STUDIA INFORMATICA 2007
Volume 28 Number 2 (71)

Michał ŚWIĄTKIEWICZ
Politechnika Śląska, Instytut Informatyki

DANE SIECI SEMANTYCZNEJ W DZISIEJSZEJ SIECI WWW

Streszczenie. Koncepcja sieci semantycznej zakłada opracowanie mechanizmów publikacji informacji umożliwiających agentom programowym korzystanie z danych opublikowanych w Internecie, podobnie jak człowiek korzysta z dzisiejszej sieci WWW. Nadal jednak człowiek pozostanie głównym użytkownikiem sieci. Powoduje to konieczność współistnienia w sieci semantycznej dwóch zasadniczo różnych sposobów publikacji. Artykuł przedstawia stosowane aktualnie sposoby w celu uniknięcia komplikacji.

Słowa kluczowe: Sieć semantyczna, RDF, eRDF, RDFa, microformats, GRDDL

SEMANTIC WEB DATA ON CURRENT WORLD WIDE WEB

Summary. The Semantic Web vision assumes that new ways of publicizing information will be prepared which allow software agents to use data available in the Web just like people use current WWW. However the Web still will be used mainly by people. Thus it is necessary for two completely different ways of publicizing information to coexist together. The article presents methods used today to avoid this complexity.

Keywords: Semantic Web, RDF, eRDF, RDFa, microformats, GRDDL

1. Wprowadzenie

Przy aktualnym stanie rozwoju technologii wymiana informacji za pomocą sieci internetowej, w szczególności WWW, stanowi nieodzowny element funkcjonowania współczesnego człowieka. Internet służy nie tylko jako medium komunikacyjne, ale również największe i najłatwiejsze do wykorzystania źródło informacji. We współczesnej cywilizacji znaczenie tego medium w zastosowaniach informacyjnych, handlowych i marketingowych jest nie do przecenienia.

Niestety, współczesna postać sieci WWW przeznaczona jest praktycznie wyłącznie dla odbiorcy ludzkiego. Mimo istnienia ogromnych zasobów informacji ich odczytanie, przeszukiwanie i połączenie danych z różnych źródeł wymaga bardzo dużego nakładu pracy. Wynika to ze sposobu prezentacji informacji na stronach WWW – stosowania wielu różnych formatów i technik prezentacji trudnych lub wręcz niemożliwych do przetwarzania programowego, różnego słownictwa do opisu tej samej dziedziny pojęciowej w języku naturalnym itd. Powoduje to sytuację, w której odczytanie i powiązanie informacji z różnych źródeł w sieci wymaga ludzkich zdolności rozumienia i kojarzenia faktów, a tym samym informacje w sieci WWW stają się w praktyce niemożliwe do zautomatyzowanego przetwarzania przez agentów programowych. Z drugiej jednak strony różnorodność Internetu, mnogość źródeł informacji i podmiotów publikujących dane stanowi jego największą siłę, i nie jest w praktyce możliwe narzucenie jednego sformalizowanego sposobu prezentacji informacji w sieci.

Próbą rozwiązania takich problemów i umożliwienia dostępu do informacji publikowanych w Internecie agentom programowym jest koncepcja sieci semantycznej (ang. *Semantic Web*)[1]. Jej podstawowe założenia opierają się na opracowaniu specjalnych mechanizmów reprezentacji danych i wiedzy [1] i w tym celu powstał szereg technologii umożliwiających publikację informacji w sieci Internet w sposób możliwy do zautomatyzowanego przetwarzania programowego. Dzięki pracom standaryzacyjnym prowadzonym w ramach konsorcjum W3C[2] języki, takie jak RDF (*Resource Description Framework*) [2, 3, 4] i OWL (*Web Ontology Language*) [5] mają aktualnie postać umożliwiającą już realizację idei *Semantic Web* w praktyce. Ponieważ jednak są to języki opisu wiedzy wraz z odpowiadającymi im sposobami reprezentacji dostosowane do specyficznych wymagań koncepcji sieci semantycznej, konieczna jest konwersja danych przechowywanych i prezentowanych dotychczas w formatach wykorzystywanych przez „klasyczną" postać sieci WWW.

Nie należy jednak przy tym zapominać o powszechności współczesnego Internetu i o tym, że jego użytkownikami są przede wszystkim ludzie. Dzięki zastosowaniu technologii sieci semantycznej jest możliwe powstawanie mechanizmów, ułatwiających wiele zadań związanych z pozyskiwaniem i przetwarzaniem informacji i wiedzy pochodzącej z sieci, jednak podstawowym odbiorcą Internetu, a w szczególności stron WWW, nadal pozostanie człowiek. Wynika stąd potrzeba równoczesnej publikacji tej samej informacji zarówno w postaci czytelnej dla mechanizmów sieci semantycznej, jak również dla „klasycznego" odbiorcy WWW, czyli człowieka.

---

[1] Rozwój koncepcji *Semantic Web* koncentruje się wokół *W3C Semantic Web Activity*, http://www.w3.org/2001/sw/.

[2] World Wide Web Consortium, http://www.w3.org/.

2. Współistnienie danych sieci semantycznej i WWW

Założenia idei *Semantic Web* uzupełnione konkluzją poprzedniego rozdziału oznaczają, iż Sieć semantyczna, w kierunku której nieuchronnie ewoluuje współczesny Internet, będzie w praktyce miała zarówno odbiorców analogicznych do dzisiejszych użytkowników WWW, jak również odbiorców w postaci agentów programowych. Sposób odbioru informacji przez te dwie grupy użytkowników jest zupełnie odmienny. Dlatego też sposób prezentacji informacji dla nich przeznaczonych jest również zupełnie odmienny.

Należy przy tym zauważyć, że ze względu na znaczne różnice w stopniu ekspresji języka naturalnego, jakim są wyrażane informacje na stronach WWW i języków sieci semantycznej, przeznaczonych do przetwarzania programowego, zakres informacji publikowanych równocześnie w obu postaciach obejmuje w praktyce jedynie „suche" fakty, dane lub metadane na temat zasobów sieci. Nie należy na przykład w żadnym przypadku oczekiwać, że walory artystyczne zawarte w wersji strony WWW przeznaczonej dla człowieka zostaną w jakikolwiek sposób przeniesione do jej wersji przeznaczonej dla agentów programowych sieci semantycznej. Jednak cała zawartość informacyjna powinna być dostępna dla obu rodzajów odbiorców strony.

Istnieją dwa możliwe podejścia do problemu równoczesnej publikacji informacji dla sieci WWW i sieci semantycznej:

- publikacja danych w dwóch osobnych dokumentach – w postaci czytelnej dla człowieka (takiej jak język naturalny) i osobno w postaci możliwej do przetwarzania programowego (z wykorzystaniem modelu RDF),
- publikacja jednego dokumentu ze wskazaniem mechanizmu umożliwiającego jego automatyczne przekształcenie z jednej postaci w drugą.

