Web Indexにおけるオーサ主導アタッチの拡張に関する研究

佐草　友也† 　遠山　元道†

† 慶應義塾大学理工学部情報工学科　〒 223-8522 横浜市港北区日吉 3-14-1
E-mail: †sakusa@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

あらまし　著者らは, キーワードと URL の組み合わせであるエントリの集合が記述された WIX ファイルを用い, Web ページ内の文章に出現するキーワードに対して, それに対応するハイパーリンクを生成する Web Index (WIX) システムを開発している. 従来, WIX システムによるリンクの生成は主にユーザ主導で行われていた. 本研究では, ユーザが Web ページを閲覧する際, Web オーサがアンカータグを記述すること無く指定したリンクを自動的に生成する, オーサ主導のリンク生成における新手法を提案し, 実装した.

キーワード　Web Index, Web 情報システム, Web, Web コンテンツ, ハイパーリンク

1. はじめに

近年, インターネットの普及により Web 上の情報検索が非常に増え, ユーザは検索エンジンによる検索を行うことで必要な情報を得ることが可能である. しかし結果として得られた Web ページの閲覧中に, そのページに含まれる情報の更に詳細な情報を得たいという要求が生まれると, 従来ではその単語をさらに検索エンジンにかけ情報の再検索・選択を行わなければならない. Web ページ作成者は関連する複数の情報をハイパーリンクによって結合することが可能だが, これは Web ページ作成者が意図した関連のみで結合され, 必ずしも閲覧するユーザのニーズに応えているとは言えない. そこで, 著者らは Web における情報資源結合を実現するために, Web Index (以下 WIX とする) と呼ぶ情報資源表現形式の提案, 開発を行っている.

WIX ファイルと呼ばれる, キーワードとそれに対応する URL のペアの集合を XML 形式で記述したファイルを用意し, それを閲覧中の Web ページに結合 (アタッチ) することで, 閲覧ページ中のキーワード部分が自動的に対応する URL へのハイパーリンクに変換される. ページを閲覧しているユーザ自身が任意のページに対してアタッチすることで新たにハイパーリンクを生成する, このような利用者主導による情報資源結合方法を WIX ではユーザ主導アタッチと呼んでいる. 従来, ユーザ主導でアタッチを行う研究が積極的に行われていた.

本論文では, クライアントサイドでユーザがアタッチボタンをクリックすることでリンク生成が行われるユーザ主導のリンク生成方法と対を成す, Web オーサがアンカータグを記述すること無く, Web ページにどの WIX ファイルを用いるか明示することによってサーバサイドでリンクを自動的に生成する, オーサ主導のリンク生成における新手法を提案し, 実装した. 提案手法によってリンクが生成された Web ページはユーザが普段閲覧している Web ページと何ら変わりない.

本論文の構成は以下の通りである. まず, 2 章で本論文の研究目的について述べる. 3 章で WIX の概要について述べる. 4 章で WIX アタッチエンジンにおける各処理の概要を述べる. 5 章で提案手法について説明する. 6 章, 7 章で評価・まとめを行う.

2. 研究の目的

従来, WIX は主にユーザ主導でハイパーリンクを生成するための情報資源形態であった. 一方, ユーザが閲覧する Web ページを提供する Web ページ作成者側, つまり Web オーサ主導による WIX ファイルを用いたリンク生成を行う研究も行われていた. 従来の方法は, 1 つの Web ページに関し, HEAD 部分にメタ情報として WIX ファイルを明示することでその Web ページに WIX ファイルを適用させる「直接的」なオーサ主導であった. しかし多数の Web ページ毎に WIX ファイルを明示することは冗長かつ面倒である.

また, 近年「Closed Web Index」の実現も目指している. 大学や企業のようなコミュニティにおいて, 外部へのリンクを生成するよりも内部リンクを生成することの方が需要がある. そのため狭いコミュニティでは, そのコミュニティ内で有用な, 関連の深いリンクを生成することが求められる.

上記の 2 点を考慮し, オーサがアンカータグを記述することなく, 複数の Web ページに有用な WIX ファイルを適用することによるハイパーリンクの生成が可能になることで, Web オーサのリンクを記述する負担を軽減すると共に, WIX の新たな側面であるオーサ主導の発展を目的とした. 内部リンクの生成は Wikipedia や, はてなダイアリーで既に実施されているが, 著者らはより汎用性に富んだものを目指し開発を行っている [5] [6].

