Enabling A Low Cost Semantic Web

Trevor Kennedy
Business Information Technology
Virginia Tech
1007 Pamplin Hall
Blacksburg, VA 24061
(540) 231-9128
tkennedy@vt.edu

ABSTRACT
Microformats are a popular, easy way to add structure to web-based content and present an opportunity to transform Web 2.0 into the Semantic Web. Microformats however, suffer from certain limitations that should be addressed to support this goal. We provide an overview of microformats and RDFa and show how they can be aligned with the goals of the Semantic Web. We propose a way to overcome the limitations of current microformat implementations. We argue that a directory of schemas is the missing link between the web of today and the Semantic Web of the future. We demonstrate that existing microformat and RDFa content can be converted into Semantic Web technologies at a low cost thus dramatically lowering the barrier to entry of the Semantic Web.

Keywords
Semantic Web, Web 2.0, Microformats, Ontology, Metadata, RDF.

1. INTRODUCTION
In this paper we will give an overview of the Semantic Web and discuss the place of microformats and Resource Description Framework – in – attributes (RDFa) within it. The main goal of this paper is to show that easy to implement semantic languages such as microformats and RDFa can be converted into Semantic Web technologies and that they align with the original vision of the Semantic Web.

1.1 Web 2.0
According to Google, the World Wide Web contained 1 trillion unique Uniform Resource Locators (URLs) in 2008 and is growing at a rate of several billion pages per day [1]. At that scale, moving from today's web to the Semantic Web of the future is a monumental task and any technique that does not reuse existing content or automate the conversion process simply isn't practical. In this paper we refer to the current state of the internet as being Web 2.0.

1.2 Microformats
Microformats are a grassroots movement that seeks to marry blog content with well established industry standards using simple extensions of standard Hyper Text Markup Language (HTML) tags [2]. Principally designed for ease of use, microformats leverage existing Extensible Hypertext Markup Language (XHTML) attributes to encode common types of semi-structured data within the content of standard web pages [3,4]. Microformats are made possible by exploiting the fact that modern web browsers simply ignore what they do not understand.

1.3 Semantic Web
Ten years after the first Semantic Web standards were published, Semantic Web technologies have failed to be implemented on the Internet in any substantial way [5]. The ideal of the Semantic Web remains largely unrealized, in part, because it is still difficult to achieve using today's web as the starting point [6]. There is a wide disconnect between Semantic Web technologies like Resource Description Framework (RDF) and the data providers of today [7].

1.4 Methodology
First, we show that there are more implementations of microformats in use today than there are implementations of Semantic Web technologies, specifically RDF and RDFa. We then propose using namespaces and XML Schemas to address the limitations of current microformat implementations. Third, we show that a conversion to Semantic Web technologies is feasible using existing standards. Fourth, we demonstrate that microformats and our proposed conversion framework are compatible with current Web 2.0 technologies. Finally, we propose a technique to create a Semantic Web ontology from a directory of microformat schemas.

2. WEB 2.0
Web 2.0 defines the Internet as a platform for applications that harnesses collective intelligence [8]. The freeing of data is a core component of Web 2.0 [9]. Linked Data is a Web 2.0 concept for lowering the barriers required to connect separate but related sources of data on the web [10]. The rules of Linked Data are to use HTTP URIs to identify resources, to provide useful information about resources when dereferenced from Uniform Resource Identifiers (URIs), and to provide links to other resources as a means of improving information discoverability [11]. This vision is inline with the goals of the microformat and Semantic Web communities.

2.1 Metadata
The XHTML specification has two mechanisms for describing metadata: the *meta* element and the *profile* attribute of the head element (although the specification explicitly does not define the format of a profile) [12]. Previous attempts to add metadata involved using the HTML *meta* tag to add hidden (invisible to human readers) metadata which was often abused to improve search engine rankings [13]. These attempts failed due to a lack of formalism by specification designers and misuse by content publishers.

2.2 Microformats
The main driving forces behind microformats are the increased use of Cascading Style Sheets (CSS) implemented with the HTML *class* attribute and the desire to extract and aggregate information from blog postings [14]. Marking up a web page with microformats using the *class* attribute, is in effect superimposing a second grammar onto the markup [14]. That is why microformats

are so important; they present a low cost way to bridge the gap between the content producers of today's web and the visionaries of a future Semantic Web.

2.2.1 Definition

Microformats (μF) are simple standards based idioms for data formats applied to XHTML and are designed to convey meaning in a manner that is human readable first and machine readable second [13,14]. The simpler microformats like relTag and relLicense are known as elemental microformats [3]. Patterns such as hCard and hCalendar that are composed of one or more microformats are known as compound microformats [3].

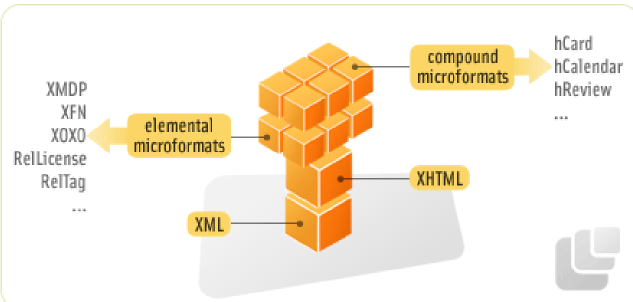

Figure 1. The structure of microformats [15]

The HTML *class* attribute was first proposed in 1993 as part of the draft specifications for HTML 3 with the intention to allow every element on the page to be styled using CSS [14]. Microformats repurpose the *class* and *rel* attributes to give structure and semantic meaning to otherwise unstructured webpage content [13].

```
<div>
<span>anonymous</span>
<a href="http://example.com">Example.com</a>
<span>5.0</span>
<blockquote>
Example.com is a great reserved domain name
</blockquote>
</div>
```

Figure 2. A basic website review in XHTML

```
<div class="hreview">
<span class="reviewer vcard">
<span class="fn">anonymous</span>
</span>
<div class="item">
<a class="fn url" href="http://example.com">Example.com</a>
</div>
<span class="rating">5.0</span>
<blockquote class="description">
Example.com is a great reserved domain name
</blockquote>
<span class="type">website</span>
</div>
```

Figure 3. A website review using the hReview microformat

2.2.2 History

The notion of annotating hyperlinks with metadata was part of the original design of the World Wide Web [14]. It was implemented in HTML using the rel and rev attributes and dubbed tagging [14]. Prominent web designers and bloggers started to publicize this capability and created the microformat initiative in the process [13].

The relTag draft specification, the microformat implementation of tagging, utilizes the *rel* attribute to annotate hyperlinks with keywords, and has made blog aggregators like Technorati possible [16]. RelTag is the most successful microformat for human generated content and within the first six months of its introduction as a way to semantically tag blog content, some 20 million blog posts had implemented the microformat [17].

2.2.3 Popularity

The microformat movement is experiencing network effects of the likes unseen by Semantic Web technologies such as RDF and RDFa [5,18]. Industry leaders have recognized the need for agreed upon microformats [19] and have won the backing of major Web industry players, with Yahoo! alone publishing over one billion microformat enabled pages [5]. They are used on an estimated few hundred million additional webpages [20].

