

Carsten M. Schulze

Mikroformate

Was sind Mikroformate und wozu braucht man sie?

Wer schon einmal etwas von Mikroformaten gehört hat, der denkt wahrscheinlich an die durch den Web 2.0-Hype bekannt gewordenen Microformats¹, über die ich später noch sprechen werde. Nun, um es vorwegzunehmen: Nicht alle Mikroformate sind Microformats, jedoch sind alle Microformats Mikroformate. Um es anders auszudrücken: Es gibt neben den bekannten Microformats eine ganze Reihe von Mikroformaten, wie Microdata, eRDF, RDFa, COinS und unAPI mit unterschiedlichen Ursprung und Zweck.

Dieser Umstand macht eine Definition von Mikroformaten schwierig, aber alle Mikroformate haben gemein, dass sie mittels semantischer Auszeichnung von Daten in (X)HTML für ihren jeweiligen Bereich Probleme lösen. Genauer nutzen Mikroformate die (X)HTML-Infrastruktur, um Daten für die jeweilige Wissensgemeinschaft semantisch zu annotieren.

Der Zweck der meisten Mikroformate ist es, Datenfragmente in (X)HTML-Dokumenten derart zu beschreiben und zusammenzuführen, dass die Daten interoperabel werden, d.h. zwischen verschiedenen Systemen, möglichst ohne Informationsverlust, ausgetauscht werden können. Mikroformate, die dem Zweck der Datenformatierung dienen, erleichtern das Arbeiten im Web und damit die Mensch-Maschine-Kommunikation.

Andere Mikroformate sind nur für Maschinen gedacht, wie z.B. das Microformat `rel="nofollow"`², das Vorschläge an Robots von Suchmaschinen macht, die die Verfolgung von Hyperlinks betrifft oder das unAPI-Mikroformat³, das das reine Vorhandensein und den Pfad eines unAPI-Servers anzeigt:

```
<link rel="unapi-server" type="application/xml"
title="unAPI" href="http://example.com/unAPI/">
```

Lower Case Semantic Web

Das Lower Case Semantic Web oder auch das Semantic Web von Unten beschreibt die Bestrebungen, Daten im Web durch Mikroformatierungen für Ma-

1 <http://microformats.org>

2 <http://de.wikipedia.org/wiki/Nofollow>

3 <http://unapi.info/specs/unapi-revision-3.html>

schinen nutzbar zu machen. Im Gegensatz zum Upper Case Semantic Web, also dem „richtigen“ Semantic Web, ist das Lower Case Semantic Web nicht unabhängig von Webseiten, HTML, CSS, Browsern etc.

Mikroformate bedürfen nicht einer komplexen RDF/XML-Syntax, sondern bleiben im Rahmen von (X)HTML.⁴ Im Gegensatz zum Upper Case Semantic Web sind hier die „Konsumenten“ in erster Linie Menschen, die mittels ihres Browsers mit den Mikroformaten interagieren. Dabei können CSS-Formatierungen den Menschen helfen, mikroformatierte Daten zu entdecken.

Der Erfolg von Mikroformaten hängt zum einen unmittelbar damit zusammen, dass Mikroformate relativ einfach zu verstehen und durch Autoren von Webseiten leicht anzuwenden sind. Zum anderen erkennen die Nutzer dieser Websites schnell den Nutzen und die Vorteile von Mikroformaten und lernen diese schätzen.

Microformats

Hat man den Vorteil von Mikroformaten erkannt und möchte man seinen Nutzern die Nachnutzung seiner Daten erlauben, dann steht die Frage im Raum, welches Mikroformat zu verwenden sei.



Abbildung 7: Microformats Logo

Wie schon oben erläutert, gibt es verschiedene Mikroformate für unterschiedliche Zwecke. Bei Daten wie z.B. Namen, Adressen oder Termine, die häufig auf Webseiten auftauchen, bieten die oben erwähnten Microformats ein umfangreiches Reservoir an Datenformatierungen an. Bekannte Microformats sind z.B. hCard⁵ für Namen, Kontakt- und Adressinformation und hCalendar⁶ für Termine.

