

Refereed Conference paper

MarginNote: AJAX 動的文書解析による一般 Web ページ注釈システムの提案

馮晨[†] 沼尾雅之^{††}

本論文では、一般の Web ページに自由に注釈を書き込み、意見交換できるようにするための仕組みを提案する。システムは、プロキシサーバを介することによって、アノテーションサーバとの連携用のスクリプトを加え、Web ページ内の HTML タグに注釈表示や入力のためのマウスイベントを追加することによって実現する。本提案では、注釈ドキュメントと被注釈ドキュメント両方に対して、XML タグセットを定義し、汎用性の高いシステムとして設計した。また、AJAX による動的構文解析を導入することによって、単語レベルでの注釈付も可能になっている。提案システムは、現在、研究室内で稼働しており、このシステム構成、および、動作実績についても報告する。

MarginNote: A proposal of an annotation system with dynamic parsing by AJAX for general Web page

Chen Feng[†] and Masayuki Numao^{††}

In this paper, we introduce a method to allow a user to add the annotations to general Web pages and share them to other users. The system is achieved by the proxy server which adds the script to the Web page, and the annotations server which stores the annotations. The script adds HTML tags some mouse events which communicate with the annotation server to retrieve the annotations and show them within the page. To make the system highly portable, we design a XML tag set to the annotate document and the source document. In addition, by introducing AJAX-based dynamic parser, annotation is possible to not only paragraph level, but also to the word-level. The proposed system is currently running in our laboratory and, we also show the system configuration and the operating performance of the system.

1. はじめに

近年、WWW の発展により、ユーザがインターネットから様々な種類の情報を簡単に得ることが可能になった。しかし、インターネットは単に情報を得るだけではなく、ユーザが意見を述べたり、他のユーザと情報交換する場所でもある。このような要求を満たすために、Web 掲示板や Twitter などの Web サービスが提供されている。

現在、インターネット上で参加者が自由に意見を交換できる場として Web 掲示板があるが、一方、ニュースサイトなど通常の Web ページは情報発信のみであり、これに対しては、読者が意見を交換できる場を別に用意しなければならない。例えば、Web ページを閲覧するとき、ユーザが自分の意見を表明したい場合には、ニュース記事のページの下にユーザの書き込みを許したコメント欄を設けたり[3]、製品価格調査ページに口コミやユーザレビューを表示できるようにしたシステムが開発されている[4]。

しかし、こうした既存のコメントフィードバックシステムは、システム開発者にとっても、コメント入力用のフォームを新たに設計しなければならない、また、ユーザにとっても、ソースドキュメントとは別の場所にコメントを書かなければならないので、ドキュメントの一部を写さなければならないといった不便が生じている。こうした不便を解消するために、ソースドキュメントの任意の位置に注釈を書き込めるような Web 注釈付システムの開発が望まれている。

本研究では、インターネット上の一般 Web ページに対し、Web サーバ側の処理や、ブラウザ側の改変を必要とせず、自由に注釈を書き込み、意見交換できるようにするための仕組みを提案する。

2. Web 注釈付けシステムの現状

Web 注釈付けシステムは、A.nnotate[5]や Diigo[6]をはじめ、様々なシステムは公開され、多くの利用者に使われている。これらのシステムは、ユーザが任意の Web ページに対し、注釈の追加が可能になっている。しかし、現時点の Web 注釈付けシステムの技術特徴により[7]、システム機能はほとんどブックマークやツールバーで実現されている。このような実現手法では、特殊なブラウザ[2]やブラウザにプラグインソフトウェアのインストール [1]が必要となる。ユーザにシステム使用上の不便を与えると同時に、システムの普及が困難になる。よって、本研究では、プロキシサーバを介して、Web ページにクライアント側で実行できるスクリプトを仕込むことによって機能

[†] 電気通信大学, 情報理工学研究所
The University of Electro-Communications Graduate School of Informatics and Engineering

^{††} 電気通信大学, 情報理工学研究所
The University of Electro-Communications Graduate School of Informatics and Engineering