Microformats are currently used by a number of popular sites including the social networking sites, online calendars, and email systems [20]. Several search services exploit the semantic information defined by microformats [21]. hReview is used by for publishing reviews while hCard is used for profile information [18]. The wine review website Cork'd, features hReview for wine reviews, hCard for reviewers, and relTag for "tasting tags" [18].

Figure 4. Current users of microformats

2.2.4 Benefits

Microformats are simple, compatible with existing technology, and decentralized [14]. Other semantic technologies, such as RDF, have a much higher barrier to entry for web developers [2]. Microformats lower the barrier of adoption for certain classes of data and in the process help spur the adoption of the Semantic Web [21].

A 2005 study of class names in over a billion documents concluded that there is no uniformity or logic in the naming of classes [21]. Microformats do not interfere with CSS, and can be implemented by mainstream web designers in a relativity straightforward fashion [13]. They do so using a compact syntax that complements existing HTML standards and evolving patterns (such as relLicense).

Due to the reuse of existing XHTML elements and CSS markup techniques, microformats add little overhead to the page. This is a brilliant idea due to its ease of implementation and backwards compatibility.

Current blogging software makes file attachments, such as RDF documents difficult [17]. Microformats address this limitation by including the metadata inside the XHTML markup. This self-containment also prevents distance decay [22] of the meta data by storing the semantics within the document itself. For example, a .vcf file would not be needed if the contact information itself were annotated with the specifications of the vCard format [17].

One of the main appeals of microformats is the ability to codify common practices in a decentralized, grassroots fashion [17]. Microformats reuse existing standards as much as possible including vCard for contact information and iCalendar for events; both are Internet Engineering Task Force standards. The hReview specification, developed jointly with collaborators from AOL, Microsoft, and Yahoo [3], builds on divergent standards including those of the Platform for Internet Content Rating Services [3].

Translating an existing schema into a microformat is painless and straightforward [3]. Browser extensions detect, parse, and share the microformat snippets [3,21]. Design tools allow for microformats to be applied by webpage authors at design time with little effort [18,23]; the only required knowledge to use microformats is basic XHTML [21].

XHTML Meta Data Profiles (XMDPs) provide semantic elements for XHTML documents in a fashion that is both human and machine readable. XMDP is a directory of property names and values defined using XHTML definition list elements (dl, dt, dd) that can be used to extend microformat class attribute values. [24].

2.2.5 Limitations

Microformats, however, are not without limitations that must be addressed before microformats can be considered a candidate migration path to the Semantic Web. For example, Microformats that use the XHTML abbreviation attribute, *abbr*, to display easy to read versions of machine-readable IISO-8601 timestamps [17,3] suffer from accessibility issues when screen readers are used [25].

Although embedded, we argue that microformats are not self contained because *a priori* knowledge of the format is required to identify the implementation and infer any semantic meaning. The simplicity of microformats can also be considered a weakness because their narrow scope does not allow complex relationships be formally defined [21].

The microformat community relies on a single wiki repository for microformat specifications which is not scalable nor universally recognized as authoritative [14]. Microformats are difficult to validate because there is no published schema to compare instances against [14]. Additionally, their use of allowing multiple tokens in a single attributes value makes validation problematic.

Microformats use a flat predefined namespace that cannot be extended [26]. The microformat specifications do however support profile URIs for vocabulary validation although this potentially violates the self-containment principal [20].

Although compound microformats incorporate elemental microformats into their design they do not truly support inheritance. For example, if a group of web developers wanted to develop an hWineReview microformat they could do so. However there would be no way to associate this new microformat with a parent hReview microformat. This results in current microformat parsers ignoring the hWineReview microformat until manually updated to support the new microformat.

2.2.6 Examples

The names of microformats typically start with a lowercase *h* to indicate HTML [20]. Established drafts exist for hCalendar (based on iCalendar), hCard (based on vCard), and hAtom (based on a subset of the Atom syndication format) [4]. Newer drafts include hResume, hReview, hProduct, and Geolocation [27]. Dublin Core Microformats (DCMF) are a way to encode the Dublin Core bibliographic metadata within XHTML [21].

Table 1. Common microformats adopted from [28,15]

Name	Elemental	Description
relLicense	Yes	Indicates a content license
relTag	Yes	Allows for annotating a hyperlink with keywords
hCard	No	Suitable for describing vCard (RFC2426) contact information
hReview	No	Suitable for describing reviews
hCalendar	No	Based on iCalendar (RFC2445) for calendaring and events
hAtom	No	Used for syndicating content.
WebSlice	No	Allows users to subscribe to a portion of a webpage in Microsoft Internet Explorer 8 [53]

2.3 XML

The eXtensible Markup Language (XML) is a widely accepted data exchange format [29]. XML uses hierarchies to implicitly encode logical relationships and does not provide any semantic meaning about data unless accompanied by a schema [7]. Like microformats, XML allows one to arbitrarily define tags [30] which is not conducive to information sharing unless it is associated with metadata.

XML is more human-readable than the Semantic Web's RDF [31] and can be mapped to RDF using Extensible Stylesheet Language Transformations (XSLTs) [32]. The XSLT language is a World Wide Web Consortium (W3C) recommendation that allows XML documents to be transformed from one schema to another [33].

2.4 XML Schema

XML Schema is backed by the W3C [34] and is the most widely accepted schema on the web [35]. XML Schema addresses many more issues than just validation [36]. XML Schemas define structure [37] and provide support for basic data types, attribute constraints, sophisticated structures and name spaces [30]. The six types of semantic relationships found in XML Schema are identical, equal, equivalent, subset, unique, and incompatible [38]. XML Schema supports the inheritance of complex types [34] and allows for substitution and uniqueness [39].

XML Schema is not without limitations. Unlike other schema languages, XML Schema does not support conditional definitions [34]. XML Schema also provides little support for semantic knowledge [40] does not apply semantic meaning [30].

2.5 RDDL

Resource Directory Description Language (RDDL) is an XHTML format that embeds machine-readable semantics in human-readable content using XLink and can be considered a precursor to microformats [33]. RDDL represents a sequence of resources and used to describe the nature of resources found within a namespace [41]. RDDL extends XHTML by adding a resource element that serves as an XLink to the referenced resource [41]. XSLT can be used to transform one resource into another [41].

```
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:rddl="http://www.rddl.org/">
<body>
 <rddl:resource xlink:href="http://www.example.com/hreview.xslt"
   xlink:title="XSLT for hReview"
   xlink:role="http://www.w3.org/1999/XSL/Transform"
   xlink:arcrole="http://www.rddl.org/purposes#xslt-extension">
  <a href=http://www.example.com/hreview.xslt">hReview XSLT</a>
 </rddl:resource>
</body>
</html>
```

Figure 5. RDDL file for an hReview XSLT adopted from [42]