Um die Interoperabilität mit verbreiteter Software zu gewährleisten, versuchen viele Mikroformate etablierte Datenformate zu imitieren und übernehmen dazu deren Vokabular. Das Microformat hCard ist z.B. dem Standard „vCard

4 Dennoch können umfangreiche Mikroformate eine komplexe Struktur annehmen.

5 <http://microformats.org/wiki/hcard>

6 <http://microformats.org/wiki/hcalendar>

MIME Directory Profile⁷ nachempfunden, welches schon lange das Standardaustauschformat vieler elektronischer Adressbücher ist.

Standard VCARD:

```

▪ BEGIN:VCARD
▪ FN:Mr. Carsten M. Schulze
▪ ORG:Staatsbibliothek zu Berlin
▪ ADR:Potsdamer Straße 33;Berlin;10785;Deutschland
▪ TEL;WORK;VOICE:+49-30/266-435781
▪ EMAIL;Internet:carsten.schulze@sbb.spk-berlin.de
END:VCARD
Microformat hCard:
<div class="vcard">
  <span class="fn n">
    <span class="given-name">Carsten</span>
    <span class="additional-name">M.</span>
    <span class="family-name">Schulze</span>
  </span>
  <div class="org">Staatsbibliothek zu Berlin</div>
  <a class="email" href="mailto:carsten.schulze@sbb.spk-berlin.de">carsten. schulze@sbb.spk-berlin.de</a>
  <div class="adr">
    <div class="street-address">Potsdamer Straße 33</div>
    <span class="country-name">Deutschland</span>
    <span class="postal-code">10785</span>
    <span class="locality">Berlin</span>
  </div>
  <div class="tel">+49-30/266-435781</div>
</div>

```

An dem Beispiel ist gut zu erkennen, wie Mikroformate aufgebaut sind. Wie schon oben erwähnt, nutzen Mikroformate die (X)HTML-Infrastruktur. In diesen Fall nutzt das Mikroformat hCard die HTML-Elemente DIV, SPAN und A, sowie das @class-Attribut.

Das @class-Attribut des obersten DIV-Elements enthält z.B. die Namenskonvention „*vcard*“, was darauf hinweist, dass die eingebetteten Elemente Kontaktdaten nach dem vCard-Standard enthalten. Und tatsächlich enthalten die @class-Attribute des hCard-Microformats die gleichen Schlüssel wie die vCard: FN (full name), ORG (organization), ADR (address) etc. Analog funktioniert dies auch mit dem Microformat hCalendar, das in Anlehnung an den iCalendar-Standard⁸ entworfen wurde.

Microformats sind zudem so modular aufgebaut, so dass sie leicht kombiniert werden können. Bei Microformats unterscheidet man von vornherein zwischen

7 <http://www.ietf.org/rfc/rfc2426.txt>

8 <http://www.ietf.org/rfc/rfc2445.txt>

elementaren Microformats (elemental microformats) und verbundenen Microformats (compound microformats).

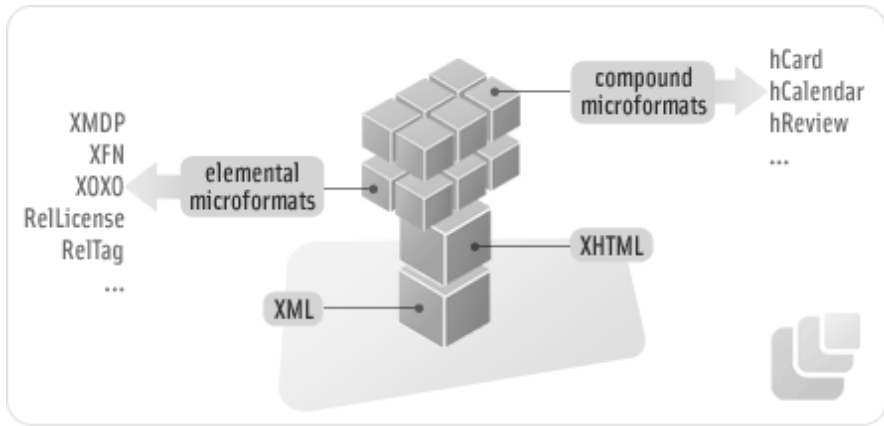

















Abbildung 8: Architektur von Microformats

Abschließend sollen hier noch einmal die wichtigsten Microformats aufgelistet werden:

 HCARD	hCard: Kontaktinformation
 HICAL	hCalendar: Termine
 HREVIEW	hReview: Berichte
 HATOM	hAtom: Syndikation von News
 HRESUME	hResume: Lebensläufe
 LICENSE	rel-license: Lizenzinformation
 NOFOLLOW	rel-nofollow: Anweisungen für Robots
 REL TAG	rel-tag: Verschlagwortung
 VOTELINKS	VoteLinks: Abstimmungen
 XFN	XFN: soziale Beziehungen
 XOXO	XOXO: Listen
 XMDP	XMDP: Metadatenprofile
 XFOLK	xFolk: Blog-Posts
 GEO	geo: Geoinformation
 ADR	adr: Adressinformation (z.B. in hCard)

Screen Scraping und Tools

Der direkte Mehrwert, den Mikroformate versprechen, liegt in der Interaktion mit dem Nutzer, der mittels des Browsers (Maschine) mit den Daten arbeiten kann (Screen Scraping). Browser benötigen dazu allerdings kleine Helfer-Programme, wie z.B. das Firefox-Add-On „Operator“⁹. Mit dem Microformat hCard ausgezeichnete Daten können so im Handumdrehen in das eigene elektronische Adressbuch kopiert werden.

Neben dem Firefox Add-On Operator gibt es allerdings noch eine ganze Reihe weiterer Tools und Services, mit denen Microformats zu erstellen sind oder die mit Microformats arbeiten können. Eine Liste von Tools ist am Ende dieses Kapitels zu finden.

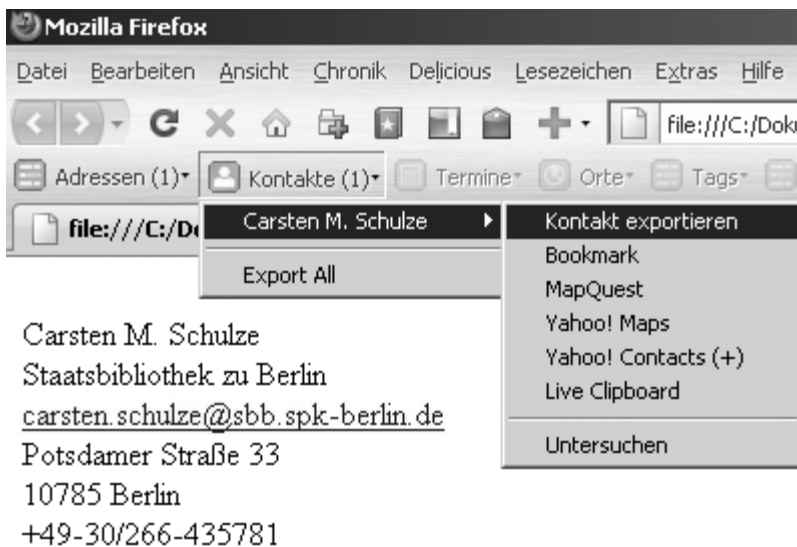


Abbildung 9: Extraktion von Kontaktdaten mittels Firefox Add-On "Operator"

COinS und unAPI

COinS

Literaturverwaltungssysteme sind ein unerlässliches Hilfsmittel zur Erstellung einer wissenschaftlichen Arbeit, und der Online-Katalog ist die primäre Quelle für die benötigten bibliographischen Daten. Bibliotheken sollten ihren Nutzern nun nicht nur den Service bieten, ihre Kontaktdaten oder Termine mit Mikroformaten

⁹ <http://kaply.com/weblog/operator>

zu versehen, sondern auch ihre bibliographischen Daten können sie dem Nutzer nun zur einfachen Nachnutzung anbieten.

