社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2009－ＦＩ－94 (1)
2009－ＤＤ－70 (1)
2009／3／25

RDFaを用いたメタデータオーサリングシステムの構築

塩澤 元† 松本 章代† Martin J.Dürst†

† 青山学院大学 理工学部
〒229–8558 神奈川県相模原市淵野辺 5–10–1
E-mail: hajimes@sw.it.aoyama.ac.jp, {akiyo, duerst}@it.aoyama.ac.jp

あらまし 2008年10月にセマンティックウェブの一環として，XHTML文書にメタデータを埋め込むための枠組みであるRDFaが勧告された．我々はWebブラウザ上で動作するRDFaのためのオーサリングシステムを開発した．現在，RDFaはソースコードからの編集が一般的であるが，本システムでは編集対象を選択してより直感的に編集可能である．加えて，すでに埋め込まれているメタデータをアウトライン形式で表示しており，そこから対象のメタデータを編集できる．また，より人間がRDFaを利用しやすくなるようにするため，Webページに関する頻繁に用いられる語彙を日常的な言葉で提示する．これによって幅広いユーザがRDFaを利用することが可能になり，その普及に繋がると考えられる．本稿ではRDFaの普及を目的とした，ユーザが簡単にRDFaを利用できるオーサリングシステムの構築について述べる．

キーワード RDFa，セマンティックウェブ，メタデータ，XHTML

The Construction of a Metadata Authoring System using RDFa

Hajime SHIOZAWA†, Akiyo MATSUMOTO†, and Martin J. DÜRST†

† College of Science and Engineering, Aoyama Gakuin University
Fuchinobe 5–10–1, Sagamihara, Kanagawa, 229–8558 Japan
E-mail: hajimes@sw.it.aoyama.ac.jp, {akiyo, duerst}@it.aoyama.ac.jp

Abstract In October 2008, RDFa was approved by the W3C as a Recommendation. RDFa allows to embed RDF metadata in XHTMLs documents, bringing the traditional Web and the Semantic Web closer together. We have created an authoring system for RDFa that works in a Web browser. Using this system, it is possible to select part of a Web page and add semantic data without editing the Web page source code directly. In addition, the embedded metadata is displayed in a outline view. This facilitates understanding of the data and allows direct editing. To help users not familiar with RDFa, we automatically translate RDF URIs for frequently used terms to plain Japanese (or English). With this work, we hope to contribute to the popularization of RDFa.

Key words RDFa, Semantic Web, Metadata, XHTML

1. はじめに

World Wide Web（以下Web）で意味を扱うため，Tim-Berners Leeによって Semantic Webが提唱された．これは既存のWebを拡張してメタデータを付与し利用することで，Webとコンピュータの親和性を高めるものである．2008年10月にXHTML文書中にメタデータを埋め込むための枠組みであるRDFaの構文仕様[1]が勧告された．RDFaを用いるとXHTMLに意味を与えることが可能になり，コンピュータ処理の精度が向上する．また，それによってXHTMLとWebアプリケーションなどの連携が簡単になる．本稿ではRDFaの普及を目的と

した，ユーザが簡単に RDFa を利用できるオーサリングシステムの構築について述べる．

2. RDFa とオーサリングシステム

本章ではまず RDFa の概念的な背景となる RDF について述べてから，RDFa について解説する．そして，ユーザが簡単に RDFa を利用できるようなサリングシステムを提案する．

2.1 RDF

RDF は Web 上にある情報を記述するための枠組みであり，W3C によって勧告されている [2]．RDF ではリソースに対して主語（Subject），述語（Predicate），目的語（Object）と呼ばれる語を用いて情報を記述する．リソースとは，Web 上に存在する文書や画像などの総称である．“この Web ページの作者は塩澤元である” という情報を RDF においてモデル化する場合，“Web ページ” は主語，“作者” は述語，“塩澤元” は目的語になる．しかし “Web ページ” という文字列では，Web 上に多数存在するリソースから特定のページだけを一意に識別することはできない．そこで特定のリソースを一意に識別するために URI を用いる．主語，述語，目的語を組み合わせて記述し，これらの組をトリプルと呼ぶ．また RDF は図 1 のように，アークとノードを用いたグラフとして表現する．