3. WIX

3.1 WIX ファイル

WIX ファイルは XML 形式で記述されたキーワードと URL の組み合わせであるエントリの集合であり, 記述例は図 1 のようになる. エントリ内には, キーワードとなる見出し語を keyword 要素に格納し, それに対応する詳細情報を示す参照先の URL を target 要素に格納する. また, WIX ファイルには, header 要素にファイル概要, 作者コメントなど, その WIX ファイル全体についてのメタ情報を格納することも可能である.

3.2 Bookmark

ユーザ主導で WIX システムを使用する場合, ユーザは予め

```
<?xml version="1.0" encoding="UTF-8"?>
<WIX>
    <header />
    <body>
        <entry>
            <keyword>慶應義塾大学</keyword>
            <target>http://www.keio.ac.jp/</target>
        </entry>
        <entry>
            <keyword>日吉</keyword>
            <target>http://www.hc.keio.ac.jp/</target>
        </entry>
        <entry>
            <keyword>矢上</keyword>
            <target>http://www.st.keio.ac.jp/</target>
        </entry>
    </body>
</WIX>
```

図 1　WIX ファイルの記述例

目的に適った WIX ファイルをブックマークしておく必要がある [1]. 図 2 のように, ページ下のツールバー上にアタッチボタンが存在する. このアタッチボタンはブックマークに対応し, ボタン一つに対して複数の WIX ファイルを登録できる.

3.3　ユーザ主導アタッチ

WIX システムのクライアントサイドは, FireFox add-on や Chrome Extension によって実装されている [1]. 図 2, 図 3 は Chrome Extension の例である. アタッチボタンをクリックすることで, 図 2 の記事にはなかったハイパーリンクが図 3 の記事には生成される. このハイパーリンクを生成することをアタッチと呼び, ユーザによって新しくリンクの生成が行われるため, これはユーザ主導によるアタッチである. アタッチすることで, WIX ファイル内の target タグに記述されている URL と結合されたことになる.

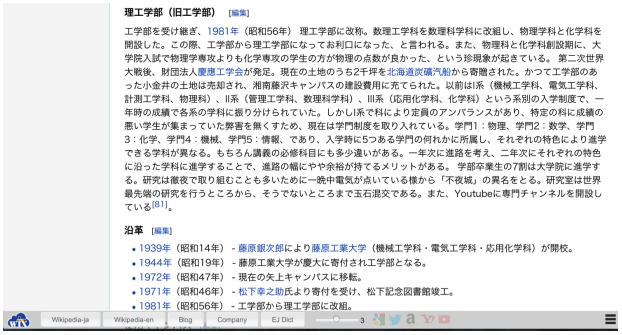

図 2　Chrome Extension(WIX ツールバー)

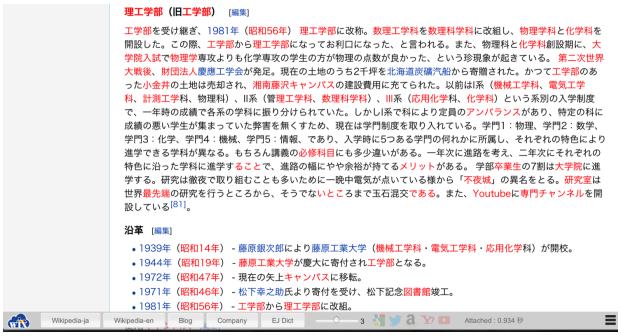

図 3　Chrome Extension(アタッチ)

4.　WIX アタッチエンジン

4.1　FSDR 処理

システムの Web 資源結合動作であるアタッチ処理について述べる.

アタッチ処理は以下の 4 フェーズに分けられる. この一連の流れを FSDR 処理とする.

- Find 処理
- Select 処理
- Decide 処理
- Rewrite 処理

4.1.1　Find 処 理

Find 処理では Web 文書と WIX ファイル集合を入力として, 全 WIX ファイル中の全てのエントリのうち文書中に存在するキーワードを持つエントリを Web 文書中の出現位置とセットにして返す. このセットの集合のことを Find 結果と呼ぶ. Find 結果はその出現位置によってソートされ, 次処理に引き継がれる.