Die Mikroformate COinS und unAPI sind dabei sinnvolle Ergänzungen zu der Z39.50-Schnittstelle oder dem Download der Daten in BibTeX10 oder RIS11. COinS12 steht für OpenURL ContextObject in SPAN und dient dazu, eine OpenURL in HTML einzubetten. Eine OpenURL (Z39.88)13 ist ein standardisiertes Datenformat, das zumeist in Form einer URL beschrieben und im Bibliotheksbereich vor allem in durch das Produkt SFX¹⁴ bekannt geworden ist.

Ein OpenURL besteht aus zwei Teilen: dem Basis-URL und dem ContextObject, welches dazu dienen kann, bibliographische Daten zu beschreiben.

Zeitschriftenaufsatz:

Chudnov (2005): Opening up OpenURLs with Autodiscovery. Erschienen in: Ariadneinfo (ISSN 1361-3200), Heft 43, April 2005.<http://www.ariadne.ac.uk/issue43/chudnov/>

ContextObject:

```
ctx_ver=Z39.88-2004
&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.title=Ariadne
&rft.aulast=Chudnov
&rft.atitle=Opening+up+OpenURLs+with+Autodiscovery
&rft.issue=43
&rft.issn=1361-3200
&rft.date=2005-04
&rft_id=http://www.ariadne.ac.uk/issue43/chudnov/
```

Um dieses OpenURL ContextObject nun auf einer Webseite seinen Nutzern zur Nachnutzung der Daten zur Verfügung zu stellen, bedient sich COinS der schon erprobten Praxis der Microformats. Ähnlich wird also das ContextObject in die Website eingebunden:

COinS:

```
<span class="Z3988" title="ctx_ver=Z39.88-2004
&amp;rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&amp;rft.title=Ariadne
&amp;rft.aulast=Chudnov
&amp;rft.atitle=Opening+up+OpenURLs+with+Autodiscovery
&amp;rft.issue=43
```

10 <http://de.wikipedia.org/wiki/BibTeX>

11 [http://en.wikipedia.org/wiki/RIS_\(file_format\)](http://en.wikipedia.org/wiki/RIS_(file_format))

12 <http://ocoins.info/>

13 National Information Standards Organization (NISO): The OpenURL Framework for context-sensitive services. An American national standard. 2004. Aufl. (2005). Bethesda, Md.: NISO Press (ANSI/NISO, ANSI/NISO Z39.88). ISBN: 1-88012-461-0.

14 <http://www.exlibrisgroup.com/category/SFXOverview>

```
&amp;rft.issn=1361-3200&amp;rft.date=2005-04
&amp;rft_id=http://www.ariadne.ac.uk/issue43/chudnov/">&nbsp;
</span>
```

Wie der Name es schon sagt, nutzt COinS das HTML-Element SPAN, sowie die Attribute @class und @title. Das Attribut @class enthält analog zu den Microformats eine Konvention (Z3988), die einem Programm den Hinweis darauf geben kann, um welches Datenformat es sich hier handelt. Das Attribut @title enthält das ContextObject mit den Schlüsseln und Werten des Datensatzes.

Wenn auch noch nicht sehr bekannt, ist COinS in Bibliothekswesen schon länger im Einsatz. COinS ist beispielsweise im WorldCat¹⁵ und im GVK¹⁶, um bekanntere Beispiele zu nennen, implementiert. Auch der Web 2.0-Katalog Vu-Find¹⁷ und die Metasuchmaschine iPort¹⁸ von OCLC unterstützen inzwischen COinS.

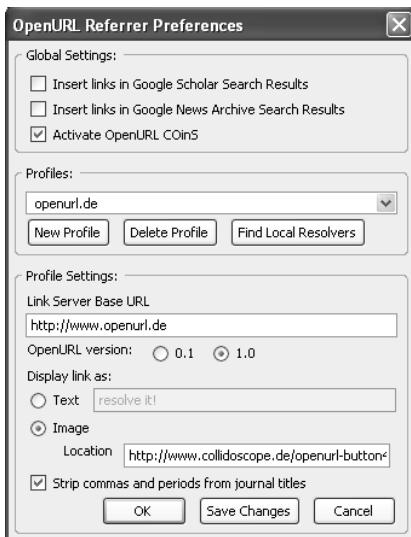


Abbildung 10: Einstellungen im Firefox Add-On OpenURL-Referrer

Ebenso unterstützen Literaturverwaltungssysteme wie z.B. Citavi, Zotero¹⁹, Ref-Base²⁰ und Mendeley²¹ das Mikroformat COinS oder sind in der Lage, ein ContextObject einer OpenURL zu verarbeiten.