を実現し、ブラウザの拡張を一切しない Web 注釈付けシステムを提案する。

また、現時点の Web 注釈付けシステムは、Web ページ或いは段落に対し、注釈を付けることが可能になっている。しかし、一つの Web ページに沢山の注釈を付けると、大量な内容からユーザが注釈を付けたい部分を特定できないという問題点がある。よって、本論文では、日本語構文解析エンジンを導入することによって、Web ページの被注釈文書に対して構文解析を行い、単語レベルの注釈付けシステムを提案する。

3. Web 注釈付けシステムの提案

本研究では、現時点の Web 注釈付けシステムの問題点を踏まえ、インターネット上の一般 Web ページを注釈可能にする汎用性の高い Web 注釈付けシステムを提案する。

3.1 システムアーキテクチャ

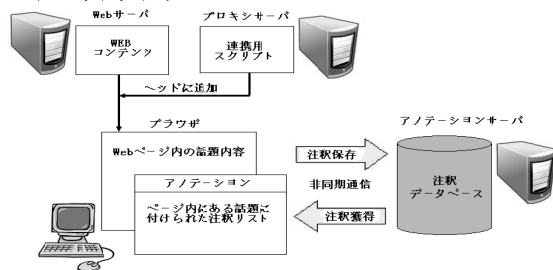


図1 システムアーキテクチャ図

提案手法は、図1のように、ソースドキュメントサーバからのHTMLに対して、プロキシサーバを介すことによって、ヘッド部分にアノテーションサーバとの連携用スクリプトを加える。そして、加えられたJavaScriptのjQueryにより、ソースドキュメントのHTMLタグに対して自動的に、その上で行われるマウスイベントに対して、アノテーションサーバと非同期通信をするAJAXタグを追加することができるようになる。これによって、ソースドキュメントサーバとは独立に設置したアノテーションサーバを利用した注釈付け機能を実現している。このアーキテクチャによって、ユーザは、あたかもソースドキュメントに対して、直接に注釈を加えているような感覚で利用することができるようになる。提案手法は以下の利点がある。

1. ソースドキュメントを保持している Web サーバ内のソースドキュメントを改変する必要がない。Web サーバとは独立に注釈付けが実現できる。

2. ブラウザの追加機能やツールをインストール必要がなく、プロキシ機能によって、システムの機能を簡単に実現できる。

3. 被注釈文に対して構文解析を行い、単語レベルで注釈を付けられる。

4. 他のユーザが入力した注釈を閲覧することによって、ユーザ間の情報共有を実現できる。

5. 注釈システムを更新する時、ブラウザ毎にアドオンを再インストールする必要がなく、プロキシサーバ内の連携用のスクリプトを改変するだけで実現できる。システム機能の拡張は容易になる。

3.2 XML タグセットの設計

提案手法はシステムの汎用性に注目し、一般 Web ページに注釈付けの機能を実現できるように設計した。Web ページの被注釈ドキュメントとユーザが追加した注釈ドキュメント双方に対し、XMLのタグセットを設計し、XML Schemaによって定義している。

J. Kahanらの研究により、注釈(annotation)のW3C標準は既に決まっている [14][15]。本研究では、このW3C標準を拡張することによって、本システムの機能拡張を実現する。さらに、注釈付け状態を管理するため、被注釈ドキュメントに対しても設計を行った。XML Schema [8]では、注釈ドキュメントを注釈(Annotation)とし、被注釈ドキュメントを話題(Post)として設計した。

注釈の標準属性は、W3C標準に基づき、以下のように定義されている。

- annotates : 注釈は被注釈ドキュメントが属するページへのリンク
- author : 注釈の作成者の名前
- body : 注釈の内容
- context : 非注釈ドキュメントの内容へのリンク
- type : 注釈が付けられた対象 (話題または注釈)
- created : 注釈が付けられた時刻
- modified : 注釈が最後に修正された時刻

被注釈ドキュメントを話題(Post)としてXMLタグを以下のように設計した(図2参照)。

```
-<Post>
  <Id>192</Id>
  <Body Resource="http://nm.cs.uec.ac.jp/post/posteddoc1.html"/>
  <Created_at>2011-05-18 16:58:33</Created_at>
  -<Annotation_info>
    <Focus_number>18</Focus_number>
    <Annotate_number>2</Annotate_number>
  </Annotation_info>
</Post>
```

