering make this capability useful only as a “last resort”.

Editors for CDA documents are rare, and most often involve complex scripting with HTML or use of application UI libraries to generate the CDA narrative and machine readable semantics. HTML 5 supports the contentEditable attribute which enables editing of content in many user agents. Also, HTML5 forms and scripting support the creation of rich semantic data within content.

A CDA document can be just one component of a collection of resources that are needed to correctly render the content. Multimedia resources can be linked to, or rendered inline. However, CDA also provides the capability to include these resources inside the CDA document as base-64 representations of the binary content of those resources. The same capability can be accomplished in HTML in some user agents through the use of the data: url, but experiments by the author on multiple desktop and mobile user agents have shown that this capability is not yet widely available[2]. There are benefits to encapsulation of the clinical document into a single package, as it makes it harder to lose one or more of the pieces necessary to provide an accurate rendering.

CDA is extensible through the inclusion of extension elements in a separate namespace. CDA is normatively defined by an XML Schema. A valid CDA document must be schema valid once all extension elements have been removed. All extension elements reviewed and/or created by this author have been attempts to address missing features in the machine readable semantics in the document. A normative or informative XML schema for HTML5 could not be found by the author in the current HTML5 publication. The microdata capabilities of HTML5 provide the same level of extension capabilities that have been used for the CDA format.

Machine readable semantics in the CDA document are added using XML markup that is created from a transformation of a UML meta-model of the HL7 RIM into a domain specific object model (the CDA R-MIM), which is finally transformed into the CDA XML schema. The narrative content is based upon a manually created schema for the human readable text and titles that appear in the document.

Documents are composed of sections, which can be composed of other sections and machine readable entries containing the semantics expressed within a section. Linking the machine readable entries to sections is managed by using internal links in URL format. In HTML, documents have a section structure that can be defined loosely using the H1-H6 tags, or in a more structured format using section, heading and hgroup tags.

Machine readable semantics in CDA are expressed in a collection of properties using data types defined in yet another HL7 standard and associations between the classes described in HL7 RIM. The data types are defined in two schemas (datatypes-base.xsd and datatypes.xsd). Unlike the simple data types appearing in HTML, the data types in CDA are fairly complex. For example, they support not just time, but a variety of different representations that can support intervals and complex expressions of time such as every Tuesday at 10:00 and every Thursday at 1:00 between Memorial Day and Labor Day[3].

In Microdata found in HTML5, machine readable semantics are expressed in something called items. An item is effectively a property bag that can be typed, but the properties themselves cannot be typed without turning them into a complete item. There is no formal mechanism to describe the types described in Microdata. Instead, the type name is expected (but not required) to resolve into a web page that documents the type.

In processing the machine readable semantics of a CDA document (and other HL7 V3 standards), many implementers draw upon mapping utilities that allow the CDA schema to be stored in database structures.

[2] These experiments took place about 9 months ago, which is forever in Internet time.

[3] A medication dosing regimen that might be used for someone with swimming lessons twice a week to combat swimmers ear.

Gaps and Opportunities

The CDA Header is effectively hidden from view. The metadata expressed in the header can be displayed, or simply recorded for machine use, depending upon the application. The HTML5 <meta> tag supports a similar capability, however, that tag does not support composition of items into larger composites except through the itemref attribute, or creative use of <span> tags. The author has suggested that the <meta> content model be changed from being an empty tag, to allowing other <meta> tags as content.

Many implementers of CDA today are well versed in XML technologies, including schema, XSLT, DOM, and database mapping tools. The lack of a schema for HTML5 concerns many developers because they are accustomed to having that resource for both development and validation purposes.

The lack of a standard representation supporting validation of microdata against its type in HTML5 documents makes it difficult for systems to ensure the correctness of microdata content. The author has experimented with generating microdata in the HTML5 rendering the type definition, and is certain that a validation mechanism could be developed based on such a format.

The author is aware of number of applications that rely on RDF and OWL to provide clinical decision support. Support for a reversible transformation from Microdata to RDFa would be desirable to enable these applications to take advantage of HTML5 for expression of clinical documentation. XML luminary Jeni Tennison provides a detailed gap analysis on this topic in a recent blog post[4].

One notable capability of RDFa is the ability to provide machine readable content using a content attribute. In a recent prototype developed to show the viability of CDA Release 3.0 (currently under development by HL7), the author notes that it would be easier to provide certain content in machine readable form using this mechanism.

Summary

There is a great deal of opportunity to replace existing markup used in CDA Release 2.0 with a constrained form of HTML5 + Microdata. HL7 is currently developing CDA Release 3.0, and is reviewing a proposal by the author to create an implementation technology specification that would allow CDA Release 3.0 and other structured documentation to be conveyed using HTML5 and Microdata.

It is the view[5] of this author that so doing would increase the number of developers with the technical skills necessary to support the use and exchange of clinical documention, and would support great increases in innovation in this space. While the points made in this document focus on similarities between HTML5 and CDA Release 2.0, other types of structured documentation used in healthcare could also benefit from the use of HTML5. This includes clinical guidelines, public health alerts, adverse event reporting, order sets, medicinal and medical device labeling, et cetera.

[4] Microdata and RDFa Living Together in Harmony, Jeni Tennison, 8/20/2011. Found on the web at http://www.jenitennison.com/blog/node/165

[5] The opinions in this document are those of the author, and are not representative of those of his employer, or the standards organizations that he represents.

About the Author

Keith W. Boone is a standards geek for GE Healthcare. In this role, he represents his employer to healthcare standards organizations and activities, including HL7, Integrating the Healthcare Enterprise (IHE), and the Standards and Interoperability Framework program of the Office of the National Coordinator of Healthcare IT. He held a prior position as Architect at eBusiness Technologies, an XML and SGML development firm. There he worked alongside numerous editors of W3C standards and cochairs of W3C working groups, including XPath, XLST, and XML, and was a member of the DOM2 Interest Goup.

He is presently a member of the board of HL7, and a past co-chair and current member of the Structured Documents Workgroup (the workgroup responsible for CDA). He also co-chairs the IHE Patient Care Coordination Planning committee. He has written more than a dozen implementation guides on HL7 CDA , and is also the author of The CDA™ Book published by Springer.