15 <http://www.worldcat.org>

16 <http://gso.gbv.de>

17 <http://vufind.org>

18 <http://oclc.org/?id=1078&ln=de>

19 <http://www.zotero.org>

20 <http://refbase.sourceforge.net>

21 <http://www.mendeley.com>

Eine weite Anwendungsmöglichkeit bezieht sich auf die eigentliche Stärke von OpenURL. Das Firefox Add-On OpenURL-Referrer²² bedient sich COinS und verbindet das ContextObject mit einer frei wählbaren Basis-URL, um somit wieder eine vollständige OpenURL zu generieren. Dieser OpenURL kann dann unter dem Basis-URL²³ von einem OpenURL-Resolver aufgelöst werden und verschiedene Services (z.B. die Konvertierung des ContextObjects in andere Formate) können angeboten werden.

unAPI

Auch wenn unAPI²⁴ mehr als ein Mikroformat ist, dient es jedoch dem gleichen Zweck: Es soll Daten auf Webseiten derart beschreiben, dass Maschinen die Daten sinnvoll verarbeiten können. unAPI bedient sich jedoch nicht der Daten, die auf der Webseite zu finden sind, wie es bei den anderen Mikroformaten der Fall ist, sondern legt eine Fährte, um weitergehende Information für Maschinen auffindbar zumachen. Die Daten die letztendlich der Nutzer bekommt stammen aus anderen Quellen, wie z.B. einer Datenbank.

Um dies zu bewerkstelligen, besteht unAPI aus drei Komponenten:

Ein Identifier Mikroformat
 Der Verweis auf den unAPI-Server
 Der unAPI-Server

Das Identifier Mikroformat bedient sich des HTML ABBR Elements mit dem Attribut `@class`, dessen Wert die Konvention „unapi-id“ darstellt, und dem Attribut `@title`, dessen Wert ein eindeutiger Identifier eines Datensatzes darstellt.

unAPI Identifier Mikroformat:

```
<abbr class="unapi-id" title=" isbn:1590598148"></abbr>
```

Der Verweis auf den unAPI-Server, wie ich ihn am Anfang dieses Kapitels schon einmal vorgestellt hatte, befindet sich im HEAD-Bereich der HTML-Seite. Er besteht aus dem HTML Element LINK und dem Attribut `@rel`, dessen Wert die Konvention „unapi-server“ ist, dem Attribut `@type` mit dem Wert „application/xml“, dem Attribut `@title`, dessen Wert die Konvention „unAPI“ ist und dem Attribut `@href`, dessen Wert der URI des unAPI-Servers darstellt.

Verweis auf den unAPI-Server:

```
<link rel="unapi-server" type="application/xml"
title="unAPI" href="http://example.com/unAPI/">
```

22 <http://www.openly.com/openurlref>

23 Ein Basis-URL mit OpenURL-Resolver ist z.B. <http://www.openurl.de>

24 <http://unapi.info>

Zuguterletzt braucht es noch den unAPI-Server, der eigentlich nur ein kleines Script sein muss, um die Aufgabe des Content Negotiation zu übernehmen. Content Negotiation meint in diesem Zusammenhang, dass der Client sich mit dem unAPI-Server „unterhält“ und die beiden unter sich die Übergabekriterien für die angeforderten Daten klären (siehe Zeichnung 1).

unAPI hat einen höheren Implementierungsaufwand als andere Mikroformate. Dafür sind die Möglichkeiten, die mit unAPI entstehen, ungleich größer, denn das Datenformat kann mit unAPI zwischen Client und Server ausgehandelt werden, die beide allein die Grenzen des Machbaren bestimmen. So können über unAPI z.B. ganze digitale Objekte wie z.B. Videofiles, METS-Dokumente oder sogar Software vom unAPI-Server an den Client übergeben werden. Viele Projekte, wie z.B. der GBV²⁵, WikiD²⁶ oder RefBase, verfügen sogar schon über unAPI. Auf der Seite der Literaturverwaltungssysteme ist es allerdings bisher nur mit Zotero möglich, über unAPI bereit gestellte Daten zu importieren.