2.6 RDFa

The impressive rate of adoption of microformats amongst web developers spurred the W3C to standardize RDFa [5]. RDFa is the newest attempt by the Web 2.0 community to annotate webpages with metadata. RDFa provides an inline syntax for expressing single graph RDF structures in pure XHTML [20].

```
<span xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Review">
  <span rel="v:itemreviewed">
     <span about="http://example.com" property="v:name">
       Example.com
     </span>
  </span>
  <span rel="v:reviewer">
     <span property="v:name">anonymous</span>
   </span>
  <span property="v:rating">5.0</span>
   <span property="v:summary">
    Example.com is a great reserved domain name
   </span>
 </span>
```

Figure 6. A website review using the RDFa adopted from [43]

The four guiding principles of RDFa are interdependence and extensibility, the Principle of Least Astonishment, DRY Principle, Principle of Visible Metadata, and self-containment [20]. The Principle of Least Astonishment states that when an element of a user interface is ambiguous, the behavior that least surprises the human user is the appropriate one [13]. The Principle of Visible Metadata refers to the practice of making metadata transparent to the consumer of the document to avoid being abused or becoming erroneous [2]. Finally, the Don't Repeat Yourself (DRY) Principle proclaims that there should only be one authoritative, unambiguous representation for each piece of knowledge in a system [13].

RDFa uses a vocabulary that can be freely extended which along the provided namespace support allow multiple vocabularies to be used simultaneously [44]. RDF and RDFa combined are in use by several portal sites [11] and an estimated few million documents [45].

Namespacing in RDFa is implemented using Qualified Names (QNames) which can break CSS [13]. RDFa also makes use of CURIEs (as opposed to URIs) which are only supported in XHTML 2 [13], a standard that has been abandoned by the W3C [46].

2.7 Comparison

Although divergent from microformats, both technologies support plain literals, are well formed and have no ill effect on browser behavior [20]. Like RDFa, the Principle of Least Astonishment, the Principle of Visible Metadata, and DRY [13] are shared by microformats. Unlike microformats, RDFa supports XHTML namespaces that allow publishers to create new and extend existing vocabulary.

Table 2. Metadata comparison adopted from [13]

Dimension	Microformats	RDFa	XML
XHTML Compatible	Yes	Yes	No
User Defined Types	No	Yes	Yes
Visible Metadata	Yes	Yes	No
Central Authority	None	W3C	W3C
Schema Language	XMDP	RDF Schema	XML Schema
Compact Syntax	Yes	Partial	Yes
Namespace Support	No	Yes	Yes
Attribute Usage	Existing	Existing	New
Popular	Yes	No	Yes

3. SEMANTIC WEB

XML is the first important technology for developing the Semantic Web because it allows content owners to annotate page text [47]. The second technology, RDF, provides inference rules for use by the Semantic Web [47]. The third technological aspect of the Semantic Web are ontologies described using OWL.

3.1 Definition

The Semantic Web is a vision of the web that has been annotated with a well-defined structure that provides the ability for software agents to carry out sophisticated tasks [47]. The Semantic Web consists of three primary layers: the data layer, the schema layer, and the logical layer [48].

The approach of the Semantic Web is to develop languages and grammar for expressing information in a machine-readable fashion [48]. The desire to create common general model that any application can be mapped to stems from the guiding principle of minimalist design [48].

The essential properties of the Semantic Web are that it be universal, decentralized, structured and provide inference rules [47]. Metadata interoperability is a fundamental requirement of the Semantic Web and has been exacerbated by the need for more complex data descriptions [40].

3.2 Ontologies

The third component of the Semantic Web are ontologies. An ontology is an explicit specification of a collection of information [30] designed to support disambiguation and inference [47]. They use controlled vocabularies to provide organization [49] and may contain explicit facts and context about objects [50]. Ontologies are defined independently of the actual instance data [51].

The argument in favor of the use of ontologies has been won [6]. The most common type of ontology created for the Semantic Web consists of taxonomies, which define object classes and their relations to each other [47], that are paired with sets of inference rules [47].

3.3 RDF

Resource Description Framework (RDF) is formal model for providing machine-readable descriptions of data [37] that was given W3C Recommendation status in 1999 [21]. RDF uses triples (subject, property, value) in a graph structure as opposed to a hierarchy like that used by XML [52]. RDF supports the interoperability of XML data [37] by providing support for XML Schema data types and QNames [30,31].

Unlike XML however, RDF allows a single property to have multiple values [30] which allows different vocabularies to be joined [31]. This approach is highly formalized and leads to considerable overhead [28].

```
<rdf:Description rdf:about="http//purl.org/stuff/rev#Review">
   <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema"/>
   <rdfs:label xml:lang="en">Review</rdfs:label>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/stuff/rev#"/>
  </rdf:Description>
 <rdf:Descriptionrdf:about="http://purl.org/stuff/rev#rating">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax"/>
  <rdfs:label xml:lang="en">rating</rdfs:label>
   <rdfs:isDefinedBy rdf:resource="http://purl.org/stuff/rev#"/>
  </rdf:Description>
 <rdf:Description rdf:about="http://purl.org/stuff/rev#reviewer">
   <df:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax"/>
     <rdfs:label xml:lang="en">reviewer</rdfs:label>
    <rdfs:domain rdf:resource="http://purl.org/stuff/rev#Review"/>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/stuff/rev#"/>
  </rdf:Description>
   <rdf:Description rdf:about="http://purl.org/stuff/rev#type">
   <df:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax"/>
   <rdfs:label>type</rdfs:label>
   <rdfs:isDefinedBy rdf:resource="http://purl.org/stuff/rev#"/>
  </rdf:Description>
```

Figure 7. The hReview microformat modeled in RDF [53]

3.4 RDF Schema

RDF Schema (RDFS) became a W3C recommendation in 2004 [6]. RDFS allows users to define classes, properties, and values [37]. RDFS enables RDF descriptions to be extended with a structured vocabulary that defines semantic relationships for ontological purposes [51].

RDFS allows for subclassing and inheritance but cannot express equivalence between properties or the union/intersection of classes [37]. Although RDFS provides support for rich semantic descriptions, it provides limited support for the specification of usage constraints [40]. RDFS supports conditional definitions [54] can be embedded with XML Schema [40].

3.5 OWL

Web Ontology Language (OWL) is a W3C recommendation that enables representation of ontologies that are amenable to making decisions [6]. OWL allows for ontologies to be processed by machines [28] and is designed to support inference and information sharing using subsumption and classification [6].

OWL uses parts of XML Schema to express the structure of domain values [7]. OWL supports complex classes and is based on RDF constructs [31,33]. The linking capability provided by RDF provides OWL the ability to define ontologies in a distributed manor across systems [6].

3.6 Challenges

Many web developers had hoped that the more strongly-typed XML would eventually replace legacy HTML. However, with the exception of XHTML, the XML and HTML communities have continued to develop separately [14] as is likely to be the case with Web 2.0 and the Semantic Web. Therefore, the main challenge is to provide a way that expresses data and reasoning rules in a way that allows for existing content to be exported to the Semantic Web [47].