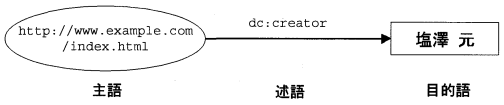

図 1　グラフ化された RDF

RDF によってモデル化された情報はコンピュータで扱うために様々な形式で表現される．代表的な形式として Notation3 [3]，RDF/XML [4] などがある．Notation3 はテキストベースの形式であり主語，述語，目的語を順に並べただけの平坦な構文が特徴である．RDF/XML は XML をベースとした構文を持ち，現在 Web 上で一番普及している形式である．

2.2 RDFa

RDFa は RDF でモデル化された情報を，XHTML 文書中に埋め込むための枠組みである．RDFa は要素や属性を用いて，情報を RDF で表現して XHTML に埋め込む．それによって，コンピュータは図 2(注1) のように，どの部分にどのような情報があるかを認識できる．

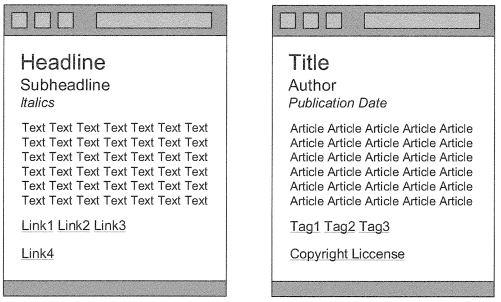

図 2　コンピュータによる意味の認識

RDFa によって従来の `meta` 要素の `keyword` 属性などより，Web ページに関する情報を詳細に記述できる．その結果，ロボット検索エンジンの精度は向上する．また，イベントに関する情報を RDFa でマークアップすれば，ブラウザ上で Google カレンダー(注2) などに自動的に登録できる．

2.3 RDFa 構文

ここからは，実際の RDFa の構文の一例として `property` 属性について説明する．`property` は文字に意味付けを行うための属性であり，以下のように記述する．“この Web ページの作者は Hajime Shiozawa である” というように，文字列の意味を明示的することができる．

```
<p property='dc:creator'>Hajime Shiozawa</p>
```

また Web ページと直接関係のない情報を RDFa として埋め込むこともできる．図 3 は人物の情報を RDFa で表現した例である．RDFa において，Web ページとは直接関係のない情報を表したい場合，`typeof` 属性にその情報の種類を記述するのが一般的である．これによって情報を構造化することができる．具体的には図 4 が示すように，人物は “Hajime Shiozawa” という単なる文字列ではなく，名前やメールアドレスなど様々な属性を持ったリソースであるとする．また，リソース以外に URI で識別されない特殊なノードである空白ノードを用いる場合もある．これは一般的に `_:hoge` と記述される．

RDFa の構文は，他の RDF の表現形式である Notation3，RDF/XML などの構文と大きく異なる．Notation3 や RDF/XML が RDF モデルに特化した構文であ

(注1)：出典：`http://www.w3.org/TR/xhtml-rdfa-primer/`

(注2)：`http://www.google.com/calendar/`

```
<div typeof='foaf:Person'>
  <p>I'm
   <span property='foaf:firstname'>Hajime</span>
   <span property='foaf:family_name'>Shiozawa</span>
  .</p>
  <p>Mail address is
   <a rel='foaf:mbox'
      href='mailto:hajimes@sw.it.aoyama.ac.jp'>here.</a>
  </p>
</div>
```

図 3　RDFa を用いた人物情報の記述

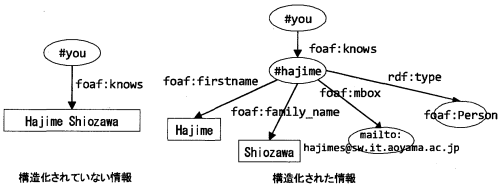

図 4　人物情報の構造化

るのに対して，RDFa の構文は XHTML に依存している．また，必ず RDF の情報以外に XHTML で書かれた情報が混在している．それによって，他の表現形式よりもソースコードが複雑であるといえる．