Zeichnung 1: Content Negotiation zwischen einem Client und einem unAPI-Server

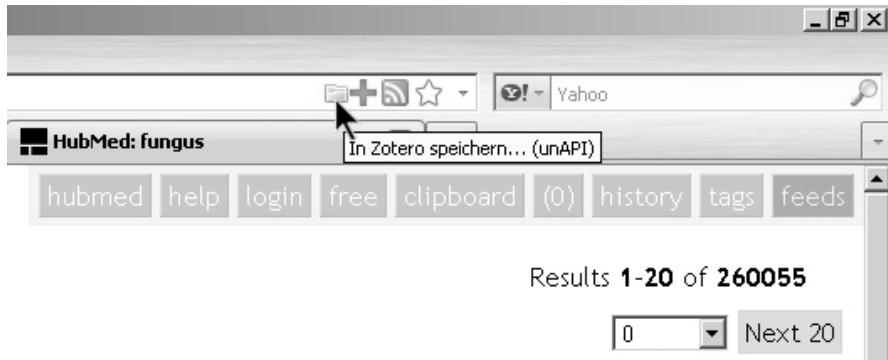


Abbildung 11: Zotero Import von Titeln mittels unAPI

RDFa und Microdata

RDFa

RDFa²⁷ (RDF attributes) ist eine W3C-Recommendation und hat das Ziel, die Aussagekraft von RDF in XHTML einzubetten, so dass aus diesem XHTML+RDFa RDF extrahiert²⁸ werden kann. XHTML+RDFa ist XHTML, das

25 <http://unapi.gbv.de>

26 <http://alcme.oclc.org/wikid/WikiDApis>

27 <http://www.w3.org/TR/rdfa-syntax>

28 Dies kann z.B. durch GRDDL geschehen. Siehe dazu <http://www.w3.org/TR/grddl>

um ein paar Attribute erweitert wurde. XHTML+RDFa hat seine eigene DTD²⁹ und hat folgenden Dokumententyp:

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
```

Die neuen Attribute, die RDFa zur Verfügung stehen, sind:

@about

Ein URI oder CURIE³⁰ bildet das Subjekt der Aussage ab.

@property

Eine durch Leerzeichen getrennte Liste von CURIEs bildet die Beziehung zwischen dem Subjekt und dem Objekt in Form eines Literals als Prädikat der Aussage ab.

@resource

Ein URI oder CURIE bildet das Objekt der Aussage ab.

@datatype

Ein CURIE bestimmt den Datentyp eines Literals.

@typeof

Eine durch Leerzeichen getrennte Liste von CURIEs, die die RDF-Typen des Subjekts beschreibt und einen leeren Knoten erzeugen kann.

Weitere Attribute aus dem XHTML-Namespaces, die RDFa nutzt, sind:

@rel

Eine durch Leerzeichen getrennte Liste von CURIEs bildet die Beziehung zwischen zwei Ressourcen als Prädikat der Aussage ab.

@rev

Eine durch Leerzeichen getrennte Liste von CURIEs bildet die inverse Beziehung zwischen zwei Ressourcen als Prädikat der Aussage ab.

@name

Enthält Werte in Form einer Zeichenkette.

@content

Enthält eine Zeichenkette in maschinenlesbarer Form.

@href

Ein URI bildet das Objekt der Aussage ab.

@src

Ein URI bildet das Objekt der Aussage ab, wenn dieses im Dokument eingebettet ist.

Der große Vorteil von RDFa ist es, verschiedene Vokabulare einbinden zu können und diese auch kombinieren zu können. RDFa ist damit nicht auf ein paar Datenformate festgelegt. Jedem steht es frei sein eigenes Vokabular zu erstellen, auch wenn bei RDFa der Grundsatz gilt, schon vorhandene und viel genutzte Vokabulare primär zu verwenden.

29 http://www.w3.org/TR/rdfa-syntax/#a_xhtmlrdfa_dtd

30 compact URIs

RDFa:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
  "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<htmlxmlns="http://www.w3.org/1999/xhtml"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:bio="http://vocab.org/bio/0.1/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xml:lang="en">
<head profile="http://www.w3.org/1999/xhtml/vocab">
[... ]
</head>
<body>
<div typeof="foaf:Person">
  <span property="foaf:givenname">Carsten</span>
  <span property="foaf:family_name">Schulze</span>
  <span rel="foaf:homepage"
resource="http://www.collidoscope.de/">
  http://www.collidoscope.de/</span>
  <span property="bio:Birth" datatype="xsd:date">
  1975-09-01</span>
</div>