4.1.2　Select 処理

Select 処理では Find 結果とユーザのブックマーク情報を入力として, Find 結果からユーザのブックマークしている WIX ファイルのエントリのみに絞り込む. さらに, 同一出現位置の Find 結果の要素に対しては最長一致をとる.

4.1.3　Decide 処理

Decide 処理では Select 結果から更にキーワードの周辺文字列を利用して, より Web 文書のトピックとマッチするエントリを抽出するなどの研究がされている.

4.1.4　Rewrite 処理

Rewrite 処理では Decide 結果を用いて入力 Web 文書を新たなハイパーリンクのついた Web 文書に書き換える. これによって, アタッチが完了する.

5.　間接的オーサ主導アタッチ

5.1　WIX のアタッチ形式

WIX のアタッチ形式は大きく分けてユーザ主導アタッチとオーサ主導アタッチの 2 種類存在する. ユーザ主導はユーザ自身が閲覧中の Web ページに対してブックマークしたアタッチボタンをクリックすることで新たにハイパーリンクを生成する, クライアントサイドのアタッチ形式である. 一方, オーサ主導によるアタッチはユーザが普段通りに Web ページを閲覧するが, その Web ページはリクエストを受け付けるサーバ群を管理するオーサ達によって指定された WIX ファイルによるサーバアタッチが行われたものである. 現在のオーサ主導アタッチは Web ページ毎にアタッチに使用する WIX ファイルをその HTML 内に明示することで行われる.

5.2　実現による需要

提案手法によるオーサ主導アタッチの目的を図 4 を用いて示す. 図 4 は慶應義塾大学の各キャンパスで管理している Web ページをユーザが閲覧する場合を想定している.

大学のような狭いコミュニティでは, コミュニティ内を遷移する内部リンクが主に生成されると思われる. WIX のオーサ主

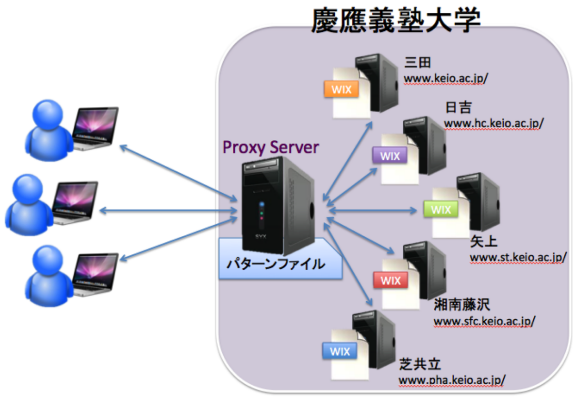

図 4　提案手法によるオーサ主導アタッチ形態図

導アタッチを用いた場合, その内部リンクの URL とキーワードのペア群を WIX ファイルとして作成するだけでよい. そのため狭いコミュニティ内でリンクを生成するのにオーサ主導アタッチは非常に相性が良い. だがオーサ主導アタッチを使用するにしても, そのコミュニティ内で, あるページには A という WIX ファイルを使用したアタッチを, このディレクトリ下の全てのページには B と C の 2 つの WIX ファイルを使用したアタッチを行いたいという状況は当然考えられる. 例えば図 4 のように慶應義塾大学内において, 慶應義塾大学全体に等しく関連のある情報や, 各キャンパス毎に異なる情報, いくつかのキャンパスに共通する情報などが存在する. それらの情報に関してリンクを生成する場合, 多数の WIX ファイルを扱って多数の Web ページに Web ページ毎にアタッチに使用する WIX ファイルを明示するのは面倒である. そのため Web ページ単位でアタッチに使用する WIX ファイルを明示するのではなく, もっと柔軟にしなければオーサ達はオーサ主導アタッチを使用してもあまり嬉しくないだろう.