2.1. Osobne dokumenty dla różnych odbiorców

Publikacja różnych dokumentów oznacza konieczność tworzenia osobno strony WWW przeznaczonej dla odbiorcy ludzkiego i osobno pliku reprezentującego model RDF, zawierający te same informacje. Strona WWW zostaje w takim przypadku uzupełniona w części nagłówkowej dokumentu (X)HTML o element `<link>`, wskazujący zserializowaną postać modelu RDF, na przykład:

```
<link rel="meta" type="application/rdf+xml" href="model.rdf" title="RDF" />
```

Agent sieci semantycznej jest zatem niejako „przekierowany" ze strony WWW przeznaczonej dla człowieka na adres, zawierający dokument bardziej dla niego zrozumiały lub dostosowany do specyficznych potrzeb takiego typu odbiorcy.

Ponieważ model danych RDF jest z definicji grafem [2], jego serializacja do postaci tekstowej jest wymagana w celu efektywnego transportu w sieci internetowej. Adres URL tego dokumentu jest wskazany przez atrybut `href`, a typ MIME jego zawartości odpowiadający zastosowanemu formatowi serializacji jest wskazany przez atrybut `type`. Sposobem serializacji grafu RDF oficjalnie stosowanym w publikacjach W3C jest format RDF/XML [6], jednak nie jest to jedyna możliwość zapisu grafu RDF w postaci tekstowej. Do innych szeroko stosowanych formatów należy na przykład N3 (*Notation 3*) [7], który nie jest formatem XML. Tabela 1 przedstawia zestawienie typów zawartości MIME stosowanych dla popularnych formatów serializacji grafów RDF [6, 7, 8, 9, 10].

Tabela 1

Typy zawartości MIME dla różnych formatów serializacji grafu RDF

Format	Typ MIME
RDF/XML	`application/rdf+xml`
N3	`text/rdf+n3`
N-Triples	`text/plain`
Turtle	`application/x-turtle`

Atrybut `rel` wskazuje charakter relacji między dwoma dokumentami i w tym przypadku najczęściej przyjmuje wartość `"meta"`, lecz inne wartości, np. `"alternate"`, mogą być również merytorycznie poprawne [6].

Poniższy przykład przedstawia rzeczywisty fragment strony WWW, opisującej projekt Apache Ant[3] (kod strony został uproszczony), oraz dokument RDF, zawierający te same Informacje opisane z wykorzystaniem ontologii DOAP (*Description Of A Project*)[4]. *The Apache Software Foundation* stosuje ontologię DOAP (oraz dodatkowe własne rozszerzenia) do opisu wszystkich prowadzonych projektów[5].

```
<html . . .>
<head>
<title>Project Details for Apache Ant</title>
<link rel="meta"
href="http://svn.apache.org/repos/asf/ant/core/trunk/doap_Ant.rdf"
type="application/rdf+xml" title="DOAP">
. . .
</head>
<body>
. . .
<h2>Apache Ant</h2>
<p>Apache Ant is a Java-based build tool. In theory, it is kind of like Make,
but without Make's wrinkles.</p>
<p>Programming Languages <a
href="http://projects.apache.org/indexes/language.html#Java">Java</a></p>
<p>Categories <a href="http://projects.apache.org/indexes/category.html#build-
management">build-management</a></p>
<p>Mailing Lists <a
href="http://ant.apache.org/mail.html">http://ant.apache.org/mail.html</a></p>
```

[3] Project Details for Apache Ant, http://projects.apache.org/projects/ant.html.
[4] Description Of A Project, http://usefulinc.com/doap.
[5] The Apache Software Foundation, http://projects.apache.org/.

```
<p>Bug/Issue Tracker <a
href="http://issues.apache.org/bugzilla/buglist.cgi?product=Ant">http://issues.a
pache.org/bugzilla/buglist.cgi?product=Ant</a></p>
<p>License <a href="http://www.apache.org/licenses/LICENSE-2.0">Apache License
Version 2.0</a></p>
<p>Project Website <a
href="http://ant.apache.org/">http://ant.apache.org</a></p>
<p>PMC <a href="http://projects.apache.org/indexes/pmc.html#Apache%20Ant">Apache
Ant</a></p>
<p>Releases can be downloaded from <a
href="http://ant.apache.org/bindownload.cgi">http://ant.apache.org/bindownload.c
gi</a></p>

<p>Most recent releases:</p>
<p>Release Apache Ant 1.7.0</p>
<p>Version 1.7.0</p>
<p>Date 2006-12-13</p>

<p>Access to the source code:</p>
<p>Browse <a
href="http://svn.apache.org/viewcvs.cgi/ant">http://svn.apache.org/viewcvs.cgi/a
nt</a></p>
<p>SVN Direct <a
href="http://svn.apache.org/repos/asf/ant">http://svn.apache.org/repos/asf/ant</
a></p>
. . .
```

Dokument RDF, znajdujący się w osobnym pliku i wskazany w powyższej stronie za pomocą elementu `<link rel="meta">`, zawiera ten sam komplet danych i wygląda następująco:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"?>
<!--
   Licensed to the Apache Software Foundation (ASF) under one or more
   contributor license agreements.
   The ASF licenses this file to You under the Apache License, Version 2.0
   (the "License"); you may not use this file except in compliance with
   the License.  You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
-->
<rdf:RDF xml:lang="en"
         xmlns="http://usefulinc.com/ns/doap#"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:asfext="http://projects.apache.org/ns/asfext#">
  <Project rdf:about="http://ant.apache.org/">
    <created>2006-02-17</created>
    <license rdf:resource="http://usefulinc.com/doap/licenses/asl20" />
    <name>Apache Ant</name>
    <homepage rdf:resource="http://ant.apache.org" />
    <asfext:pmc rdf:resource="http://ant.apache.org" />
    <shortdesc>Java-based build tool</shortdesc>
    <description>Apache Ant is a Java-based build tool. In theory,
      it is kind of like Make, but without Make's wrinkles.</description>
    <bug-database rdf:resource=
      "http://issues.apache.org/bugzilla/buglist.cgi?product=Ant" />
    <mailing-list rdf:resource="http://ant.apache.org/mail.html" />
    <download-page rdf:resource="http://ant.apache.org/bindownload.cgi" />
    <programming-language>Java</programming-language>
    <category rdf:resource=
      "http://projects.apache.org/category/build-management" />
    <release>
      <Version>
        <name>Apache Ant 1.7.0</name>
```