2.4　RDFa の現状

RDFa と似たような先行技術として microformats [5] がある．microformats はすでに Web 上の様々な場所で利用され，多くのテンプレートが用意されている．構文が RDFa と比べて簡単であり手軽に利用できる点が特徴であるが，テンプレートが決められているので，hAtom，hCalendar，hCard などの限られた情報しか扱うことができない．RDFa は microformats と比較すると後発の技術であるため普及は遅れているが，るが，技術的には非常に利点が多いとされている [6]．また，BBC の Web サイト上の番組プログラムでは microformats の hCalendar による配信が廃止され，RDFa を採用する代替案があるなどの積極的な動きがある [7]．

2.5　RDFa のためのオーサリングシステムの提案

RDFa にはより簡単に利用，編集できる環境が必要である．例えばユーザがブログを閲覧していたとき，記事の中に建物の位置情報が RDFa として付与されているとする．その場合すぐにシステムを起動して RDFa を確認できると，RDFa を効果的に利用可能である．また，新しいの RDFa 付与やすでに埋め込まれている RDFa の編集の場面でも，編集操作をブラウザ上のみで簡単に行える環境が必要である．そこで RDFa の気軽な利用と編集が可能な，人間への親和性を高めたオーサリングシステムを構築する．またそれによって RDFa の普及させ，Semantic Web の考え方を広めることを目的とする．

3.　RDFa の表示

本章ではシステムで実現した RDFa の表示機能について述べる．RDFa が独立して表示されることによって，ユーザに RDFa を意識させることができる．また，XHTML の情報と分離して表示されるとその編集を行いやすくなる．加えて，編集後の RDFa を表示ですぐ確認することもできる．しかし RDFa で表される情報は一般的なブラウザ上で確認できない．そこでシステムでは XHTML から RDFa だけを抽出し，人間が見やすいように可視化する．

3.1　アウトライン形式での可視化

RDFa をアウトライン形式という形式を用いて表示する [8]．これは，[主語, (述語, 目的語)^n] という一つの主語を中心とした集合を一つのテーブルで表現して記述する形式である．

図 5　アウトライン形式での表示

この形式はテキストベースでありながら，RDF の持つ構造が非常に読み取りやすい．トリプル形式の場合，単にトリプルを列挙していることに加えて，同一主語に関する情報が一箇所にまとまっていない．それによってすぐにその情報構造を読み取ることはできない．一方，アウトライン形式では同一主語の情報を一つのテーブルとしてまとめて，テーブルに階層構造を持たせることでその情報構造を人間が読み取ることができる．また，もう一つの特徴として対象のテーブルを開閉し，限られたスペースを必要に応じて利用できる．

アウトライン形式は構造的にツリー構造と同等であり，アウトライン形式でメタデータを表示するため，RDFa をツリーへと変換する．まず構文解析器によって XHTML 文書から RDFa だけを抽出する．その後，得られるトリプルをツリー構造へと変換する．また，図 6 が示すようにツリーはその深さができるだけ浅くなるように最適化する．なぜなら，深さを優先してツリー構造へと変換すると階層構造が深くなり，見づらくなってしまうからである．

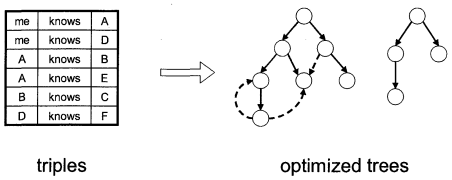

図 6　トリプルから最適化されたツリーへの変換

3.2　代表的な語彙の抽出と置換

RDFa は図 4 のように構造化されていた方がコンピュータにとっては扱いやすい．しかし，このまま表示すると人間にとって非常に理解しづらい．またアウトライン形式では主語を選択して情報を表示するので，主語が空白ノードだと人間にとって分かりにくい．そこで，その情報を代表する人間に分かりやすいような語彙を自動的に抽出し，主語と置換する．具体的には，`typeof` 属性の値ごとに代表的なプロパティをシステム内に登録し，主語がリソースや空白ノードの場合に自動的にそれを抽出する．そして，表示の際だけリソースや空白ノードと置き換える．