Microformats have the potential to bridge the gap between today's web and the Semantic Web of the future by brining structure to the content of existing web pages. We propose a low cost framework to bring the billions of pages of existing microformat and RDFa content into the Semantic Web.

4. METHODOLOGY

Microformats have the backing of industry content providers and microformat annotated pages have already reached the billions mark. Several previously published papers have positioned microformats as the missing link to the Semantic Web [3,17,28]. However, none of these authors have presented a detailed methodology for how to implement such a link.

To overcome the limitations of microformats we propose several ideas. First, we propose addressing their profile limitations using XML Schema. Additionally we seek to further the concept of using namespaces in microformats to enhance support for inheritance, interoperability, and logical comparison. Although microformats do not support inference natively, we believe they can be mated with RDF to do so. In addition to microformats, we integrate support of RDFa into our framework. We also propose a schema directory of microformat and RDFa and specifications that could be contributed to and used by both the Web 2.0 and Semantic Web communities.

4.1 Ideology

Although microformats sidestep the W3C Semantic Web “technology stack” of RDF and related concepts, they still support the same goals [17]. Microformats support the original appeal of XML and since they provide machine-readable metadata but do not support implicit inference, they can be thought of as first step towards the Semantic Web [17] The main point of contention between microformats and the Semantic Web is one of philosophy.

Table 3. Comparing microformats and the Semantic Web

Dimension	Microformats	Semantic Web
Central Authority	None	W3C
Readability	Human first	Machine First
Implementation	Bottom up	Top down
Learning Curve	Low	High
Validation Support	No	Yes
Inheritance	No	Yes
Barrier to Entry	Low	High
Inference Support	No	Yes
XHTML Compatible	Yes	No
Webpage Usage	Billions	Millions

4.2 Overview

There are four components to our proposed low cost Semantic Web. The first component is comprised of easy to implement, XHTML compatible semantic content annotation languages. Web 2.0 XHTML documents annotated with microformats and RDFa will be used for our purposes.

The second component is a schema framework that allows these semantic content annotation languages to be associated with robust, machine-readable specifications and definitions. We implement this framework using XML Schema.

The third component is a conversion process that can transform semantically annotated content into machine-readable versions compatible with the Semantic Web. We propose using GRDDL and XSLT to perform this translation.

The fourth piece of our implementation is a repository of schemas and translation specifications. This machine-readable directory will allow for related ontologies to be created, thus, ushering in the Semantic Web at a low cost.

4.3 Framework

Schemas give rise to validation, inheritance, namespacing, and data typing. The reason that schemas are need for microformats is to determine if a particular implementation is valid and known, valid but unknown, or invalid and known without human intervention.

Schemas that define rules and data types will allow microformats contained within XHTML documents to be validated [55]. With proper schemas defined, applications would be able to recognize any microformat without needing to have prior knowledge of any particular format. If the microformats are defined with a XML Schema simple queries could be used to retrieve the information and translate it into RDF [55].

4.3.1 XMDP

XMDP is the most popular schema for microformats. XMDP is guided by the principles of simplicity, reuse, and minimalism [12]. These guiding principles are design limitations when it comes to describing data in a machine-readable fashion. We argue that since XMDP is not visible metadata it is not imperative that be human readable because the content of the page is human readable and this supplemental XMDP metadata should be primarily designed to support automation and the Semantic Web.

Because XMDP implemented at the page level using the HTML *profile* attribute, the first profile specified in a document has precedence over subsequent ones, so there is the potential for conflicts between two XMDP schemas to occur [56]. Furthermore, XMDP does not provide a way to specify data types or relationships, which make machine validation difficult [56]. Although, XMDP is not designed to support namespacing it could support a lightweight namespacing mechanism using the *link* HTML element to create a QName [24].

```
<dl class=”profile”>
<dt>class</dt>
<dd>
<dl>
<dt>hreview</dt>
<dt>version</dt>
<dt>type</dt>
<dt>item</dt>
<dt>fn</dt>
<dt>url</dt>
<dt>reviewer</dt>
<dt>rating</dt>
<dt>description</dt>
</dl>
</dd>
```

Figure 8. XMDP schema for hReview

4.3.2 MDL

Markup Definition Language (MDL) is a grammar that can be used to specify a schema for a namespaced microformat. MDL builds on the HTML Profile concept used by XMDP that associates natural language text with identifiers.

An MDL schema provides human readable documentation of the format as well as a declarative syntax that can be processed automatically using a simple XSLT stylesheet [14]. When transformed using XSLT the microformats can be extracted into a purely XML document that supports validation [14]. Even though MDL supports namespaces and allows microformat to be translated into XML, it does not provide support for data types or inheritance.

```
<?mdl version="1.0"?>
<microformat version="0.3">
  <name>hreview</name>
  <class>
    <inhtml>type</inhtml>
    <inxml>type</inxml>
  </class>
  <class>
    <inhtml>item</inhtml>
    <inxml>item</inxml>
  </class>
  <class>
    <inhtml>reviewer</inhtml>
    <inxml>reviewer</inxml>
  </class>
  <class>
    <inhtml>rating</inhtml>
    <inxml>rating</inxml>
  </class>
  <class>
    <inhtml>description</inhtml>
    <inxml>description</inxml>
  </class>
 </microformat>
```

Figure 9. MDL schema for hReview

4.3.3 Schematron

Schematron, an International Organization for Standardization (ISO) standard [57], is unique amongst schema languages because it focuses on validating schemas based on patterns instead of grammatically defining them [34]. Schematron is only intended for validation and does not provide a data model [30]. Schematron is fully compatible with XML Schema and rule markup can even be embedded within XML Schema [30].