図2 話題(Post)のXML例

Id : Web ページにある話題はアノテーションサーバに保存されるとき、自動的に生成された番号である。注釈に関連するとき使われている。

Body : 被注釈ドキュメントの内容である。提案システムでは、Web ページの中に HTML タグに囲まれた文書を話題と定義する。Body に直接文書を記入することも、別の文書へのリンクにすることも可能である。

Created_at : 話題はアノテーションサーバに保存された時刻である。

Focus_number : 話題に関連した注釈は閲覧された回数。

Reply_number : 話題に付けられた注釈の数。

注釈(Annotation)のW3C標準属性に基づいた拡張は以下のようになる(図3参照)。

```

- <Annotation>
- <Append_to>
  <Link> http://example.org/page.html </Link>
  - <Context>
    http://example.org/page.html#xpointer(/html/body/p[2])
  </Context>
  <Word>school</Word>
</Append_to>
- <User_info>
  <User_name>Feng</User_name>
  <User_group>Student</User_group>
</User_info>
<Type>Annotation</Type>
<Body Idref="192" Resource="http://nm.cs.uec.ac.jp/annotation/annodoc1.html"/>
<Created_at>2011-05-19 13:27:52</Created_at>
<Modified_at>2011-07-11 19:10:03</Modified_at>
<Id>275</Id>
<Mood>Agree</Mood>
</Annotation>

```

図3 注釈(Annotation)のXML例

Append_to : 注釈と話題の関連関係を示す複合属性である。被注釈ドキュメント(話題)が属するページのリンク(Link)、xpointerを用いた話題のページ内の位置情報(Context)と関連する単語(Word)から構成される。

User_info : 注釈の作成者の情報を示す複合属性で、作成者の名前(User_name)と作成者の所属グループの名前(User_group)から構成される。

Type : 標準属性と同じく注釈が付けられた対象(話題または注釈)を示す属性である。

Body : 標準属性と同じく注釈の内容である。BodyのIdref属性とは、注釈に関連した話題のId番号を示している。Bodyに直接文書を記入することも、別の文書へのリンクにすることも可能である。

Created_at : 標準属性と同じく注釈が付けられた時刻である。

Modified_at : 標準属性と同じく注釈が最後に修正された時刻である。

Id : 注釈がアノテーションサーバに保存されたとき、自動的に生成された番号で

ある。注釈に対して注釈を付けるための拡張である。

Mood : ユーザが注釈を記入するときの感想である。今後、注釈データを分析し、ユーザの嗜好抽出の参考になる拡張である。

3.3 単語レベル注釈付け

本研究では、構文解析エンジンを導入し、被注釈文に対して動的構文解析をし、単語レベルで注釈を付けられるように設計した。構文解析の結果を簡単に処理するために、結果をタグ付き文書で返す日本語構文解析エンジン[9]を導入する。たとえば、文書「庭には二羽ニワトリがいる。」に対し、以下の解析結果を得られる。

```

<na_result>
  <Total_count>9</Total_count>
  <Filtered_count>9</Filtered_count>
  <word_list>
    <word>
      <surface>庭</surface>
      <reading>にわ</reading>
      <pos>名詞</pos>
      <baseform>庭</baseform>
    </word>
    <word>
      <surface>に</surface>
      <reading>に</reading>
      <pos>助詞</pos>
      <baseform>に</baseform>
    </word>
    .....
  </word_list>
</na_result>

```

よって、処理の手順は：

1. jQuery[10]の機能により、HTML タグレベルの文をマウスイベントによって取り出す。
2. DataMapper[13]でインタフェースを提供し、AJAX[11]の非同期通信機能により、取り出されたHTMLタグレベルの文をアノテーションサーバに送り、構文解析をする。
3. jQuery UI[12]の機能により、構文解析結果のタグ(<surface>タグ)を特殊の「文書タグ」(図4の左参照)に書き換え、オーバーレイウィンドウの中で表示する。
4. 「文書タグ」とマウス動作を注目し、動作ルール(図4の右参照)に従い、単語レベルの注釈表示および注釈付けを行う。