```
        <created>2006-12-13</created>
        <revision>1.7.0</revision>
      </Version>
    </release>
    <repository>
      <SVNRepository>
        <location rdf:resource="http://svn.apache.org/repos/asf/ant"/>
        <browse rdf:resource="http://svn.apache.org/viewcvs.cgi/ant"/>
      </SVNRepository>
    </repository>
  </Project>
</rdf:RDF>
```

Widać tutaj zatem dokładne powielenie informacji przy zastosowaniu dwóch zupełnie odmiennych sposobów jej prezentacji, dostosowanych do możliwości i potrzeb różnych odbiorców.

Utrzymywanie kilku (dwóch lub więcej) takich osobnych dokumentów może być w niektórych przypadkach łatwe i technicznie uzasadnione. Przykładowo, w sytuacji kiedy strona WWW jest generowana dynamicznie na podstawie zawartości bazy danych, generacja dodatkowego dokumentu RDF, opisującego zawarte na niej informacje, zazwyczaj nie stanowi znaczącego utrudnienia lub zmiany techniki funkcjonowania danego serwisu. Mogą występować również sytuacje, gdy rolę bazy danych pełni sam model RDF, a strony XHTML są generowane na jego podstawie (jak to w rzeczywistości ma miejsce w przypadku wspomnianej powyżej strony ASF i plików DOAP). Jednak w zależności od stopnia skomplikowania strony i sposobu tworzenia jej treści konieczność utrzymywania spójności danych w obu wersjach może być uciążliwa.

Trochę inna sytuacja występuje w przypadku stosowania modelu RDF do opisu metadanych na temat strony lub serwisu WWW, które ze swej natury nie są tak zmienne jak treści lub informacje publikowane na bieżąco. W takim przypadku komplet metadanych z zasady przeznaczonych do przetwarzania programowego, a nie dla czytelnika ludzkiego, może być wydzielony do osobnego pliku RDF. Nawet jeśli dane te znajdują się wśród informacji publikowanych na stronie, ich przeniesienie do oddzielnego dokumentu może ułatwić zadanie agentom programowym poszukującym konkretnych zestawów informacji.

Ponadto, elementów `<link>` może być umieszczonych na stronie wiele, tym samym mogą one wskazywać zestawy metadanych o różnym przeznaczeniu i opisujące stronę w różnym kontekście. Atrybut `type` umożliwia dodatkowo zastosowanie różnych formatów serializacji dołączanych do strony dokumentów RDF, nawet jeśli dokumenty te dotyczą tych samych danych, co może zwiększyć możliwości wykorzystania takiej strony w zależności od specyficznych potrzeb lub preferencji odbiorcy.

Często występującym w dzisiejszej sieci WWW przykładem tego typu zastosowań jest dołączenie do strony metadanych na temat jej autora (na przykład wizytówki, „profilu", z wykorzystaniem ontologii FOAF[6]) lub opisu licencji, czyli warunków korzystania z

[6] FOAF Vocabulary Specification 0.*9*, http://xmlns.com/foaf/spec/.

serwisu, albo konkretnego umieszczonego w nim dzieła (najczęściej za pomocą ontologii zgodnej z licencjami serii Creative Commons[7]). Nagłówek dokumentu XHTML może w takiej sytuacji wyglądać następująco:

```
<html . . .>
<head>
. . .
<link rel="meta" type="application/rdf+xml" title="FOAF" href="foaf.rdf" />
<link rel="alternate meta" type="text/rdf+n3" title="FOAF" href="foaf.n3" />
<link rel="license" type="text/rdf+n3" title="Licencja" href="license.rdf" />
. . .
```

2.2. Jeden dokument dla różnych odbiorców

Publikacja jednego dokumentu z przeznaczeniem zarówno dla czytelnika ludzkiego, jak i agentów programowych oznacza konieczność przygotowania takiej jego wersji, aby możliwe było jego odczytanie w dwóch postaciach lub automatyczne przekształcanie z jednej postaci w drugą, w zależności od kontekstu i potrzeb. Ze względu na to, że ekspresywność języków sieci semantycznej jest mniejsza niż języka naturalnego, w praktyce wersja przeznaczona dla użytkownika ludzkiego jest wersją podstawową, a wersja RDF jest w niej „osadzona" lub „ukryta" i uzyskiwana za pomocą odpowiedniego przekształcenia.

Istnieje kilka możliwych technik łączenia modelu RDF z dokumentem XHTML [11]. Począwszy od umieszczania bloku RDF/XML wprost w jego treści, poprzez umieszczanie zserializowanej postaci grafu w elemencie `<script>` lub `<object>`, aż po różne sposoby wykorzystania mechanizmu modularyzacji języka XHTML. Jednak prace prowadzone w ostatnich latach wskazują, że najbardziej praktyczne i najczęściej stosowane podejście sprowadza się do osadzenia wartości RDF w ramach treści XHTML za pomocą oznaczenia w specyficzny sposób tych fragmentów dokumentu, które należą również do modelu RDF. Techniki te obejmują aktualnie trzy główne podejścia: *microformats*, *eRDF* oraz *RDFa*.

2.2.1. Microformats

Technika określana mianem *microformats* [12] umożliwia zapisanie w dokumencie XHTML pewnego określonego zestawu metadanych oraz oznaczenia fragmentów jego treści w sposób umożliwiający jego maszynowe przetwarzanie. Technika ta nie umożliwia w praktyce zawarcia dowolnego modelu RDF w dokumencie XHTML i jest często traktowana jako pewien etap pośredni między stronami WWW przeznaczonymi wyłącznie dla odbiorcy ludzkiego a siecią semantyczną.

Umieszczenie metadanych w dokumencie XHTML, zgodnie z techniką *microformats*, następuje poprzez odpowiedni dobór wartości atrybutów elementów, występujących w doku-

[7] Describing copyright in RDF, http://web.resource.org/cc/.

mencie. Istotny jest przy tym fakt, że nie są stosowane żadne dodatkowe atrybuty, które nie istniałyby w specyfikacji języka HTML 4.0 i XHTML 1.0 – najczęściej wykorzystywany jest atrybut `class` lub `rel`. Znaczenie poszczególnych wartości wybranych atrybutów jest specyficzne dla danego mikroformatu i może zostać określone przez wskazanie profilu metadanych w atrybucie `profile` elementu `<head>` danej strony. Atrybut ten wskazuje adres URI dokumentu, zawierającego dany profil, czyli opis specyficznego znaczenia poszczególnych atrybutów elementów na stronie i ich dozwolonych wartości. Specyfikacja języka HTML 4.0, określająca zastosowanie atrybutu `profile`, nie opisuje sposobu, w jaki profil taki powinien zostać zbudowany. Powstała jednak inicjatywa XMDP (*XHTML Meta Data Profile*) [13], definiująca sposób opisu profilu metadanych i jest to sposób najczęściej wykorzystywany w technikach z rodziny *microformats*.

Przykładowo, następujący zestaw danych teleadresowych wyrażonych w standardowej postaci vCard [14]