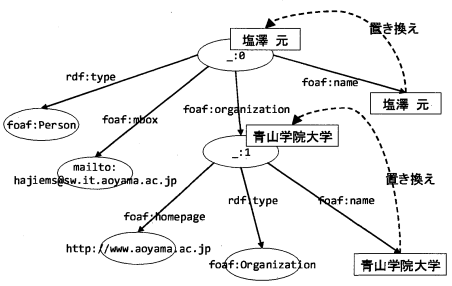

図 7　代表的なプロパティの抽出と置き換え

例えば図 7 では空白ノードである `_:0` と `_:1` が主語である．この場合 `foaf:Person` と `foaf:Organizaiton` の代表的なプロパティとして登録してある，`foaf:name` の値を自動的に抽出する．そして，アウトライン形式で表示する時だけ主語と置き換える．その結果，本来の主語である `_:0` と `_:1` が，“塩澤元” と “青山学院大学” に置き換わって表示され，人間に理解しやすいものになる．

3.3　語彙の登録と多言語化

RDFa は語彙を URI で識別するので，利用する際には自らが利用する語彙を選ぶ．そこで Web 上で頻繁に用いられる語彙セットである Friend of a Friend（FOAF）(注3)，Dublin Core（DC）(注4)，vCard(注5)，vEvent(注6) などから 300 語程度の語彙をリストに登録した．これによってユーザが語彙を考える手間を省き RDFa の簡単な付与が可能である．その一部を表 1 に示す．また，多言語化の一環として語彙のリストを英語と日本語に対応させた．一部の語彙については Christine Frodl らによる翻訳 [9] を利用し，ドイツ語での表示も可能である．

表 1　DC Metadata Element Sets の翻訳

URI	英語訳	日本語訳	ドイツ語訳
dc:contributor	contributor	貢献者	Mitwirkende(r)
dc:coverage	coverage	範囲	Geltungsbereich
dc:creator	creator	作者	Urheber(in)
dc:date	date	日付	Zeitangabe
dc:description	description	説明	Beschreibung
dc:format	format	形式	Format
dc:identifier	identifier	識別子	Identifikator
dc:language	language	言語	Sprache
dc:publisher	publisher	公開者	Verleger(in)
dc:relation	relation resource	関係	Beziehung
dc:rights	rights	権利	Rechte
dc:source	source	ソース	Quelle
dc:subject	subject	主題	Thema
dc:title	title	タイトル	Titel
dc:type	type	タイプ	Typ

4.　RDFa の編集

本章ではシステムで実現した RDFa の編集機能について述べる．Web ページの作成者が RDFa を付与する際の手間を減らすとともに，積極的に RDFa を付与できるような環境を構築する．

4.1　ブラウザ上での編集

RDFa は XHTML を用いて表現されており，今のところ編集はプレーンテキストから行うことが一般的である．それによって RDFa 付与には手間がかかり，ユーザは積極的に RDFa を付与できない．そこでリッチテキスト編集を用いて RDFa を編集可能にする．本システムではブラウザ上で編集対象を選択し，より直感的な操作で RDFa を追加することが可能である．実際には RDFa の編集を以下のように区分し，実装を行った．

（1）　既存の RDFa の編集
（2）　選択対象への RDFa の付与
（3）　構造化と `typeof` 属性の付与

4.1.1　既存の RDFa の編集

RDFa は XHTML で記述されるので，既存の RDFa の編集には DOM ツリーの編集が必要である．ブラウザ上でそれを可能にするため RDFa の表示を利用する．表示された RDFa とその抽出源となる DOM ツリー上の要素

(注3)：`http://xmlns.com/foaf/spec/`
(注4)：`http://dublincore.org/documents/dcmi-terms/`
(注5)：`http://www.w3.org/2006/vcard/ns`
(注6)：`http://www.w3.org/2002/12/cal/rfc2445`

を対応付けておく．図 8 が示すように，編集の際には表示されている RDFa の一覧から対象を選んで編集する．そして，その変更を解釈したシステムが直接 DOM ツリーを編集し，ブラウザ上での RDFa の修正が可能である．