そこで, 著者らは提案手法を実現するに至った. 提案手法は, そのコミュニティで必要となる全ての WIX ファイルと, アタッチ対象ページをまとめたパターンファイルを用意しておく. このパターンファイルを用いて, オーサがアンカータグを記述することなく Web をブラウジングしているユーザにアタッチ済みの Web ページをレスポンスとして返すことが可能になればオーサ達にとって大変有用なものであるに違いない. また, URL の階層毎に, または特定のページにだけ異なる WIX ファイルを用いたアタッチを行うこともでき, 先程図 4 を例に述べた柔軟なオーサ主導アタッチが可能になった.

5.3　概　　要

従来のオーサ主導アタッチの考えは, 各 Web ページに WIX ファイルを明示することでその WIX ファイルをアタッチに使用するといった, オーサによる直接的なアタッチであった. 図 5 のように HEAD 部分にメタ情報として WIX ファイルを明示することでオーサ主導アタッチを行う. 多数の Web ページをサーバが所持している場合や, 大学など狭いコミュニティでクローズドな情報を提供するにはこのような記述方法では結果冗長になってしまう. そこで提案手法では, ユーザからリクエストを受け付けるサーバに, URL のパス名とデータベース上で管理している WIX ファイルを表す ID(以下 wid とする) のペア群で構成されたパターンファイルを 1 つ置く. これにより, 複数の Web ページに WIX ファイルを関連付け, さらにこれらを階層的に適用させたアタッチを行うことが可能になり, レスポンスとして図 2.3 のようにアタッチ済みの Web ページをユーザに返すことになる. HTML に対してアタッチに使用する WIX ファイルを「直接」明示する従来の手法と比較し, 提案手法を間接的オーサ主導アタッチと呼ぶ.

```
<HTML>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <meta name="wix" content="5, 128"/>
  <title>
   直接的オーサ主導アタッチ
  </title>
</head>
```

図 5　従来のオーサ主導アタッチを行うための記述方法

5.4　実 現 形 態

本論文において間接的オーサ主導アタッチを実現したが, 集中型と分散型の 2 種類の方法が考えられる. 図 4 のようにパターンファイルをプロキシサーバに置き, そのサーバがバックグラウンドに存在する実際のリクエスト先のサーバ群の代わりに処理を行う集中型と, パターンファイルはプロキシサーバに置くが, バックグラウンドに存在する各サーバが処理を行う分散型である.

集中型は 1 台のプロキシサーバにパターンファイルを置き, バックグラウンドのサーバでビルドや設定を行うことなくそのサーバで処理を行うことが可能だが, 処理を一箇所で行うためボトルネックになる場合があるというデメリットが存在する. 一方分散型は実際にリクエストを受け付けるサーバごとにビルド・設定を行わねばならないが, サーバへの負担は軽くなる. 提案手法では分散型の間接的オーサ主導アタッチのみ実現しており, 集中型は今後実装を行う予定である. 分散型では, 各サーバへ配布されるパターンファイルは全て同一のものであるようプロキシサーバで管理を行う. 各サーバはユーザからのリクエスト情報を取得するため, パターンファイルに必要な事柄は, リクエストを受け付けるサーバのホスト名とリクエスト URL のパス名であることが分かる.

5.5　システム内部仕様

間接的オーサ主導アタッチを実現するためのアプリケーションは全て Java で実装した. 以下にシステム内の処理の流れを示す.

最初にユーザからリクエストを受け取った Web サーバにて, アプリケーションサーバへのリバースプロキシを行う. 次にそのリクエスト情報を基にアプリケーションから内部的にリクエストページを取得する一方, wid の決定を行う. その後, Web ページと決定済み wid によるアタッチを行い, ユーザにアタッチ済み Web ページを返す.

wid の決定は, リクエスト URL のホスト名とパス名を基に, オーサが用意したパターンファイル内の URL パスパターンと

照らし合わせることで行う．間接的オーサ主導アタッチの核となるパターンファイルの構成から wid 決定の流れを次節で述べる．

5.6 パターンファイル

5.6.1 ホスト名による URL パスパターン群の決定

第一にリクエスト URL のホスト名とパターンファイル内のホスト名の照合を行う．パターンファイルは< >内にホスト名を, そして異なるホスト名の出現までが, このホスト名のサーバが受け付けるリクエスト URL のパスパターンと wid を表す数字のペア群で構成される．リクエスト URL からマッチしたホスト名におけるペア群が wid 決定のための候補となる．図 6 は慶應義塾大学のサーバに関するパターンファイルの構成例である．リクエスト URL のホスト名が www.keio.ac.jp である場合, www.hc.keio.ac.jp までに記述された URL パスパターンと wid を表す数字のペア群が wid 決定の候補となる．

```
<www.keio.ac.jp>
        /index-jp.html : 1
        /ja/student/index.html : 2
<www.hc.keio.ac.jp>
        "/prs/.*" : 3, 4
<www.st.keio.ac.jp>
        /students/index.html : 5
```