Unlike XML Schema, Schematron supports choice among attributes and used a content model that is open by default. [34]. Schematron uses XPath to define relationship rules and provides support conditional validation based on instance values [30]. This is important functionality for the validation of microformats because the content of a microformat will depend on the value of various XHTML *class* attributes specified on the instance webpage.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron">
<xs:element name="hreview">
 <xs:complexType>
   <xs:sequence>
    <xs:element name="item">
       <xs:annotation>
       <xs:appinfo>
        <sch:pattern id="itemNameTest">
          <sch:rule context="item">
           <sch:report test="count(@fn)=1">
                  Item being reviewed must have a name
            </sch:report>
          </sch:rule>
        </sch:pattern>
       </xs:appinfo>
      </xs:annotation>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
 </xs:schema>
```

Figure 10. A Schematron rule for hReview adopted from [58]

4.3.4 Proposal

We recommend using a profile schema that explicitly states the required and optional attributes along with their strongly typed legal values, allows for versioning and inheritance, as well as nesting, and supports migration to RDF. The schema would define common structures and usage conventions such as which XHTML attribute and elements a property can be bound to [55]. The schema will be linked to the microformat using a namespace URI.

Table 4. Schema comparison adopted from [34]

Dimension	MDL	Schematron	XMDP	XML Schema
Includes	No	No	No	Yes
User Defined Types	No	No	No	Yes
Domain Constraints	No	Yes	No	Yes
Conditional Definition	No	Yes	No	No
Default Values	No	Yes	No	Yes
Min and Max Occurrence	No	Yes	No	Yes
Inheritance	No	No	No	Yes
Uniqueness	No	Yes	No	Yes
Dynamic Constraints	No	Yes	No	No
Explicit Null	No	No	No	Yes
RDF Conversion	No	No	No	Yes
Authority	None	ISO	None	W3C

Based on a comparison of the features provided by each schema definition language and the requirements imposed by microformats, we propose using a combination XML Schema and Schematron to create schemas for microformats.

XML Schema provides the structure, validation, namespaces, datatypes, constraints, inheritance, and W3C compliance needed by microformats. The complexity introduced by the microformat implementation of attribute values requires the pattern and rules based functionality of Schematron to support conditional definitions. The Schematron rules would be embedded within the XML Schema file. XSLT may then be used to extract the attribute values from an XHTML page.

We propose that the *profile* attribute of the XHTML Head attribute be used just like in XMDP, only that the file linked to use XML Schema nomenclature to describe any formats used on the page. As an alternative, the DCMF implementation of schema referencing using the XHMTL *link* element's *rel* attribute could be utilized to contain the value of the class name of the microformat and a hypertext link to the corresponding XML Schema file.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="hreview">
  <xs:complexType name="hreview">
   <xs:sequence>
    <xs:element maxOccurs="1" name="version" type=”xs:decimal”>
    </xs:element>
    <xs:element maxOccurs="1" minOccurs="0" name="type">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="product"/>
       <xs:enumeration value="person"/>
       <xs:enumeration value="place"/>
       <xs:enumeration value="website"/>
       <xs:enumeration value="url"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
    <xs:element maxOccurs="1" minOccurs="1" name="item">
     <xs:complexType>
      <xs:choice>
       <xs:element maxOccurs="1" minOccurs="1" name="fn">
        <xs:complexType>
         <xs:complexContent>
         <xs:element maxOccurs="1" name="url" type="xs:anyURI"/>
         </xs:complexContent>
        </xs:complexType>
       </xs:element>
      </xs:choice>
     </xs:complexType>
    </xs:element>
    <xs:element maxOccurs="1" name="reviewer" ref="hCard"/>
    <xs:element maxOccurs="1" name="rating" type="xs:decimal"/>
    <xs:element maxOccurs="1" name="description" type="xs:string"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Figure 11. An XML Schema for hReview

A third option is to use the vendor-specific extensions as defined in the specifications for CSS 2.1 [59] to link a schema to a microformat instance. A dozen known vendor-specific extensions are known to exist [59] and they have the benefit of being unobtrusive and standards complaint. This method is similar to the method proposed by the RDF-EASE draft which attempts to bridge RDF and XHTML using CSS [60].

QNames are implemented using namespace prefixes which are declared using the at-keyword of the CSS specification [59]. The vendor identifier “-microformat-” would be used to create the extension. To map each microformat XHTML class *id* to the appropriate schema property would require the web page's CSS style sheet to be modified with the vendor extension.

4.3.5 Namespaces

Namespacing allows for peer microformats to be related to one another and supports the creation of an ontology. Namespaces that provide links to the appropriate XML Schemas and Schematron rules will make validation possible, mitigate ambiguities, and eliminate versioning issues.

Name conflicts can be overcome by using QNames to connect the markup used by a document with the URI of the corresponding schema [14]. Qnames are created when a namespace prefix followed by a colon is added as a suffix to each local name [33]. The namespace prefix is associated with a namespace URI by placing a namespace declaration at the document level [33]. Because there is no single accepted way to convert a QName into a URI, a mapping mechanism must be provided [33].

```
@prefix ex "http://example.org/hreview";
.hreview
{
 -microformat-schema: url("http://example.org/hreview");
 -microformat-typeof: "ex:hcard";
}
.hreview .fn
{
 -microformat-property: "ex:name";
 -microformat-datatype: "xs:string";
}
.hreview .url
{
 -microformat-property: "ex:url";
 -microformat-datatype: "xs:anyURI";
}
.hreview rating
{
 -microformat-property: "ex:rating";
 -microformat-datatype: "xs:decimal";
}

<div class="hreview">
<span class="reviewer vcard">
<span class="fn">anonymous</span>
</span>
<div class="item">
<a class="fn url" href="http://example.com">Example.com</a>
</div>
<span class="rating">5.0</span>
<blockquote class="description">
Example.com is a great reserved domain name
</blockquote>
<span class="type">website</span>
</div>
```

Figure 12. A vendor-specific extension example for an hReview implementation adopted from [60]

One way microformats could support QNames is to add a namespace prefix to the value of the name present in the *class* attribute e.g. *class=”hcard”* becomes c*lass=”mf:hcard”* [14]. Implementing the namespace within the value of the class attribute has the benefit of being fully XHTML compatible and eliminating any ambiguities as to which schema or profile belongs to a given instance. Note that the W3C CSS Recommendations do not include support for namespaces although modern browsers do as long as fixed element prefixes are used [14].

We propose using RDDL and XHTML to create a machine-readable namespace. RDDL would associate a microformat instance with its corresponding schema, the microformat directory, help documentation, XSLT files, XML Schema, and RDF Schema. For RDFa implementations, RDDL provides a mechanism to link an implementation to a standard grammar in the namespace or directory.

An existing XMDP implementation could be extended to support a lightweight namespacing mechanism using the *link* element to create a QName for the associated RDDL file [24]. Support for uniform XML output will be provided by specifying a link to an XSLT in addition to the XML Schema in the namespace RDDL file. Schematron can be used to perform validation on RDDL implementations [33].