```

Dieses RDFa-Beispiel entspricht in etwa dem hCard Microformat und verwendet die Vokabulare FOAF und bio, sowie RDF- und XSD Sprachelemente. Ein RDFa-Parser würde folgende Triple aus dem Code extrahieren:

RDF Turtle-Syntax:

```

@prefix foaf:      <http://xmlns.com/foaf/0.1/> .
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix bio:      <http://vocab.org/bio/0.1/> .
_:a      rdf:type      foaf:Person ;
foaf:givenname      "Carsten" ;
foaf:family_name    "Schulze" ;
foaf:homepage<http://www.collidoscope.de/> ;
bio:Birth           "1975-09-01"^^xsd:date .

```

Die RDFa-Syntax scheint etwas komplizierter zu sein als die der Microformats, was daran liegt, dass HTML nicht dafür entwickelt wurde, RDF zu transportieren. RDFa verfolgt also mehr die Verbreitung des Semantic Web, anstatt sich an kleinen Copy&Paste-Problemen aufzuhalten. Das bedeutet allerdings nicht, dass RDFa nicht auch für andere Problemstellungen eine Lösung bieten kann.³¹

31 Siehe hierzu <http://rdfa.info/wiki/Rdfa-use-cases>

Microdata

Microdata ist das jüngste Kind in der Mikroformat-Familie, und eigentlich ist es zum Erstellungszeitpunkt dieses Kapitels noch gar nicht geboren. Da Mikrodata im Zusammenhang mit HTML5³² entwickelt wird und dieser Prozess noch nicht abgeschlossen ist, liegt mir aktuell nur der W3C Working Draft³³ vor. Dennoch sollte Mikrodata in diesem Kapitel nicht fehlen, denn es vereint die Stärken von Mikroformats (Einfachheit) und RDFa (Ausdrucksfähigkeit).

Das RDFa-Beispiel würde mit Microdata folgendermaßen aussehen:

Microdata:

```
<!doctype html>
<div itemscope
itemtype="http://xmlns.com/foaf/0.1/Person">
  <span itemprop="givenName">Carsten</span>
  <span itemprop="familyName">Schulze</span>
  <span itemprop="http://vocab.org/bio/0.1/Birth"
  datetime="1975-09-01">01.09.1975</span>
</div>
```

Wie zu erkennen ist, hat Microdata ebenfalls ein paar neue Attribute hinzubekommen, um seine Aufgabe zu erfüllen. Bevor HTML5 jedoch nicht veröffentlicht wurde, dürfte es sinnlos sein, hier neue Attribute aufzulisten, da sich diese täglich ändern könnten.

Festzuhalten bleibt jedoch, dass Microdata eine ernst zu nehmende Alternative für Mikroformats und RDFa darstellen könnte.

Ausblick

Das Thema Mikroformate kommt langsam in den Bibliotheken an. Immer mehr Bibliotheken implementieren COinS und unAPI in ihre Kataloge, da diese Mikroformate ihren Nutzern eine verbesserte Nutzung ihrer bibliographischen Daten garantieren. Jedoch sind die Grenzen des Machbaren hier noch lange nicht erreicht.

Andere Mikroformate warten noch auf ihren Durchbruch im Einsatz von Visualisierungen von bibliographischen Daten auf Karten (z.B. das Mikroformat geo³⁴) oder bei der Einbindung von Lizenzinformation mittels RDFa.