<pre> <sentence word_count="9"> <word type="noun">庭</word> <word type="auxiliary">に</word> <word type="auxiliary">は</word> <word type="numeral">二</word> <word type="noun">羽</word> </sentence> </pre>	<pre> ((sentence),(mouseover)) → 動作1 ((word),(click)) → 動作2 ((word),(mouseout)) → 動作3 </pre>
--	--

図4 文書タグと動作ルール

4. システム構築の提案

本節では、提案システムの構成、処理の流れおよび実装の実績について説明する。

4.1 システムの構成

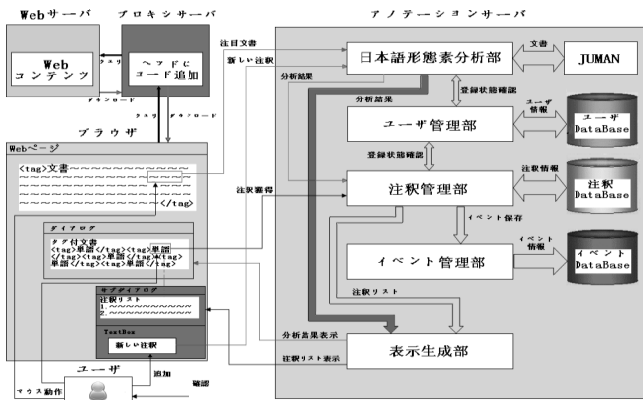


図5 システムの構成図

図5にシステムの構成を示す。本システムは、プロキシサーバとアノテーションサーバから構成される。二つのサーバを追加することによって、任意の Web ページに注釈付けすることができる機能が実現できる。

プロキシサーバの機能はユーザがアクセスしようとする Web ページのヘッドの部分に、アノテーションサーバとの連携用のスクリプトを加えることである。実現手法は、プロキシサーバにコンテンツフィルタを作り、その機能を利用し、Webサーバからダウンロードした Web ページのソースコードの中から、<head>という HTML タグを探し出し、<head>の後ろに連携用のスクリプトのファイルのアドレスを追加する。

例: <html>

<head>

```
<script type="text/javascript" charset="utf-8"
src="http://192.168.88.233:9292/js/annotation.js"></script>
```

</head>

</html>

アノテーションサーバは連携用のスクリプトファイルを提供し、データベースにアクセスことによって、ユーザのプロファイル、注釈情報、イベント情報を管理する機能を持っている。アノテーションサーバは日本語形態素分析部、ユーザ管理部、注釈管理部、イベント管理部、表示生成部から構成される。

- 日本語形態素分析部

Web ページから取り出された HTML タグレベル文に対して構文解析を行う。その分析結果を表示生成部に送る。

- ユーザ管理部

ユーザのプロファイルを管理する。本システムを利用する際、ユーザが登録する必要がある。ユーザ管理部はユーザのログイン、ログアウト機能を提供する。

- 注釈管理部

Web ページにある話題の内容と話題の位置情報を用いて、注釈データベースを検索し、話題の注釈状態情報と注釈リストを獲得し、表示生成部に送る。また、注釈管理部は新規話題の保存および新しい注釈の追加の機能を持っている。

- イベント管理部

ユーザが Web サイトをアクセスするときのページ閲覧履歴、注釈の閲覧履歴、注釈の追加履歴などのイベント情報を取得し、イベントデータベースに保存する。

- 表示生成部

注釈管理部から話題の注釈状態情報と注釈リストを受け、HTML コードを生成し、ユーザに示す。注釈情報をスムーズに表現し、システムの利用者へのフィードバックを行う。

4.2 処理の流れ

提案システムは、クライアント/サーバ型の Web アプリケーションとなっている。クライアントは、話題情報の収集と注釈の表示をする。サーバは、注釈の保存、注釈の検索及びイベント情報の管理をする。

本システムを利用する際、使用者はユーザ名とパスワードを入力し、アノテーションサーバに送り、登録を行う。その後、ブラウザ上のマウス動作に対し、クライアントに仕込まれたスクリプト (jQuery) の機能により、話題の内容と位置情報を獲得し、AJAX の非同期通信機能を用い、アノテーションサーバの日本語形態素分析部に送る。

話題を検索するときの AJAX 通信例は:

```
$.getJSON(SERVER_URL + "/api/getannotation", {
page_url : http://example.com/frontpage
tag_index : p:3
post_body : Example Document.
});
```