図 6　パターンファイルの記述式

5.6.2 URL パスパターンと wid

URL パスパターンの記述方法は 2 種類存在する．

（1） パスパターン

URL パスパターンを記述する場合, < >内のホスト名に続く URL パスとして認識する．ただし, Linux コマンドで扱われる＊や?のようなワイルドカードの使用も可能となっている．図 6 を例に取って説明する．リクエスト URL が http://www.keio.ac.jp/index-jp.html である場合, ホスト名に続くパスが/index-jp.html であるため, 1 つ目のパターンである/index-jp.html は適合する．一方, 2 つ目のパターンはホスト名に続くパスと一致しないため不適となる．

（2） 正規表現パスパターン

” ”内に URL パスパターンを記述した場合, リクエスト URL パスのどこに出現した場合でもマッチングが行われるよう正規表現を用いた URL パスとして認識する．図 6 を例に取って説明する．リクエスト URL が http://www.hc.keio.ac.jp/ja/prs/students.html である場合, ” ”内の正規表現パスパターンは「prs ディレクトリ以下の 0 文字以上の任意の文字列」と読み取ることができるため, このパスパターンはリクエスト URL パスに適合する．

5.6.3 wid の決定

上記のパスパターン記述方法に従って記述したパスパターンが適合することで, そのパスパターンとペアである wid を選出する．パスパターンに対応する wid を複数用意することが可能だが, 先に記述されたものが優先される．また, 複数選出された wid に優先順位を付けることで, ユーザが閲覧する Web ページに有用な WIX ファイルを適用したアタッチを行う．例として, 図 7 と図 8 のようにパターンファイルが構成され, http://www.hc.keio.ac.jp/ja/prs/students.html のようなリクエスト URL を受け取った場合の wid 決定の流れを説明する．

図 7 の場合, パターンファイル内のホスト名とリクエスト URL のホスト名が適合するのは www.hc.keio.ac.jp であるため, wid 決定の候補に 3 つの URL パスパターンが選出される．これらのどの URL パスパターンもリクエスト URL パスに適合するため, 選出される wid は 2, 3, 4, 5 となる．ただし, ユーザが閲覧する予定である students.html に対して, より近傍のディレクトリがその Web ページにより関連が深いと考え, 優先順位を考慮した場合の wid は 2, 4, 5, 3 と決定される．

同様に図 8 の場合, wid 決定の候補に選出された 3 つの URL パスパターン /ja, ”/prs/. ＊”, ”/students.html” のどれもがリクエスト URL パスに適合する．ただし 2 つ目と 3 つ目のパスパターンで優先順位を考慮すべきである．「prs ディレクトリ以下に存在する students.html」と「students.html」を比較した場合, 前者がより詳細なパス情報を与えることから, 最終的な wid は 3, 4, 5, 2 と決定される．

このように, ユーザが閲覧する予定である Web ページに対して関連が深く, より詳細な情報を与えるパスパターンを優先することで, 階層的に WIX ファイルの適用が可能になる．また, このような階層的な WIX ファイルの決定がユーザに有用なアタッチ結果を与えるために重要であると言える．

```
<www.keio.ac.jp>
        /index-jp.html : 1
<www.hc.keio.ac.jp>
        /* : 2
        /ja : 3
        "/prs" : 4, 5
<www.st.keio.ac.jp>
        /students/index.html : 6
```

図 7　パターンファイルの例 (1)

```
<www.keio.ac.jp>
        /index-jp.html : 1
<www.hc.keio.ac.jp>
        /ja : 2
        "/prs/.*" : 3
        "/students.html" : 4, 5
<www.st.keio.ac.jp>
        /students/index.html : 6
```

図 8　パターンファイルの例 (2)

5.7 静的リンクの付与

4.1 小節で述べたように, アタッチ処理は 4 つのフェーズに分けられている. ただしこれらの処理は主にユーザ主導アタッチの発展を目的に開発が行われているため, 4.1.4 小節で述べた Rewrite 処理の変更が必要である.