```
<html xmlns="http://www.w3.org/1999/xhtml"
 xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:rddl="http://www.rddl.org/">
<body>
<rddl:resource link:href="http://www.example.com/hreview.xslt"
  xlink:title="XSLT for hReview"
 xlink:role="http://www.w3.org/1999/XSL/Transform"
 xlink:arcrole="http://www.rddl.org/purposes#xslt-extension"/>
<rddl:resource xlink:href="http://www.example.com/hreview.css"
  xlink:title="CSS style sheet for hReview"
 xlink:role="http://www.isi.edu/in-notes/iana/assignments/media-
types/text/css"
  xlink:arcrole="http://www.rddl.org/purposes#render"/>
<rddl:resource xlink:href="http://www.example.com/hreview.xsd"
  xlink:title="XML Schema for hReview"
  xlink:role="http://www.w3.org/2000/10/XMLSchema"
 xlink:arcrole="http://www.rddl.org/purposes#schema-validation"/>
<rddl:resource xlink:href="http://www.example.com/hReview.html"
  xlink:title="XMDP profile for hReview"
  xlink:role="http://www.w3.org/1999/xhtml"
 xlink:arcrole="http://www.rddl.org/purposes#definition"/>
<rddl:resource xlink:href="http://www.example.com/parent.xsd"
  xlink:title="Parent XML Schema for hReview"
  xlink:role="http://www.w3.org/2000/10/XMLSchema"
 xlink:arcrole="http://www.rddl.org/purposes#reference"/>
<rddl:resource
  xlink:href="http://www.microformats.org/directoy#hReview"
  xlink:title="Link hReview to in the schema directory"
  xlink:role="http://www.rddl.org/"
  xlink:arcrole="http://www.rddl.org/purposes#directory"/>
</body>
</html>
```

Figure 13. Example hReview RDDL namespace

4.3.6 Versioning
Microformat versioning is currently implemented using a version attribute. There are two ways to provide versioning: mimic the XML *version* attribute on the root attribute, or use a linking mechanism to provide links to previous versions as exhibited by OWL [33]. The first approach could be problematic especially if different versions use the same URI [33]. The second linking approach is more compatible with the Semantic Web and allows the creator to specify backwards compatibility, incompatibility, and depreciated classes [33] and could be implemented using the RDDL namespace.

4.3.7 Inheritance
Since a requirement of the Semantic Web is to be able to evolve from schema to schema using rules, microformats must be associated with machine-readable schema specifications using namespaces. hReview is a good example of a microformat in need of schemas and namespaces to support inheritance since it incorporates elements of hCard, hEvent, rel-tag, and rel-license.

We recommend using RDDL as the linking mechanism between a microformat, its XML Schema, and any parent schemas it has inherited from. Since RDDL is machine-readable, the namespace can be recursively parsed to find the specifications of all parent schemas.

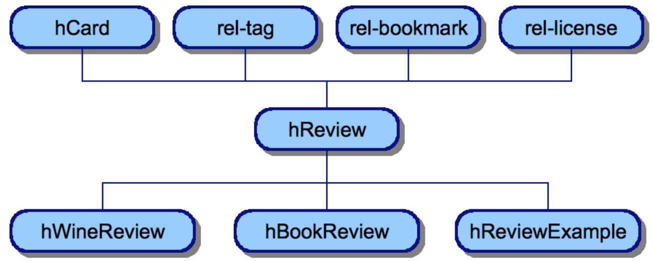

Figure 14. hReview microformat inheritance hierarchy

4.4 Conversion
Gleaning Resource Descriptions From Dialects of Languages (GRDDL) specification is a W3C Recommendation for mechanisms that bootstrap RDF content from uniform XML dialects [61]. The specification introduces markup with namespace-qualified attributes that link to algorithms required to extract the document content and convert it into RDF [61]. GRDDL provides a limited way to transform microformats into RDF and XML [20] but does so at a very low cost [61,62].

Another technology, hGRDDL, overcomes a limitation of GRRDL, whereby once data has been extracted from the document there is no practical way to associate the RDF metadata with the instances of the source data in the document. hGRDDL preserves the visual-semantic context of the metadata within the original page by producing a derived XHTML file the renders identically to the source file. hGRDDL can be implemented in the browser using XSLT [20].

If certain XML attributes are reserved to define element ID references, RDF can be defined within the XML document itself [63]. XML languages having formal semantics defined in RDF already exist [64] and a unified model for merging XML syntax and RDF semantics has already been proposed [65]. Micromodels are an attempt to bridge certain aspects of microformats and RDF using XSL transformations [14]. The Krextor Extraction Framework extracts RDF triples from XML and RDFa formatted web pages [64].

Without an underlying data model, it is difficult for content authors to verify that their implementations are correct [66]. We propose using Schematron to validate microformat implementations against their corresponding schemas. The first step in the validation process is to transform the microformat instance into XML using XSLT and the microformat's profile. Once it is expressed as XML, Schematron can be used to validate the instance against the appropriate XML Schema.

We suggest using GRDDL first extract the microformat enabled content webpage. This could be implemented within the browser using the XSLT files discovered in the RDDL namespace. The next step is to perform validation on the XML output produced by GRDDL. Once validated, hGRDDL can be run against the original webpage to produce a derived page that is visually identical to the previous one but is now contains metadata expressed using RDFa.

For RDFa content or microformat content now encoded as RDFa, GRDDL or the Krextor Extraction Framework can be used to extract the metadata as RDF. Either process can be completed server side or at the browser level. Note that the previous GRDDL validation technique can also be used to extract RDFa annotated content from the page as XML to validated it against an XML

Schema found in the RDDL namespace. This completes the process of converting microformats to the Semantic Web technology RDF.

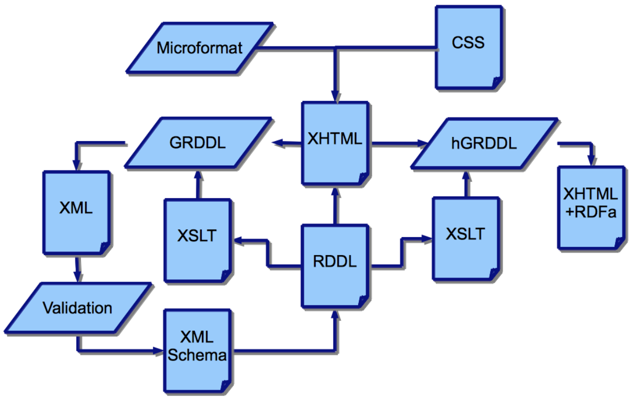

Figure 15. Microformat conversion and validation process

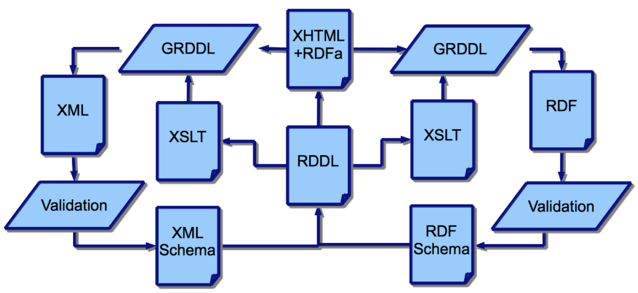

Figure 16. RDFa conversion and validation process

4.5 Ontologies

The last step required to bring microformats into the Semantic Web is to utilize their schemas to create a machine readable ontology. The main obstacle to ontology building is that of schema integration, the process of integrating multiple independently developed schemas to create a global view [67].