話題の位置情報は話題が属する Web ページのアドレスと話題を囲んだ HTML タグインデックスを含む。HTML タグインデックスは HTML タグの種類名とページ内でのインデックスから構成される。例の “p:3” とは、話題はページ内の 4 番目の <p> タグに囲まれたことを示す。このような位置情報により、話題の位置を特定できる。

日本語形態素分析部は話題情報を受け、3.3 節で述べた手順で処理を行う。処理の結果を表示生成部に送ると同時に、注釈管理部を通じて注釈データベースを検索する。もし話題がデータベースに存在しないなら、話題情報を保存し、新しい話題を追加する。

表示生成部は構文解析結果を受け、それに対して単語レベル HTML コードを追加し、ユーザに示す。ユーザは単語を「クリック」すると、注釈の検索条件を生成し、注釈管理部を通じて注釈データベースを検索する。単語に付けられた注釈を獲得すると、表示生成部に送る。また、注釈を検索するとき、検索条件を調整することによって、注釈の検索レベルを変更できる。例えば、話題の HTML タグのインデックスという条件を除けば、同じページ内の違う位置にある同じ文書に関連する注釈を閲覧できるし、ページアドレスを除けば違うページにある同じ文書の注釈を閲覧できる。したがって、注釈の検索レベルを調整することによって、ユーザの様々なリクエストを満たせる。

また、注釈を閲覧するとき、話題に新しい注釈を付けることができる。ユーザが記入した注釈の内容は日本語形態素分析部によって解析される。解析結果は注釈管理部を通じて、注釈データベースに保存される。

注釈を追加するときの AJAX 通信例は：

```
$.ajax({
  url : SERVER_URL + "/api/newanntation",
  data : {
    annotation_body : New anntation
    mood : agree
    Idref : 18
    created_at : 2011-07-12 10:28:53
    user : Feng
  }
});
```

注釈管理部を通じて注釈を保存及び検索をするとき、全ての履歴をイベントとして保存される。イベント管理部は注釈管理部からイベント情報を受け、イベントデータベースに保存する。

イベントを保存するときの Ajax 通信例は：

```
$.ajax({
  url : SERVER_URL + "/api/newevent",
  data : {
```

```
    created_at : 2011-08-03 19:07:23
    event_type : Annotation
    page_url : http://example.com/sample
    tag_index : span:2
    context_body : Sample data.
    annotation_body : Demo anntation
    mood : suggest
    user : Numao
  }
});
```

5. システムの実装

現時点では、提案システムの試作品を研究室内用のサイトで稼働しておる。この節では、システムの実行例を示し、提案の Web 注釈付けシステムの動作実績を紹介する。

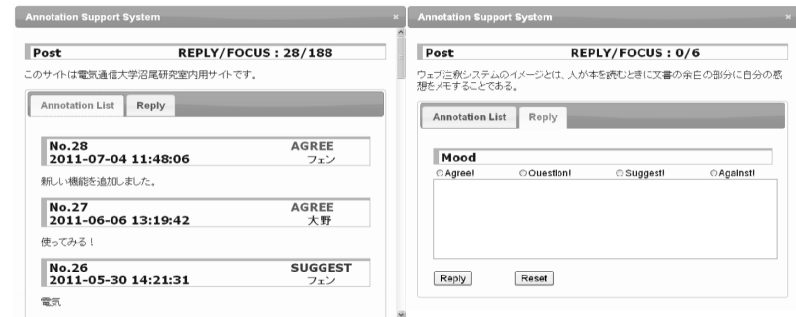


図 6 注釈の表示と追加

(1) 注釈の表示

Web ページを閲覧する時、ユーザがマウスで話題の一部を選択することによって、アノテーションサーバから話題の注釈付け状態情報と話題に付けられた注釈リストを取得し、ブラウザにダイアログボックスを生成し、図 6 の左のように示す。ダイアログボックス Annotation List ページでは、話題に付けられた注釈の数 (REPLY)、話題に付けられた注釈が閲覧された回数 (FOCUS) などの注釈付け状態情報と注釈の番号、注釈の内容、注釈が追加された時刻、作成者の名前、ユーザの感想などの注釈情報を