```
BEGIN:VCARD
VERSION:3.0
N:Kowalski;Mikolaj
FN:Mikolaj Kowalski
ADR;TYPE=home:;;Zielona 7;Opole;;47-010;Polska
TEL:+48-600230113
TEL;TYPE=fax:+48-774501120
EMAIL;TYPE=internet,pref:mkowalski@example.org
URL:http://example.org/mkowalski/
END:VCARD
```

można oznaczyć na stronie internetowej za pomocą mikroformatu hCard [15] poprzez dobór odpowiednich wartości atrybutów `class` zagnieżdżonych elementów XHTML (w przykładzie wykorzystano `<div>` , `<p>` i `<span>`):

```
<html . . .>
<head profile="http://www.w3.org/2006/03/hcard">
. . .
<div class="vcard">
  <p class="fn">Mikołaj Kowalski</p>
  <div class="adr">
    <p class="street-address">Zielona 7</p>
    <p>
      <span class="postal-code">47-010</span>,
      <span class="locality">Opole</span>
    </p>
    <p class="country-name">Polska</p>
  </div>
  <p>Telefon: <span class="tel">+48-600230113</span></p>
  <p class="tel"><span class="type">Fax</span>:
    <span class="value">+48-774501120</span>
  </p>
  <p>E-mail: <span class="email">mkowalski@example.org</span></p>
  <p>WWW: <a class="url"
     href="http://example.org/mkowalski/">http://example.org/mkowalski/</a>
  </p>
</div>
. . .
```

Specyfikacja hCard definiuje metadane, odpowiadające niemal identycznie standardowi vCard. Ta sama informacja zapisana w formacie RDF/XML z wykorzystaniem ontologii opartej na tym formacie [16] może wyglądać następująco:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2006/vcard/ns#">

<vcard:VCard>
  <vcard:fn>Mikołaj Kowalski</vcard:fn>
  <vcard:adr>
    <vcard:Address>
      <vcard:street-address>Zielona 7</vcard:street-address>
      <vcard:postal-code>47-010</vcard:postal-code>
      <vcard:locality>Opole</vcard:locality>
      <vcard:country-name>Polska</vcard:country-name>
    </vcard:Address>
  </vcard:adr>
  <vcard:tel>tel:+48-600230113</vcard:tel>
  <vcard:fax>tel:+48-774501120</vcard:fax>
  <vcard:email rdf:resource="mailto:mkowalski@example.org"/>
  <vcard:url rdf:resource="http://example.org/mkowalski/"/>
</vcard:VCard>

</rdf:RDF>
```

Metadane umieszczone na stronie WWW za pomocą techniki *microformats* mogą być wykorzystywane przez różnego rodzaju oprogramowanie do najróżniejszych celów, niekoniecznie związanych z ideą *Semantic Web*. Automatyczne wyszukiwanie w dokumencie (X)HTML metadanych określonego typu (na przykład danych kontaktowych opisanych według mikroformatu hCard jak w przykładzie powyżej) może być realizowane zupełnie w oderwaniu od tej idei. Podobnie jak funkcjonujące dzisiaj rozpoznawanie przez wiele serwisów sieciowych atrybutów typu `rel="tag"` lub `rel="license"`, co również stanowi przykład techniki *microformats*. Jest to bardzo istotny krok w kierunku realizacji wizji strony internetowej, stanowiącej źródło informacji równocześnie dla czytelnika ludzkiego i dla agenta programowego. Jednak włączenie tych technik w główny nurt rozwoju sieci semantycznej wymaga zastosowania dodatkowego mechanizmu, umożliwiającego agentowi programowemu odnalezienie na stronie WWW metadanych i tworzenie na ich podstawie modelu RDF w sposób automatyczny i bez konieczności posiadania jakiejkolwiek wcześniejszej wiedzy na temat zakresu metadanych, występujących na danej stronie, a także bez wcześniejszych założeń co do rodzaju poszukiwanych informacji. Mechanizm taki jest stosunkowo prosty do zastosowania w połączeniu z aktualnie funkcjonującymi profilami *microformats* – zostanie on opisany w rozdziale 2.2.4.

2.2.2. eRDF

Technika eRDF (*Embedded RDF*) [17] umożliwia osadzenie w dokumencie XHTML dowolnego zestawu wyrażeń RDF poprzez odpowiednie stosowanie wartości atrybutów w poszczególnych elementach strony. Jest to technika zasadniczo podobna do *microformats*, jednak nieograniczona tylko do określonego, z góry zdefiniowanego zestawu metadanych.

Również tutaj nie wprowadzono żadnych nowych elementów lub atrybutów, które nie istniałyby w definicji języka XHTML. Takie ograniczenie ułatwia stosowanie tej techniki, jednak odbywa się kosztem ekspresyjności – nie wszystkie konstrukcje modelu RDF mogą być w ten sposób umieszczone w stronie WWW [18], chociaż zakres możliwości jest z reguły wystarczający dla przewidzianych zastosowań.

Ponieważ format eRDF w złożeniu umożliwia zapisanie w dokumencie XHML dowolnych wyrażeń RDF, konieczne jest określenie przestrzeni nazw i ontologii, jakie zostaną użyte. Uzyskuje się to przez zastosowanie w nagłówku dokumentu elementów `<link>` ze specjalną wartością atrybutu `rel`. Przykładowo:

```
<html . . .>
<head profile="http://purl.org/NET/erdf/profile">
  . . .
<link rel="schema.rdf"  href="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
<link rel="schema.dc"   href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.foaf" href="http://xmlns.com/foaf/0.1/" />
<link rel="schema.doap" href="http://usefulinc.com/ns/doap#" />
. . .
```

W praktyce stanowi to bezpośredni odpowiednik deklaracji przestrzeni nazw w dokumencie RDF/XML:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:doap="http://usefulinc.com/ns/doap#">
. . .
```

W treści dokumentu zadeklarowane prefiksy przestrzeni nazw są wykorzystywane w wartościach atrybutów w postaci `"prefix-nazwa"`. Specyfikacja eRDF [17] definiuje wiele konkretnych sposobów wykorzystania atrybutów `class`, `id`, `rel`, `rev`, `href` oraz `title` w poszczególnych elementach dokumentu XHTML wraz z wzajemnymi relacjami tych elementów względem siebie oraz opisuje, jak powinny być interpretowane zakodowane w ten sposób wyrażenia RDF. Zasady te prowadzą do stworzenia przemyślanej struktury strony WWW, która pozostaje czytelna dla człowieka, a równocześnie zawiera niemal dowolny model RDF.

Przykładowo poniższy fragment strony:

```
<html . . .>
<head profile="http://purl.org/NET/erdf/profile">
. . .
<link rel="schema.vcard" href="http://www.w3.org/2006/vcard/ns#" />
. . .
<div id="mk" class="-vcard-VCard">
  <p class="vcard-fn">Mikołaj Kowalski</p>
  <div id="adr" class="vcard-adr -vcard-Address">
    <p class="vcard-street-address">Zielona 7</p>
    <p>
      <span class="vcard-postal-code">47-010</span>,
      <span class="vcard-locality">Opole</span>
    </p>
    <p class="vcard-country-name">Polska</p>
  </div>
```

```
  <p>Telefon: <span class="vcard-tel">+48-600230113</span></p>
  <p>Fax: <span class="vcard-fax">+48-774501120</span></p>
  <p>E-mail: <a rel="vcard-email"
     href="mailto:mkowalski@example.org">mkowalski@example.org</a></p>
  <p>WWW: <a rel="vcard-url"
     href="http://example.org/mkowalski/">http://example.org/mkowalski/</a>
  </p>
</div>
. . .
```

zawiera następujące wyrażenia (trójki) RDF:

```
<#mk> rdf:type vcard-VCard .
<#mk> vcard-fn "Mikołaj Kowalski" .
<#mk> vcard-adr <#adr>
<#adr> rdf:type vard:Address .
<#adr> vcard:street-address "Zielona 7" .
<#adr> vcard:postal-code "47-010" .
<#adr> vcard:locality "Opole" .
<#adr> vcard:country-name "Polska" .
<#mk> vcard:tel "+48-600230113" .
<#mk> vcard:fax "+48-774501120" .
<#mk> vcard:email <mailto:mkowalski@example.org> .
<#mk> vcard:url <http://example.org/mkowalski/> .
```

co stanowi ten sam model RDF, jaki został przedstawiony w postaci RDF/XML w poprzednim rozdziale. Uzyskanie postaci RDF/XML na podstawie dokumentu XHTML zakodowanego zgodnie z konwencją eRDF wymaga poddania tego dokumentu pewnej transformacji. Najczęściej jest to dokonywane za pomocą transformacji XSLT, która powinna zostać wskazana na danej stronie, na przykład za pomocą techniki GRDDL, jak opisano w rozdziale 2.2.4.

2.2.3. RDFa

Sposobem osadzania zawartości RDF w dokumentach XHTML zasadniczo odmiennym od dwóch poprzednio opisanych jest technika RDFa [19]. Jest to specyfikacja opracowywana przez W3C (aktualnie jeszcze w fazie roboczej), której głównym założeniem jest umożliwienie wyrażenia dowolnego modelu RDF wprost w ramach dokumentu XHTML bez konieczności kodowania go w sztuczny i nieintuicyjny sposób.

Pierwszym z elementów wykorzystywanych przez RDFa jest zastosowanie przestrzeni nazw XML wprost, w postaci atrybutu `xmlns:prefix="uri"` w sposób taki sam jak w przypadku RDF/XML i innych dokumentów XML. Tak zdefiniowane przestrzenie nazw są następnie wykorzystywane w wartościach atrybutów w postaci `"prefix:nazwa"`.

Drugim zasadniczym elementem specyfikacji RDFa jest zastosowanie specjalizowanych atrybutów do zapisu wyrażeń RDF. Do atrybutów tych należą:

- `about` – do określenia identyfikatora URI, stanowiącego podmiot wyrażeń RDF; rola taka sama jak atrybutu `about` lub `id` w dokumencie RDF/XML,
- `property` – do wskazania właściwości, czyli predykatu wyrażenia RDF,
- `rel` – do wskazania właściwości wyrażenia, którego trzecim elementem jest referencja do zasobu (`href`),

- `href` – do wskazania identyfikatora URI zasobu/obiektu stanowiącego wartość danego wyrażenia,
- `content` – opcjonalnie do wskazania wartości wyrażenia, jeżeli jest ona inna niż zawartość danego elementu XHTML,
- `datatype` – opcjonalnie do wskazania typu wartości wyrażenia.

Takie podejście ułatwia zapisywanie wyrażeń RDF w sposób niemal bezpośredni, znacznie bardziej efektywny i czytelny (zwłaszcza w porównaniu do konstrukcji eRDF) oraz umożliwia zastosowanie pełniejszego zestawu konstrukcji RDF w dokumencie XHTML (na przykład stosowanie węzłów anonimowych). Niestety, stosowanie nowych rodzajów atrybutów, a także możliwość umieszczenia elementów `<meta>` oraz `<link>` w dowolnych miejscach dokumentu powoduje, że dokument RDFa przestaje być poprawny w rozumieniu funkcjonujących dzisiaj na WWW specyfikacji HTML 4.0 i XHTML 1.0. Dopiero specyfikacja XHTML 2.0 umożliwi w pełni zgodną ze standardami obsługę dokumentu zakodowanego według koncepcji RDFa.

Przykład analogiczny do przedstawionych w poprzednich rozdziałach z wykorzystaniem formatu RDFa może wyglądać następująco:

```
<html . . .>
<head xmlns:vcard="http://www.w3.org/2006/vcard/ns#">
. . .
<div id="mk" class="vcard:VCard">
  <p property="vcard:fn">Mikolaj Kowalski</p>
  <div id="adr" rel="vcard:adr" about="#adr" class="vcard:Address">
    <div property="vcard:street-address">Zielona 7</div>
    <p>
      <span property="vcard:postal-code">47-010</span>,
      <span property="vcard:locality">Opole</span>
    </p>
    <p property="vcard:country-name">Polska</p>
  </div>
  <p>Telefon: <span property="vcard:tel">+48-600230113</span></p>
  <p>Fax: <span property="vcard:fax">+48-774501120</span></p>
  <p>E-mail: <a rel="vcard:email"
     href="mailto:mkowalski@example.org">mkowalski@example.org</a></p>
  <p>WWW: <a rel="vcard:url"
     href="http://example.org/mkowalski/">http://example.org/mkowalski/</a>
  </p>
</div>
. . .
```

2.2.4. Automatyczna transformacja XHTML do RDF – mechanizm GRDDL

Aby metadane lub fragmenty modelu RDF osadzone w dokumencie XHTML funkcjonowały jako źródła danych dla agentów sieci semantycznej, musi istnieć mechanizm automatycznego pozyskania „czystego” dokumentu RDF na podstawie danej strony WWW.

Jedną z dyskutowanych aktualnie propozycji standaryzacji takiego mechanizmu jest opracowywana w ramach prac W3C technika o nazwie GRDDL (*Gleaning Resource Descriptions from Dialects of Languages*) [20]. Oparta jest ona na bardzo prostej obserwacji – ponieważ strona WWW, będąca poprawnym dokumentem w języku XHTML, jest z definicji

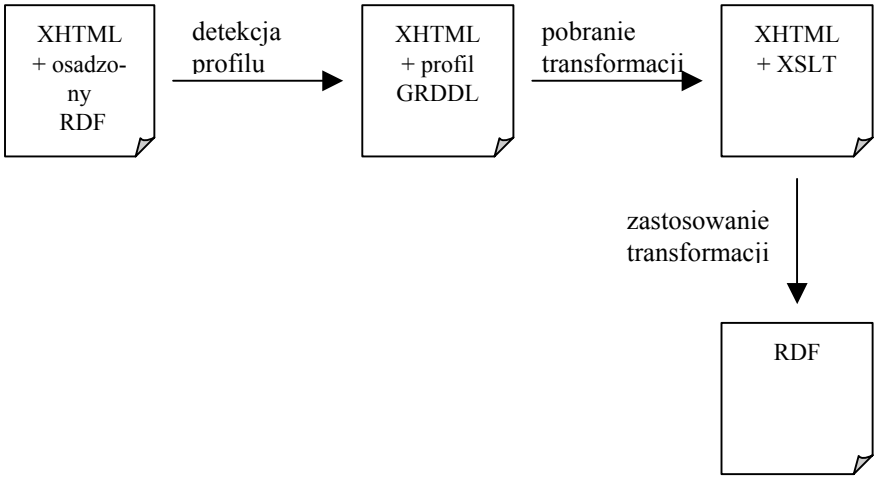
dokumentem XML, oraz równocześnie jeden z najpopularniejszych sposobów serializacji grafu RDF, jakim jest format RDF/XML, również jest dokumentem XML, zatem mamy tu do czynienia z przekształceniem jednego dokumentu XML w drugi. Przy aktualnym stanie technologii wykorzystywanych w Internecie przekształcenie takie najłatwiej wykonać za pomocą transformacji XSLT.

GRDDL zakłada określenie odpowiedniego profilu metadanych dokumentu, dzięki czemu nadaje się znaczenie atrybutom wskazującym poszczególne elementy transformacji. Najważniejszym z tych elementów jest wskazanie arkusza XSLT opisującego sposób przekształcenia strony do postaci dokumentu RDF/XML.

W poniższym przykładzie atrybut `profile` elementu `<head>` wskazuje, że dana strona może zostać poddana transformacji opisanej zgodnie z mechanizmem GRDDL (`profile="http://www.w3.org/2003/g/data-view"`), a znajdujący się w nagłówku strony element `<link rel="transformation">` wskazuje arkusz XSLT wykorzystywany przy tej operacji.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head profile="http://www.w3.org/2003/g/data-view">
  <link rel="transformation" href="transformacja.xsl" />
  . . .
```

Ideę działania tego mechanizmu przedstawia rys. 1. Po stwierdzeniu obecności w dokumencie XHTML profilu GRDDL odszukiwany jest element `<link>`, wskazujący arkusz transformacji XSLT, dokument ten jest następnie pobierany i zastosowany do dokumentu źródłowego, w efekcie czego uzyskuje się dane w formacie RDF/XML.

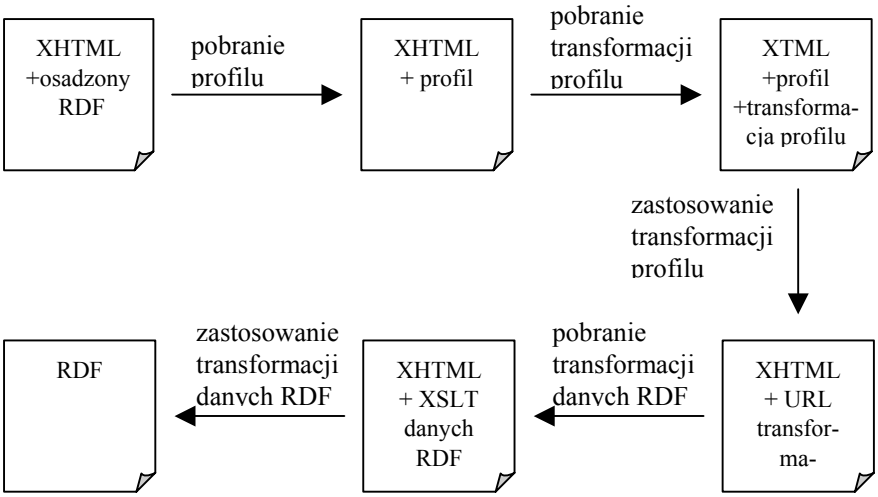


Rys. 1. Zastosowanie mechanizmu GRDDL
Fig. 1. Applying GRDDL transformation

Dokument XHTML może zawierać wiele fragmentów informacji wykorzystujących różne ontologie. Z tego względu może podlegać wielokrotnym transformacjom za pomocą kolej-

nych arkuszy XSLT wskazanych przez umieszczenie w nagłówku strony kolejno kilku elementów `<link rel="transformation">`.[8]

GRDDL dopuszcza również określanie sposobów transformacji pośrednio, przez wskazanie odpowiedniego profilu dokumentu, który sam wskazuje profil GRDDL i właściwy mechanizm transformacji. W takim przypadku w pierwszym kroku następuje pobranie dokumentu profilu, zastosowanie do niego wskazanej transformacji profilu (`rel= "transformation" href="http://www.w3.org/2003/g/glean-profile"`), która ujawnia wskazany odnośnikiem z atrybutem `rel="profileTransformation"` adres URL właściwego arkusza XSLT. Ten plik określa transformację dokumentu źródłowego wykonywaną w kolejnym kroku (rys. 2).