従来の Rewirte 処理では, ユーザがハイパーリンクをクリックする際に自身がアタッチを行った WIX ファイルの target 要素である URL にリダイレクトするようハイパーリンクに情報を持たせている. そのためリンクにマウスオーバーした場合「javascript:void(0)」が表示される. アタッチ済みの Web ページをユーザに提供する場合, このように Rewrite 処理を行ってしまうとユーザがリンク先の URL 情報を認識することが不可能である. 本提案手法では Rewrite 処理の際, 事前にアタッチされる各単語に対して WIX ファイルの target 要素である URL をアンカータグに付加するように, 静的リンクへの書き換えを行うよう実装した. ユーザは普段閲覧している Web ページと何ら変わりなくアタッチ済み Web ページを閲覧することになる. また, 現在 WIX では同じキーワードに対して target 要素が異なるエントリが存在する場合には複数のリンク先をポップアップで表示している. こうすることで, ユーザ自身が遷移したいリンク先を決定することが可能である. ただし提案手法では適用する WIX ファイルに優先度を付けている. そのため, オーサがユーザに提供したいリンクを静的リンクで付与する.

6. 評　　価

6.1 実験・評価方法

間接的オーサ主導アタッチの有用性を評価するため, Body サイズ増加に伴う 1 リクエストに費やす応答時間を普段 Web ページをブラウジングする場合と比較, パターンファイルにおける URL パスパターン増加に伴う 1 リクエストに費やす応答時間, 同時アクセスユーザ数増加に伴う平均応答時間を普段 Web ページをブラウジングする場合と比較, 計 3 項目について実験を行い, 評価を行った. 実験結果を示す図及び表において, リクエストを受け取ってから wid の決定とリクエストページの取得まで行い, アタッチ機構へパラメータを送信せずに, 取得したリクエストページをそのまま出力した指標を「提案手法アタッチなし」とした. 提案手法によるアタッチ済みページをレスポンスとする指標が「提案手法アタッチあり」である.

6.2 Body サイズ増加に伴う応答時間

6.2.1 実 験 条 件

アタッチの対象となるキーワードを 100 個含む HTML ファイルをリクエスト対象とし, アタッチに使用する WIX ファイルは Wikipedia.wix のみとした. 比較対象は普段通りに Web ページをブラウジングする際の平均応答時間とした. この実験ではキーワード数 100, リクエストを送るユーザ数 1 人に固定し, リクエスト対象の Body サイズを 100KB から 1MB まで変化させた場合に 1 リクエスト当たりの応答時間の変化を測定し, 比較を行った.

6.2.2 実 験 結 果

図 9 は, 対象 Web ページの Body サイズ増加に伴う 1 リクエスト当たりの応答時間を示している. 図より, 提案手法における応答時間は Body サイズが増加するにつれ, どちらもほとんど線形に増加していることが分かる. 1MB の HTML ファイルを対象とした場合でも 840msec でアタッチ済みのレスポンスが返ってくるため, ユーザがページのロード時に不快感を抱くことは無いだろうと言える. 「提案手法アタッチあり」では比較対象より Body サイズ 300KB までは約 1/3 倍, 400KB から 1MB までは約 1/5 倍の応答速度となっている. アタッチ対象となるキーワードを固定しているためアタッチの Rewite 処理に時間を使っているのではなく, 特に全 WIX ファイル中の全てのエントリのうち文書中に存在するキーワードを持つエントリを探しだす Find 処理に時間を費やしていると思われる. また, Body サイズの増加に伴いアタッチ済みの新しい Body の生成に時間を費やしているとも考えられる. 提案手法ではアタッチ機構は従来のものを使用しているため, 提案手法における評価として優先すべき指標は「提案手法アタッチなし」の方である. 「提案手法アタッチなし」は比較対象より約 1.35 倍の時間を費やすが, 1MB で 260msec なので許容であると言える.