示している。また、ユーザに常に最新の注釈を見せるために、注釈の追加時刻の逆順で注釈リストを表示する。

(2) 注釈の追加

ダイアログボックスの Reply ページで、話題に新しい注釈を付けることができる。図6の右のように示す。新しい注釈を追加するとき、ユーザの感想と注釈の内容を入力する必要がある。本システムでは、話題に対して賛成 (Agree)、疑問 (Question)、助言 (Suggest) と反対 (Against) 計4つの感想を選ぶことができる。今後、ユーザの感想データに対して分析を行い、ユーザの嗜好抽出の重要な参考になると考えられる。

(3) 注釈情報の獲得



図7 注釈情報を示すツールチップ

提案手法は汎用性を注目し、Web ページ内容部分のソースコードの変更を避け、マウスの動作を利用し、話題の注釈付け状態を速やかに獲得する手段を提供した。ユーザは Web ページを閲覧するとき、マウスで話題を「クリック」すると、図7のようにツールチップを生成し、ユーザに話題に付けられた注釈の数や話題に最後に注釈を付けたユーザの名前とその記入時刻などの情報を表示する。

(4) 一般 Web ページに対する実験



図8 一般 Web ページにシステムの実装例

提案システムを研究室内用サイト以外の Web ページ (Yahoo ニュース) に実装し、実行例は図8のように示す。一般 Web ページに対して、提案システムの注釈付け機能を実現した。よって、提案システムの汎用性が確かめられた。

6. おわりに

本論文では、Web ページ内の HTML タグとマウスの動作を注目し、被注釈ドキュメントと注釈ドキュメント双方に対して XML のタグセットを設計し、XML Schema によって定義した。これによって、本提案以外のシステムによっても、注釈付ドキュメントの処理が可能になる。それに基づく簡単に利用可能な汎用性高い Web 注釈システムを提案した。また、提案システムの試作品の実行例を示し、その動作実績について紹介した。

今後、アンノテーションサーバに保存したページの閲覧履歴、注釈イベント、ユーザの感想データに対して分析し、ユーザの嗜好および閲覧習慣を抽出していく。また、構文的構造を導入し、動作ルールを設計し、単語レベルでの注釈付を実装する。これから XML タグセットの更新とともに、システム機能の拡張を行っていく。そして、ユーザのグループ属性を既に定義したため、これを利用し、違うグループのユーザに違う使用権限を与え、注釈付システムにアクセス制御の機能を追加していく。さらに、提案システムの使いやすさを考慮し、Web ページのオリジナルソースコードを一切変更なしの前提のもとに、マウスの動作を利用し、話題に付けられた注釈の情報を簡単に獲得できる手段を提供する。

参考文献

- 1) 駄竹栄太, 中山泰一: 選別機能を持った web 注釈システムの提案, 情報処理学会創立 50 周年記念 (第 72 回), 全国大会 (2010)
- 2) 齊藤哲生, 清光英成, 田中克己: ページ動的再構成を行う Web 注釈付けシステムの提案, データベースシステム 125-35 (2001)
- 3) <http://edition.cnn.com/>
- 4) <http://kakaku.com/>
- 5) A.nnotate, a web service for storing and annotating documents, <http://a.nnotate.com/>
- 6) Diigo, a social bookmarking website, <http://www.diigo.com/>
- 7) Web annotation, http://en.wikipedia.org/wiki/Web_annotation
- 8) XML Tutorial, W3Schools Online Web Tutorials, <http://www.w3schools.com/xml/default.asp>
- 9) 日本語形態素解析エンジン, <http://developer.yahoo.co.jp/webapi/jfp/ma/v1/parse.html>
- 10) jQuery, a new kind of JavaScript Library, <http://jquery.com/>
- 11) Dave Crane, Bear Bibeault, Jord Sonneveld, Ted Goddard: Ajax in Practice (ペーパーバック - 2007/6)
- 12) jQuery UI, <http://jqueryui.com/>
- 13) DataMapper, <http://datamapper.org/>
- 14) J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, R.R. Swick: Annotea: an open RDF infrastructure for shared Web annotations(2002)
- 15) <http://www.w3.org/2001/Annotea/>