Rys. 2. Zastosowanie mechanizmu GRDDL za pośrednictwem profilu transformacji
Fig. 2. Applying GRDDL transformation via profile

Jest to metoda łatwiejsza w zastosowaniu z punktu widzenia autora strony WWW i stosowana przede wszystkim dla standardowych sposobów osadzania danych RDF w dokumencie XHTML, dla których opracowano publicznie dostępne profile GRDDL. Przykładowo, pod adresem http://purl.org/NET/erdf/profile dostępny jest profil dla standardu *eRDF* o postaci:

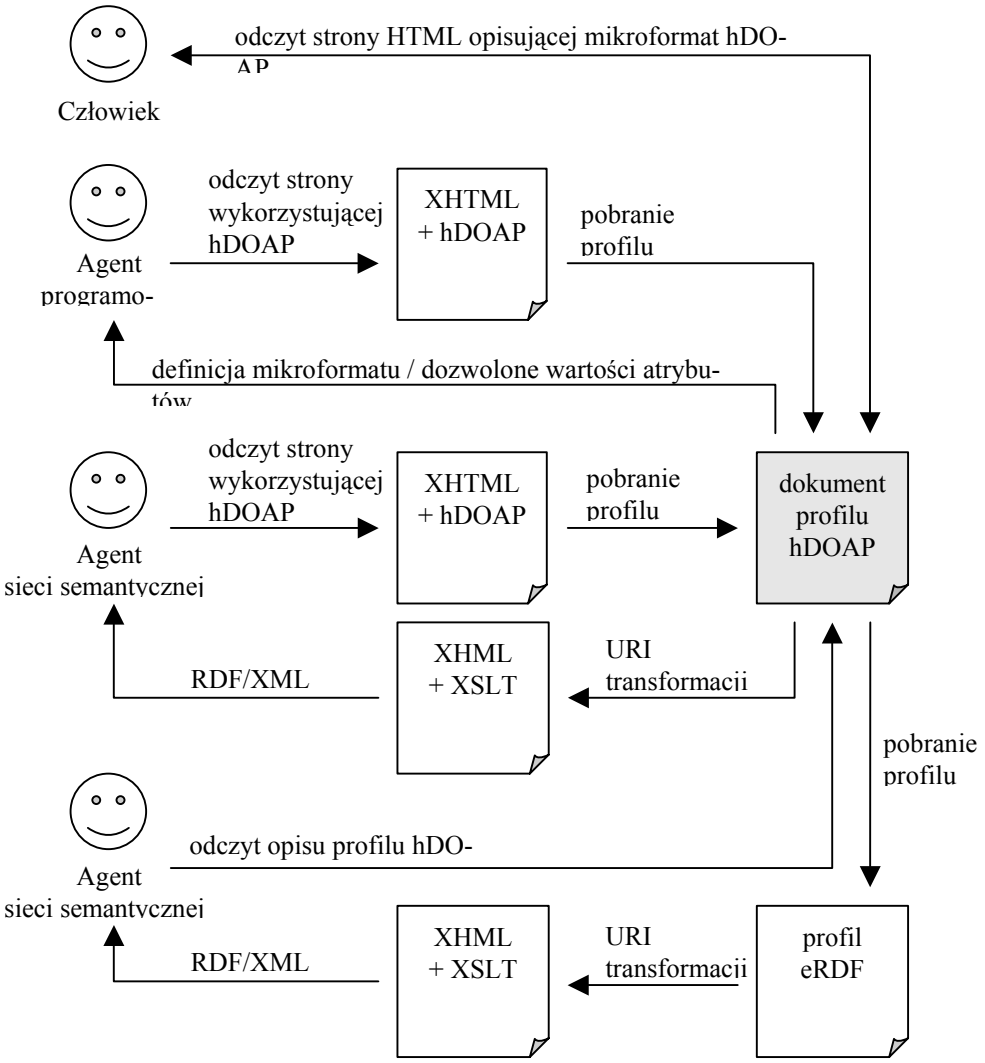
```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head profile="http://www.w3.org/2003/g/data-view">
    <title>Embedded RDF HTML Profile</title>
    <link rel="transformation" href="http://www.w3.org/2003/g/glean-profile" />
  </head>
  <body>
    <p>
      <a rel="profileTransformation"
          href="http://purl.org/NET/erdf/extract-rdf">GRDDL transform</a>
    </p>
```

[8] Przykład zastosowania techniki GRDDL – *Joe Lambdas homepage* dostępny pod adresem http://www.w3.org/2004/lambda/Sites/index.html przedstawia wykorzystanie kilku różnych transformacji XSLT do pobrania informacji RDF z fikcyjnej strony domowej.

```
  </body>
</html>
```

W nagłówku strony, w której został osadzony RDF zgodnie z konwencja eRDF, wystarczy wskazać adres powyższego dokumentu, a mechanizm opisany powyżej odczyta adres `http://purl.org/NET/erdf/extract-rdf` jako źródło arkusza transformacji XSLT, którym należy poddać dokument źródłowy. Autor strony ogranicza się zatem jedynie do określenia odpowiedniej wartości atrybutu `profile` bez względu na umiejscowienie i zawartość plików XSLT, dokonujących faktycznego przekształcenia jego strony na format RDF.

Warto przy tym również zaznaczyć, że przedstawiony powyżej profil jest przykładem bardzo skromnym – w praktyce może to być dowolny dokument XHTML, spełniający warunek posiadania odpowiednich wartości wskazanych powyżej elementów (atrybuty `profile`, `rel="transformation"`, `rel="profileTransformation"`). Przykładowo, profil mikroformatu hDOAP dostępny pod adresem `http://purl.org/stuff/hdoap/profile` stanowi w praktyce dokument opisujący jego zastosowanie (znaczenie poszczególnych wartości atrybutu `class`) w postaci czytelnej dla człowieka oraz zgodny ze standardem XMDP [13]. Dodatkowo, dokument ten zawiera osadzony model RDF zakodowany w konwencji eRDF. Jest to zatem bardzo dobry przykład dokumentu funkcjonującego na pograniczu sieci semantycznej i sieci WWW, spełniającego równocześnie różne role w zależności od możliwości i potrzeb odbiorcy (rys. 3).



Rys. 3. Różne przypadki użycia dokumentu profilu hDOAP
Fig. 3. Different use cases of hDOAP profile document

3. Podsumowanie

Sieć semantyczna stanowi przejaw ewolucyjnego rozwoju sieci WWW – rozwoju zarówno nieuchronnego, jak i całkowicie naturalnego z punktu widzenia wizji twórcy WWW, Tima Bernersa-Lee [1]. Mimo że idea *Semantic Web* jest rozwijana od około 10 lat, dopiero w ostatnich latach opracowano i zaczęto stosować mechanizmy efektywnego łączenia zawartości dzisiejszych stron WWW z zawartością przystosowaną do przetwarzania programowego w sieci semantycznej. Mechanizmy takie okazują się koniecznie nie tylko w okresie przejściowym, gdy ilość informacji dostępnej w postaci modeli RDF jest jeszcze znikoma w porównaniu do ilości informacji na stronach WWW przeznaczonych wyłącznie dla czytelnika ludzkiego. Również w dalszych etapach rozwoju sieci semantycznej nie należy zapominać,

że głównym odbiorcą treści publikowanych na WWW pozostanie człowiek, a jego sposób odczytu tych informacji nigdy nie będzie się znacząco różnił od obecnego. Oznacza to zatem równoczesne funkcjonowanie w sieci semantycznej dwóch rodzajów „czytelników" – człowieka i agentów programowych.