Auch wenn die Implementierung und Nutzung von Mikroformaten noch etwas schwierig und kompliziert erscheint, wird, wenn der Bedarf weiterhin beste-

32 Siehe aktuelle Publikation des W3C der HTML5-Spezifikation: <http://www.w3.org/TR/html5>

33 <http://www.w3.org/TR/2010/WD-microdata-20100304>

34 <http://microformats.org/wiki/geo>

hen bleibt, Daten auf Webseiten semantisch zu annotieren, um diese besser für Maschinen nutzbar zu machen, das Thema Mikroformate an Gewicht gewinnen. Unter dieser Voraussetzung könnten auch HTML5 und Microdata die Entwicklung einige Schritte voran bringen.

Liste von Tools

Diese Liste soll einen Überblick über die aktuellen Anwendungen im Bereich Mikroformate geben und zum Ausprobieren und Mitmachen einladen.

Microformats erstellen:

- hCard Creator
Formular zur Erstellung eines hCard Microformats
<http://microformats.org/code/hcard/creator>
- hCalendar Creator
Formular zur Erstellung eines hCalendar Microformats
<http://microformats.org/code/hcalendar/creator>
- hReview Creator
Formular zur Erstellung eines hReview Microformats
<http://microformats.org/code/hreview/creator>
- XFN Creator
Formular zur Erstellung eines XFN Microformats
<http://gmpg.org/xfn/creator>
- Windows Live Writer
Blogging Tool mit Plug-Ins zur Erstellung von hCard und hCalendar Microformats
<http://download.live.com/writer>

COinS Generator Formular zum Erstellen von ContextObjects in Span (COinS)
<http://generator.ocoin.info>

Auslesen und Transformation von Microformats

Operator

Firefox Add-On zur Extraktion und Weiterverarbeitung von Mikroformaten
<http://kaply.com/weblog/operator>

- michromeformats
Erweiterung für den Browser Chrome zur Anzeige verschiedener Microformats
<http://ryckbost.com/blog/archives/2010/04/21/chrome-microformats-michromeformats/>
- Swignition
Umfangreiches Transformationstool für verschiedene Mikroformate
<http://buzzword.org.uk/swignition>
- Oomph
Toolkit zur Extraktion, Transformation und Gestaltung von Microformats
<http://oomph.codeplex.com>

- Optimus-Microformats Transformer
API zur Validierung und Transformation von Microformats in die Datenformate XML, JSON und RSS
<http://microformatique.com/optimus>
- transformr
Extraktion und Transformation verschiedener Mikroformate
<http://transformr.co.uk>
- X2V
Transformation von hCard und hCalendar in die entsprechenden Austauschformate
<http://suda.co.uk/projects/X2V>
- GEO Microformats to XML
Transformation von geo Microformats in verschiedene XML-Dialekte
<http://suda.co.uk/projects/microformats/geo>
- OpenURL-Referrer
Add-On für Firefox und Internet Explorer zur Nutzung von OpenURL ContextObject in Span (COinS)
<http://www.openly.com/openurlref>
- Greasemonkey
Firefox Add-On zum Ausführen von Userscripts. Userscripts stehen für diverse Mikroformate bereit.
<http://www.greasespot.net>
- Talis Export
Firefox Add-On zur Extraktion von Microformats
<https://addons.mozilla.org/de/firefox/addon/2240>
- LibX
Individuell anpassbares Browser Add-On speziell für Bibliotheken mit OpenURL/COinS-Funktionalitäten
<http://www.libx.org>
- Minimap Sidebar
Firefox Add-On zum Auslesen von geo Microformats und Darstellung der Geoinformation auf einer Karte
<http://minimap.spatialviews.com>
- Tabulator
Firefox Add-On zur Visualisierung von RDF-Triples
<http://dig.csail.mit.edu/2007/tab>

Literaturverzeichnis

(Schulze, 2008) Schulze, C.: Mikroformate für bibliographische Daten : Vergleich verschiedener Konzepte zur semantischen Annotation. Potsdam, Fachhochschule Potsdam, 2008