We propose a community directory to promote the automatic discoverability of microformats and data that is marked up using microformats thus supporting the goal of the Semantic Web. This directory would ensure that everyone's definition of a particular microformat is the same and would support the discoverability of new and inherited microformats. The inclusion of microformat schemas in the directory would allow for a proper ontology could be created.

The directory would have to be machine and human readable using web standards as much as possible as well as being authoritative and universally recognized. This directory of global microformat schemas and namespaces would also contain translation information for other Semantic Web technologies such as RDF. Global schemas should meet the criteria of completeness, minimalist, and understandability [38].

Methods exist for converting XML Schemas into OWL Ontologies [29]. OWL ontologies can be generated automatically from XML instance data [29] and XML Schemas can be automatically bound to OWL ontologies using reasoners [68]. However, the main issue that make mappings from XML to RDF difficult stems from the fact that XML Schema lacks the expressive power of OWL [7].

The Piazza system enables the interoperability of XML data with RDF data that is linked to an OWL ontology [7]. Piazza uses XML Schema and XML Query (XQuery) to provide the semantic mediation between disparate data sources in a decentralized, peer-to-peer way [7]

We propose using the Piazza system to create a global microformat schema directory. Additionally we propose using XQuery to convert the XML data into OWL format [69]. The schema and RDDL namespace of a particular microformat should support schema integration by providing the information required to link it a global schema to the directory. This completes the low cost conversion to the Semantic Web.

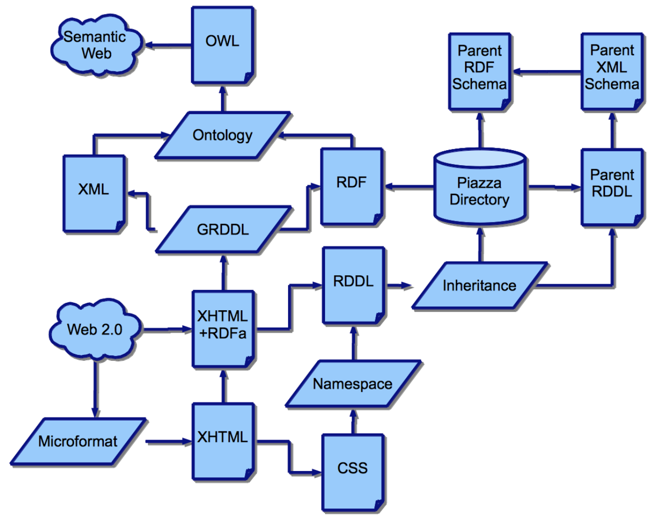

Figure 17. The Low Cost Semantic Web conversion overview

5. CONCLUSION

Microformats are the best way of bringing semantics to Web 2.0 content because of their ease of implementation using CSS methodology, low overhead, and XHTML backwards compatibility. Microformats should be used as a stepping-stone to RDF and the Semantic Web.

5.1 Microformats

Although, not standards, microformats provide a common grammar for specific scenarios. Using microformats that are defined by schemas that provide facilities for validation, data typing, name spaces, and inheritance can eliminate structural conflicts among multiple websites. Microformats support the Semantic Web by imposing a grammar that is readable by intelligent agents. Both the creation and implementation of microformats are decentralized just like Tim Berners-Lee envisions.

The niche of microformats in the Semantic Web is where a universal, decentralized, structure is needed to define meaningful content. Microformats alleviate heterogeneity by being community driven and reusing established technical specifications as much as possible. They limit semantic barriers to information sharing by providing profiles with metadata descriptions. Microformats can be made machine-readable by imposing a strongly-typed XML Schema.

5.2 Semantic Web

Like the Semantic Web, microformats are guided by the principle of minimalist design. Microformats address the challenge of exporting existing content to the Semantic Web by providing an easy to implement, backwards-compatible lexicon of reusable, standards-based grammar. We propose that microformats and RDFa backed by XML Schema as a low cost language for semantically expressing data.

$$XML + RDF + OWL = Semanic\ Web$$

Figure 18. Original formula for the Semantic Web

$$Microformats + XML\ Schema + RDF\,(a) + OWL = Semanic\ Web$$

Figure 19. Proposed formula for the Semantic Web

Proof of concepts exist for generating ontologies and RDF from XML using technologies such as GRDDL and XSLT. This allows RDF and OWL to be repositioned as the languages of choice to provide inference rules about the data. Microformats fit into the data layer of the Semantic Web while the schema directory would be found in the schema layer. This change impacts the current Semantic Web layer cake only slightly.

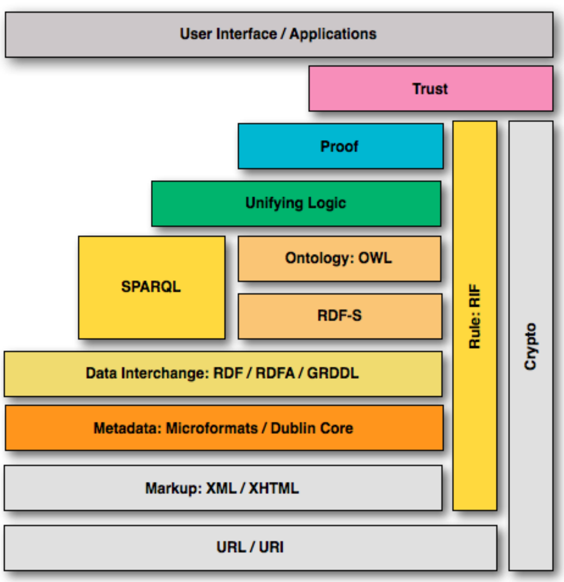

Figure 20. An improved Semantic Web layer cake [70]

5.3 Future Work

This paper presents a theoretical framework for converting microformats into the beginnings of the Semantic Web. The next step is to empirically validate this method. The conversion method proposed incorporates many steps and several difference technologies therefore research into improving the efficiency of the conversion process should be performed.

Research into lowering the barriers and costs of the Semantic Web should continued to be studied. Automation techniques should be researched to determine if the process of applying microformats to unstructured content can be accomplished with minimal user intervention. Additionally, the microdata features present in the HTML 5 working draft should be explored to determine how semantic data should be migrated to the next version of HTML [67].

6. REFERENCES

[1] J. Alpert and N. Hajaj, "We knew the web was big...," *The Official Google Blog2009*, Jul. 2008.
[2] Knowledge@Wharton, "What's the Next Big Thing on the Web? It May Be a Small, Simple Thing," *Knowledge@Wharton*, Feb. 2009.
[3] R. Khare, "Microformats: the next (small) thing on the semantic Web?," *Internet Computing, IEEE*, vol. 10, 2006, pp. 68-75.
[4] E. Tonkin, "A Discount Approach to the Semantic Web," *SWAP*, Deutsche Bibliothek, 2006.
[5] P. Mika, "Microsearch: An Interface for Semantic Search," *SemSearch*, 2008.
[6] N. Shadbolt, W. Hall, and T. Berners-Lee, "The semantic web revisited," *IEEE Intelligent Systems*, vol. 21, 2006, pp. 96-101.
[7] A.Y. Halevy, Z.G. Ives, P. Mork, and I. Tatarinov, "Piazza: data management infrastructure for semantic web applications," 2003.
[8] T. O'Reilly, "Web 2.0 Compact Definition: Trying Again," Dec. 2006.
[9] P. Miller, "Web 2.0: Building the New Library," *Ariadne*, Oct. 2005.