Przedstawione w artykule techniki umożliwiają współistnienie danych przeznaczonych dla tych dwóch rodzajów odbiorców nie tylko w postaci grupy powiązanych dokumentów, ale także w postaci dokumentów wspólnych, zawierających modele RDF umieszczone wewnątrz treści XHTML i automatycznie konwertowanych na potrzeby konkretnego typu odbiorcy. Omówione techniki różnią się zarówno stopniem skomplikowania, jak i zakresem oferowanych możliwości – od oznaczania w treści zestawu ściśle określonych metadanych, aż po umieszczanie w niej praktycznie dowolnie skonstruowanego modelu RDF. Wszystkie te mechanizmy są w założeniach bardzo proste i poza nielicznymi wyjątkami zgodne z aktualnym stanem technologii stosowanych w Internecie. Dzięki temu minimalizują one nakłady, jakie należy ponieść na zamianę aktualnie stosowanych sposobów publikacji informacji na stronach WWW w trakcie ich ewolucji w stronę sieci semantycznej. Ponieważ techniki takie są już dzisiaj stosowane na wielu stronach internetowych, należy uznać, że jeden z najuważniejszych kroków w kierunku integracji dzisiejszej sieci WWW i przyszłościowej sieci semantycznej został już wykonany.

LITERATURA

1. Berners-Lee T.: Semantic Web Roadmap. 1998. http://www.w3.org/DesignIssues/Semantic (2007.06.05).
2. Klyne G., Carroll J.: Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, World Wide Web Consortium 2004. http://www.w3.org/TR-/2004/ REC-rdf-concepts-20040210/ (2007.06.05).
3. Hayes P.: RDF Semantics. W3C Recommendation, World Wide Web Consortium 2004. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/ (2007.06.05).
4. Manola F., Miller E.: RDF Primer. W3C Recommendation, World Wide Web Consortium, 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ (2007.06.05).
5. McGuinness D., van Harmelen F.: OWL Web Ontology Language Overview. W3C Recommendation, World Wide Web Consortium 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/ (2007.06.05).
6. Beckett D.: RDF/XML Syntax Specification (Revised). W3C Recommendation, World Wide Web Consortium 2004. http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/ (2007.06.12).

7. Berners-Lee T.: Notation 3, An RDF language for the Semantic Web. http://www.w3.org/DesignIssues/Notation3 (2007.06.12).
8. Grant J., Beckett D.: RDF Test Cases. W3C Recommendation, World Wide Web Consortium, 2004. http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/ (2007.06.12).
9. Swartz A.: application/rdf+xml Media Type Registration. RFC 3870, The Internet Society 2004. http://www.ietf.org/rfc/rfc3870.txt (2007.06.12).
10. Beckett D.: Turtle – Terse RDF Triple Language. Dave Beckett 2006. http://www.dajobe.org/ 2004/01/turtle/ (2007.06.12).
11. Palmer S.: RDF in HTML: Approaches. 2002. http://infomesh.net/2002/rdfinhtml/ (2007.06.14).
12. Allsopp J: Microformats: Empowering Your Markup for Web 2.0. Springer-Verlag New York Inc., 2007.
13. Çelik T.: XMDP: Introduction and Format Description. Global Multimedia Protocols Group, 2003. http://gmpg.org/xmdp/description (2007.06.14).
14. Dawson F., Howes T.: vCard MIME Directory Profile. RFC 2426, The Internet Society 1998 http://www.ietf.org/rfc/rfc2426.txt (2007.06.15).
15. Çelik T., Suda B.: hCard Specification. Microformats.org 2007. http://microformats.org/-wiki/hcard (2007.06.15).
16. Halpin H., Suda B., Walsh N.: An Ontology for vCards. 2006. http://www.w3.org/2006-/vcard/ns (2007.06.15).
17. Davis I.: RDF In HTML, Talis Information Ltd 2005. http://research.talis.com/2005/erdf-/wiki/Main/RdfInHtml (2007.06.16).
18. Davis I.: Limitations Of Embedding RDF In HTML, Talis Information Ltd 2005. http://research.talis.com/2005/erdf/wiki/Main/LimitationsOfEmbeddingRDFInHTML (2007.06.16).
19. Adida B., Birbeck M.: RDFa Primer 1.0 – Embedding RDF in XHTML. W3C Working Draft. World Wide Web Consortium 2007. http://www.w3.org/TR/2007/WD-xhtml-rdfa--primer-20070312/ (2007.06.16).
20. Connolly D.: Gleaning Resource Descriptions from Dialects of Languages (GRDDL). W3C Candidate Recommendation. World Wide Web Consortium 2007. http://www.w3.org/TR/-2007/CR-grddl-20070502/ (2007.06.19).

Recenzent: Dr inż. Sławomir Niedbała

Wpłynęło do Redakcji 13 lipca 2007 r.

Abstract

Accessing information available on WWW nowadays requires a lot of effort from human reader and is usually inaccessible for automated processing by software agents. The Semantic Web vision aims to change this situation by providing infrastructure for expressing data and knowledge in machine-understandable format. Using Resource Description Framework (RDF) and associated techniques from W3C will make it possible for software agents to use data available in the Web just like people use current WWW. But still human readers will be most important part of WWW users. Hence it is necessary to create efficient mechanisms of publication of information for these two completely different kinds of "web users" at the lowest cost possible.

Techniques described in the article allows authors of web pages to publicize information in separate documents for both kinds of readers or in single document in which RDF data are inserted as an integral part of a web page. Embedding RDF data directly inside an XHML document is more interesting from practical point of view, since it allows to maintain only one document for both WWW and Semantic Web purposes. There are several techniques for embedding machine readable data in XHTML content – most practical and widely used are described in the article: microformats, eRDF and RDFa. These three mechanisms differs significantly in their construct complexity and expressiveness. From easy inserting of simple metadata (microformats), through complicated but much more capable embedding of any RDF statements (eRDF) up to most powerful and convenient but not yet conforming to current specification of XHTML language (RDFa).

To make RDF statements or metadata embedded in human readable WWW pages available to autonomous software agent, some mechanism of RDF discovery and automated translation to RDF must be provided. Using GRDDL technique described by W3C, web page author can provide a set of transformation instruction in form of XSLT stylesheet, and link them to the page directly (fig. 1) or via metadata profile (fig. 2). With this mechanism information included in a web page as an embedded RDF can be easily obtained by software agents in form of RDF model serialized to RDF/XML representation.

All described techniques are available today and are already used on many web pages. Thus it can be stated that one of the most important steps from nowadays WWW to future Semantic Web is already done.

Adres

Michał ŚWIĄTKIEWICZ: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, michal@swiatkiewicz.net .