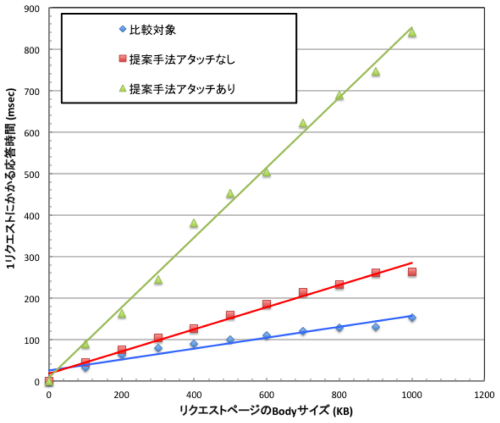

図 9　Body サイズ増加に伴う 1 リクエスト当たりの応答時間の比較

6.3 パターンファイル内のパスパターン増加に伴う応答時間

6.3.1 実 験 条 件

6.2 小節で使用した同じ HTML ファイルをリクエスト対象とし, 今回は Body サイズを 100KB とした. この実験ではリクエストページの Body サイズとリクエストを送るユーザ数を 1 人に固定し, パターンファイル内の URL パスパターンを 50 個ずつ増加させることでアタッチに使用する WIX ファイルの決定に費やす時間の変化と, 1 リクエスト当たりの応答時間を測定し, 比較を行った. 増加させる 50 個の URL パスパターンの内訳は, リクエスト URL パスにマッチする URL パスパターンを 40 個, 残りはマッチしないものとした.

6.3.2 実 験 結 果

図 10 は, パターンファイル内の URL パスパターン増加に伴う 1 リクエスト当たりの応答時間を示している. 図より, 提案手法におけるどちらの指標においても URL パスパターン増加に伴う応答時間の増加はほとんど見られない. 「提案手法アタッチあり」指標では URL パスパターン数 100 個までに若干の増

加が見られるが, これはアタッチに使用する WIX ファイルの指定が原因であると考えられる. WIX ファイル内のキーワードと URL のペアであるエントリ群は各 WIX ファイルによって数が異なるため, 著者がエントリ数の多い WIX ファイルを指定してしまったことが応答時間増加を生んだと言える. そのため, 「提案手法アタッチなし」指標では URL パスパターン数が増加しても応答時間はほとんど一定である. 前述の実験と当実験から, 「提案手法アタッチなし」指標が比較対象と比べて応答時間を費やしている原因は, 内部的にリクエストページを取得している部分と考えられる.

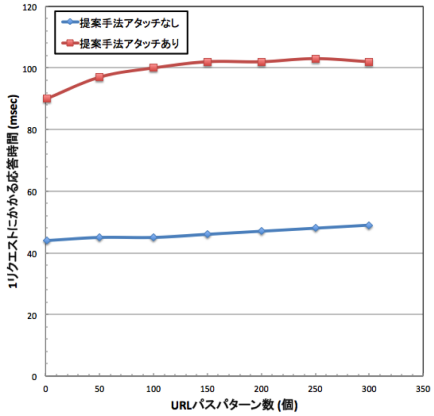

図 10　URL パスパターン増加に伴う 1 リクエスト当たりの応答時間の比較

6.4　同時アクセスユーザ数増加に伴う平均応答時間

6.4.1　実 験 条 件

6.3 小節で使用した HTML と同様, キーワード数 100 個含む Body サイズ 100KB の HTML ファイルをリクエスト対象とし, アタッチに使用する WIX ファイルは Wikipedia.wix のみとした. 比較対象は普段通りに Web ページをブラウジングする際の平均応答時間とした. この実験ではキーワード数と Body サイズを固定し, 同時にリクエストを送るユーザ数の増加に伴い, 1 リクエスト当たりに費やす平均応答時間を測定し, 比較を行った. 本実験により, 提案手法における実装部分の脆弱性が明らかになる.

6.4.2　実 験 結 果

表 1 は, 提案手法を実装したサーバに同時にリクエストを送るユーザ数が増加することによる平均応答時間の変化を示している.