[10] A. Shakya and H. Takeda, "A Report on Linked Data."
[11] T. Berners-Lee, "Linked Data - Design Issues," Jul. 2006.
[12] T. Çelik, "XMDP: Introduction and Format Description," Apr. 2009.
[13] A. Graf, *RDFa VS. Microformats*, Digital Enterprise Research Institute, 2007.
[14] L. Quin, "Microformats: Contaminants or Ingredients? Introducing MDL and Asking Questions," *Talk delivered*, vol. 10, 2006.
[15] microformats.org, "Elemental Microformat," Apr. 2009.
[16] Microformats.org, "rel="tag" Draft Specification," Feb. 2009.
[17] R. Khare and T. Çelik, "Microformats: a pragmatic path to the semantic web," 2006.
[18] J. Allsopp, "The Big Picture on Microformats," *Digital Web Magazine*, Mar. 2009.
[19] B. Gates, "Remarks by Bill Gates," Mar. 2009.
[20] B. Adida, "hGRDDL: Bridging microformats and RDFa," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, 2008, pp. 54-60.
[21] E. Méndez, L. López, A. Freelance, A. Siches, A. Bravo, and S. Webposible, "DCMF: DC & Microformats, a Good Marriage," p. 141.
[22] M.J. de Smith, M.F. Goodchild, and P.A. Longley, *Geospatial Analysis*, Troubador Publishing, 2007.
[23] F.C. Flores, V. Quint, and I. Vatton, "Templates, microformats and structured editing," 2006.
[24] T. Çelik, E. Meyer, and M. Mullenweg, "XHTML meta data profiles," ACM New York, NY, USA, 2005, pp. 994-995.
[25] J. Craig, "hAccessibility - The Web Standards Project," Apr. 2007.
[26] A. Soylu, S. Kuru, F. Wild, and F. Mödritscher, "E-Learning and Microformats: A Learning Object Harvesting Model and a Sample Application," *Mash-Up Personal Learning Environments (MUPPLE'08)*, 2008, p. 57.
[27] B. Suda, "Microformat Cheatsheet," Mar. 2009.
[28] M.P. Schapranow, *Microformats: A XML-document-immanent way for semantic annotations of web contents*, 2006.
[29] H. Bohring and S. Auer, "Mapping xml to owl ontologies," *Leipziger Informatik-Tage*, vol. 72, 2005, pp. 147-156.
[30] P. Walmsley, *Definitive XML Schema*, Prentice Hall, 2001.

[31] S. Powers, *Practical RDF*, O'Reilly & Associates, Inc. Sebastopol, CA, USA, 2003.

[32] C. Sperberg-McQueen and E. Miller, “On mapping from colloquial XML to RDF using XSLT,” 2004, pp. 2-6.

[33] H. Halpin, “XMLVS: Using Namespace Documents for XML Versioning,” *IDEAlliance, editor, Extreme Markup Languages*, 2006.

[34] D. Lee and W.W. Chu, “Comparative analysis of six XML schema languages,” *SIGMOD Rec.*, vol. 29, 2000, pp. 76-87.

[35] G.J. Bex, F. Neven, and J.V.D. Bussche, “DTDs versus XML schema: a practical study,” 2004.

[36] E. Van der Vlist, *RELAX NG*, O'Reilly Media, Inc., 2003.

[37] J. Cardoso and MyiLibrary, *Semantic Web Services Theory, Tools and Applications*, Idea Group Pub., 2007.

[38] K. Passi, L. Lane, S. Madria, B. Sakamuri, M. Mohania, and S. Bhowmick, “A Model for XML Schema Integration,” *E-Commerce and Web Technologies*, 2002, pp. 193-202.

[39] E. Van der Vlist, *XML schema*, O'Reilly Media, Inc., 2002.

[40] J. Hunter and C. Lagoze, “Combining RDF and XML schemas to enhance interoperability between metadata application profiles,” 2001.

[41] J. Borden, “Resource Directory Description Language,” *The Open Healthcare Group and Antarctica Systems*, 2002.

[42] E.R. Harold and W.S. Means, *XML in a nutshell*, O'Reilly Media, Inc., 2002.

[43] Google, “Reviews - Webmasters/Site owners Help.”

[44] V. Tomberg and M. Laanpere, “RDFa versus Microformats: Exploring the Potential for Semantic Interoperability of Mash-up Personal Learning Environments.”

[45] P. Mika, J. Atserias, M. Ciaramita, and H. Zaragoza, “Exploiting Explicit and Implicit Semantics on the Web,” *The Future of Web Search*, Grandvalira, Andorra: 2008.

[46] P. Le Hegaret and I. Jacobs, “Frequently Asked Questions (FAQ) about the future of XHTML,” Jul. 2009.

[47] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Scientific American*, vol. 284, 2001, p. 34.

[48] T. Berners-Lee, “Semantic web road map,” 1998.

[49] N.F. Noy and M. Klein, “Ontology Evolution: Not the Same as Schema Evolution,” *Knowledge and Information Systems*, vol. 6, 2004, pp. 428-440.

[50] K. Weller, “Folksonomies and Ontologies. Two New Players in Indexing and Knowledge Representation,” 2007, pp. 108-115.

[51] B. Amann and I. Fundulaki, “Integrating Ontologies and Thesauri to Build RDF Schemas,” *Research and Advanced Technology for Digital Libraries*, 1999, pp. 672-672.

[52] H. Alani, W. Hall, K. O'Hara, N. Shadbolt, M. Szomszor, and P. Chandler, “Building a Pragmatic Semantic Web,” *Intelligent Systems, IEEE*, vol. 23, 2008, pp. 61-68.

[53] semanticweb.org, “HReview,” Jun. 2009.

[54] D. Brickley and R. Guha, “RDF Vocabulary Description Language 1.0: RDF Schema,” Apr. 2009.

[55] T. Çelik, B. Gibson, and T. Inkster, “XMDP Brainstorming,” Apr. 2009.

[56] T. Agulnick, “MicroFormats: What’s their problem?,” Apr. 2009.

[57] “ISO Schematron,” 2009.

[58] Microsoft, “Improving XML Document Validation with Schematron.”

[59] W3C, “CSS 2.1 Specification: Syntax and basic data types,” Sep. 2009.

[60] T. Inkster, “RDF Extracted Attributes from Styled Elements (RDF-EASE),” Jan. 2009.

[61] R. Cover, “W3C GRDDL Recommendation Bridges HTML/Microformats and the Semantic Web,” *Cover Pages*, Mar. 2009.

[62] F. Bry, T. Furche, A. Hang, and B. Linse, “GRDDLing with Xcerpt: Learn one, get one free!.”

[63] P.F. Patel-Schneider and J. Siméon, “Building the Semantic Web on XML,” 2002.

[64] C. Lange, “Krextor – An RDF →XML Extraction Framework that Strenghtens the Layer Cake,” 2009.

[65] P.F. Patel-Schneider and A. Simeon, “The Yin/Yang Web: a unified model for XML syntax and RDF semantics,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 15, 2003, pp. 797-812.

[66] T. Heath, “Microformat Authoring Not Necessarily Easy,” Apr. 2009.

[67] E. Rahm and P. Bernstein, “On matching schemas automatically,” *VLDB Journal*, vol. 10, 2001, pp. 334-350.

[68] M. Ferdinand, C. Zirpins, and D. Trastour, “Lifting XML Schema to OWL,” *Web Engineering*, 2004, pp. 776-777.

[69] C. Wang, J. Lu, and G. Zhang, “Generation and Matching of Ontology Data for the Semantic Web in a Peer-to-Peer Framework,” *Advances in Data and Web Management*, 2007, pp. 136-143.

[70] W3C, “W3C Semantic Web Activity,” Apr. 2009.