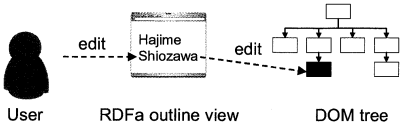

図 8　既存の RDFa の編集

4.1.2　選択対象への RDFa の付与

RDFa が付与されていない文字列やリンク，画像などへのメタデータの付与はブラウザで対象を選択して行う．ブラウザで対象が選択されると，選択された要素を取得してシステムが DOM ツリーを編集する．図 9 は，“Hajime” という文字列を選択し編集するという操作の結果を示している．下線はブラウザ上での選択範囲を表し，太字斜線は編集操作によって新しく追加された箇所を示している．この表記法は図 10 から図 13 においても同様である．図 9 のように文字列が選択された場合には，その部分をタグで囲み RDFa を付与する．また，図 10 のようにリンクや画像など，新たにタグを追加する必要がない場合は属性を追加して RDFa を付与する．

4.1.3　構造化と typeof 属性の付与

RDFa を用いて Web ページと直接関係のない情報を付与する多くの場合に `typeof` 属性を用いて型指定を行う．しかし，`typeof` 属性を付与するためには，その情報が記述されている部分を要素で構造化しなければならない．例えば，一連の文章の中でイベントに関しての文がある．

```
<p>I'm Hajime Shiozawa</p>
        ⇩
<p>I'm
  <span property='foaf:firstName'>
    Hajime
  </span>
 Shiozawa</p>
```

図 9　文字列への RDFa 付与

```
<p>photo:
  <img src='me.jpg' />
</p>
        ⇩
<p>photo:
  <img rel='foaf:depiction' src='me.jpg' />
</p>
```

図 10　画像への RDFa 付与

```
<p>***Statement***
***Event information***
***Statement***
</p>
        ⇩
<p>***Statement***
  <div tyepof='event:Vevent'>
    ***Event information***
  </div>
***Statement***</p>
```

図 11　div 要素と typeof 属性の追加

```
<p>***Statement***
  <div>
    ***Event information***
  </div>
***Statement***</p>
        ⇩
<p>***Statement***
  <div tyepof='event:Vevent'>
    ***Event information***
  </div>
***Statement***</p>
```

図 12　typeof 属性のみの追加

```
<p>***Statement***
  <em>Event name</em>
  ***Event information***
***Statement***</p>
        ⇩
<p>***Statement***
  <div typeof='event:Vevent'>
    <em>Event name</em>
    ***Event information***
  </div>
***Statement***</p>
```

図 13　well-formed 処理をともなう div 要素と typeof 属性の追加

その場合，そのイベントの部分だけを特定の要素で囲む．そうするとその部分は一連の文章の中でも特別な意味を持っていることが明示的になる．そして `typeof` 属性を付与して，その情報の型を指定する．システムでは選択範囲を場合に応じて修正し，`div` 要素で囲み，`typeof` 属性を追加する機能を実装した．

図 11 はテキストノード内の一部を選択した場合であり，この場合は選択したテキスト部分を `div` 要素で囲み構造化を行う．図 12 は `div` 要素に含まれる全てのテキストを選択しているので，新たに `div` 要素で囲む必要はない．この場合は既存の `div` 要素に `typeof` 属性を追加する．図 13 では選択範囲が複数の要素にわたっている．この場合，選択範囲をそのまま `div` 要素で囲むと文書が well-formed

でなくなってしまう．そこで文書が well-formed になるように自動的に選択範囲を拡大し，div 要素で囲む．

5. システムの評価

我々は，システムの使いやすさを評価するため文献 [10] を参考にして，ユーザビリティテストを実施した．テストではシステムを用いて RDFa の利用と編集に関する一連のタスクを行い，使いにくい点を調査した．被験者は本学の理工学部に所属する学生 5 名であり，タグの存在や XHTML についての知識を持っているが RDFa の知識を全く持っていない．

テストの結果，RDFa の表示に関して問題はなかった．語彙リストの翻訳によって語彙 URI に関する知識がないユーザでも，何の弊害も無く RDFa の内容が読み取れることが分かった．また，既存の RDFa の修正に関しても問題はなかった．

しかし，RDFa の追加にはデザイン上の欠陥が見つかった．追加処理の具体的な手順を明記していなかった点である．システムには自動的に適切な処理が実行されるが，どのような選択を行えば意図する編集処理が実行されるかユーザに伝わらなかった．この結果を考慮し，システムのインタフェースに関する修正と編集手順に関する説明の追加を行った．

6. システムの利用例 - ブログでの利用

本章では RDFa の埋め込まれたブログを対象にシステムの利用例を述べる．

6.1 RDFa の表示

システムは，対象のブログを開いてブックマークから起動を行う．ブログ内に RDFa として埋め込んでいる説明，作成日，適用しているスタイルシートや作者自身に関する情報は，図 14 のように確認できる．このブログでは “塩澤 元” という人物についての情報の中に “青山学院大学” の情報が階層構造で埋め込まれており，必要に応じてそれらを展開・折畳できる．RDFa の表示の際には語彙は日常的な言葉へと自動的に置き換える．本来の語彙 URI を確認したい場合は語彙上にカーソルを合わせればよい．

6.2 RDFa の修正

ブログの中に大学キャンパスの位置情報を埋め込む．ブログの作成時には緯度と経度が分からなかったため，一時的に 0 と入力していた．システムを用いてブラウザ上でそれらを変更する．図 15 のように表示されている情報から修正対象を選ぶ．そして編集パネルから実際の緯度である 35.564 を入力して変更する．このようにしてソースコードを見ることなく RDFa の修正が可能である．

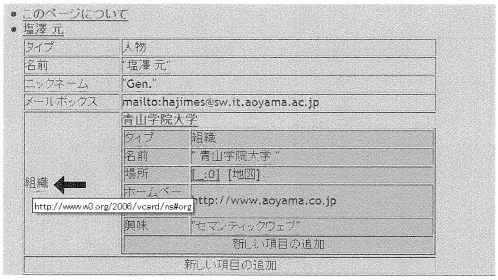

図 14 ブログ内の RDFa の確認

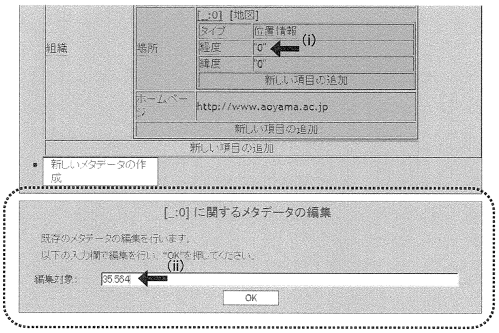

図 15 位置情報に関する RDFa の修正

```
<span about="[_:aoyama]" typeof="foaf:Organization">
  <a property="foaf:name" rel="foaf:homepage"
     href="http://www.aoyama.co.jp">青山学院大学</a>
  <div rel="vevent:location">
     <div typeof="vcard:Location">
       <span property="vcard:latitude" content="35.564"></span>
       <span property="vcard:longitude" content="139.406"></span>
     </div>
  </div>
</span>
```

図 16 位置情報の RDFa

6.3 RDFa と他のアプリケーションの連携

RDFa で表される情報は typeof 属性によって型を指定できる．その型ごとに他のアプリケーションとの連携が可能である．このブログには大学のキャンパスの位置情報を埋め込む用意があり，前節でシステムを用いて図 16 のように変更した．本システムでは vcard:Location 型に対応するアプリケーションとして Yahoo!地図(注7)を登録している．これによって，クリックだけで Yahoo!地図でキャンパスの場所が表示可能である．

6.4 新たな RDFa の追加

1 月 13 日にデジタルドキュメント研究会の発表についての記事がある．システムを用いてこの記事に対して

(注7)：http://map.yahoo.co.jp/

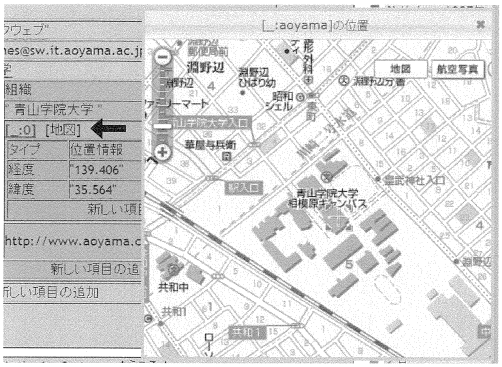

図 17　位置情報と Yahoo!地図との連携

RDFa を追加する．新しく RDFa を追加する場合には，図 18 の（i）に示すように RDFa を付与する範囲を選択する．そして自動的に well-formed になるように修正された選択範囲をシステム上で確認する．次に提示されている語彙リストの中から “イベント” を選択する．この時用意された語彙リストの中に意図する語彙がなければ，横の入力欄に直接語彙の URI を入力してもよい．これによって選択範囲が `div` 要素で囲まれて構造化され，`typeof` 属性が付与される．その後，図 19 のように概要や日時の部分を選択し，リストから語彙を選んで RDFa を付与する．

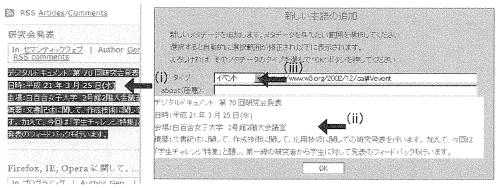

図 18　新しいメタデータの追加

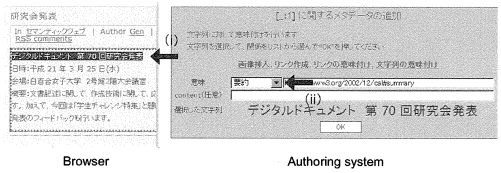

図 19　文字列の意味付け

7.　結論と今後の課題

本稿では RDFa の普及を目的とした，ユーザが簡単に RDFa を利用できるオーサリングシステムの構築について述べた．まず，RDFa を人間が確認しやすいように表示した．また，それを利用してブラウザ上で編集可能にした．それによって本システムは，RDFa を利用するためのハードルを下げることができた．それだけではなく語彙を日常的な言葉で提示することで RDFa をより積極的に付与できる環境を整え，メタデータの付与を促した．

今後の課題として，システムによって変更した DOM ツリーをファイルへ保存する方法を考えなければならない．セキュリティ上の問題から JavaScript で直接ローカルファイルを取り扱うことはできない．そこで，Wiki のようなブラウザ上で編集が可能なシステムと組み合わせる方法が考えられる．それによって，ブラウザさえあればネットワーク上のどこからでも編集が可能になり，更なる RDFa の普及が見込める．

本システムは近日中に Web 上で公開予定である．

文　献

[1] Ben Adida, Mark Birbeck, Shane McCaron, and Steven Pemberton. RDFa in XHTML : Syntax and Processing W3C Recommendation. `http://www.w3.org/TR/rdfa-syntax/`, 2008.

[2] Graham Klyne and Jeremy J. Carroll. RDF Concepts and Abstract Syntax W3C Recommendation. `http://www.w3.org/TR/rdf-concepts/`, 2004.

[3] Tim Berners-Lee. Notatoin 3 (N3). `http://www.w3.org/DesignIssues/Notation3`, 2006.

[4] Dave Beckett. RDF/XML Syntax Specification W3C Recommendation. `http://www.w3.org/TR/rdf-syntax-grammar/`, 2004.

[5] John Allsopp. マイクロフォーマット. 毎日コミュニケーションズ, 2008.

[6] James Simmons. RDF vs Microformats. `http://www.semanticfocus.com/blog/entry/title/microformats-vs-rdf-how-microformats-relate-to-the-semantic-web`, 2007.

[7] Edd Dumbill. The BBC, microformats, RDFa and Resig. `http://times.usefulinc.com/2008/06/24-uf-rdfa`, 2008.

[8] Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the sematic web. In *Proceedings of The 3rd International Semantic Web User Interaction Workshop*, 2006.

[9] Christine Frodl, Thomas Fischer, Tom Baker, and Stefanie Rühle. Deutsche Übersetzung des Dublin-Core-Metadaten-Elemente-Sets. `http://www.kim-forum.org/material/pdf/uebersetzung_dcmes_20070822.pdf`, 2007.

[10] Joel Spolsky. *User Interface Design for Programmers*. Apress, 2001.