表 1　同時アクセス数増加に伴う 1 リクエスト当たりの平均応答時間

ユーザ数 (人)	比較対象 (msec)	提案手法アタッチなし(msec)	提案手法アタッチあり(msec)
100	263	308	419
200	798	963	1192
300	1287	1784	2304
400	1884	2406	2904
500	2496	3309	3826

表 1 より, どの指標においても同時アクセスユーザ数が増加しても平均応答時間は線形に増加していることが分かる. 結果として, 同時アクセスユーザ数が 500 人に達した場合でもアタッチ済みページをレスポンスとして返すのに約 4sec, これはユーザが不快に感じることはまずないと思われる.「提案手法アタッチあり」指標において, もしこのまま線形に増加するのであれば同時アクセスユーザ数が 1000 人で平均応答時間が約 8sec となる. この仮定も踏まえて, 提供手法の実装はスケーラビリティが高いと言える.

ただし, 提案手法におけるどちらの指標もそれぞれ約 1.3 倍, 1.5 倍平均応答速度が比較対象より遅くなっている点は改善すべきである. 実験条件としてアタッチに使用する WIX ファイルは 1 つとしており, wid の決定に時間を費やしているとは考えにくい. 本実験におけるユーザのリクエストは逐次アクセスではなく同時アクセスである. そのため, 「提案手法アタッチあり」指標における平均応答速度低下の主な原因は, ユーザ数の増加に伴ってアタッチ済みのページを新しく生成する時に遅延が発生していると考えられる. また, 「提案手法アタッチあり」及び「提案手法アタッチなし」指標, どちらも比較対象より平均応答速度が遅い原因は, 提案手法では JavaAPI の HttpURLConnection クラスを用いてリクエストページを取得しており, ネットワーク上の遅延が影響していると考えられる.

7.　お わ り に

本論文では, Web オーサがアンカータグを記述すること無く指定したリンクを自動的に生成する, オーサ主導アタッチの新手法を提案し, 実装した.

URL パスと wid のペア群を記述したパターンファイルを, ユーザからリクエストを受け付けるサーバに 1 つ置くだけで, ユーザが普段閲覧しているような Web ページを提供することが可能となった. また, Web オーサが多数のリンクを記述する負担を軽減すると共に, WIX ファイルを利用することで, 古い Web ページから新しい Web ページへ遷移できる新しい Web の形態となった Web ページをユーザへ提供がすることが可能となった.

パターンファイルの 2 種類の記述方法に従い, 容易に新しく Web ページにリンクを生成することが可能であると共に, 正規表現を用いた複雑な設計をすることで URL の各階層に異なる WIX ファイルを適用させることも実現した. その結果として, ある狭いコミュニティにおいてグローバルな情報をユーザに提供するのではなく, そのコミュニティ内で有用な WIX ファイルを適用した Web ページの提供が可能となり, 「Closed Web Index」の実現へ近づいたと言える.

実験・評価では, 提案手法におけるユーザへの応答速度を様々な側面から測定, 比較を行った. 実験結果から, 提案手法による間接的オーサ主導アタッチによる Web ページをユーザが閲覧する際, その応答速度が許容範囲内であることが示された. ただしリクエストページの Body サイズの増加による応答速度の低下を改善することが今後の一番の課題である.

参 考 文 献

[1] 林 昌弘, 青山 峻, 朱 成敏, 遠山 元道. Keio WIX シス テム (1) ユーザインターフェース. データ工学ワークショップ, DEIM2011. 2011.

[2] 森　良介, 藪　達也, 朱　成敏, 遠山　元道. Keio WIX システム (2) サーバーサイド実装. データ工学ワークショップ, DEIM2011. 2011.

[3] 小須田達哉, 松崎 智子, 遠山　元道. WebIndex アタッチエンジンの大規模分散構成. データ工学ワークショップ, DEIM2012. 2012.

[4] 石崎文規, 遠山　元道. 大規模 Aho-Corasick オートマトンにおける追加更新手法の提案. データ工学ワークショップ, DEIM2012. 2012.

[5] http://ja.wikipedia.org/wiki/Wikipedia:記事どうしをつなぐ

[6] http://hatenadiary.g.hatena.ne.jp/keyword/リンクを簡単に記述する (http 記法、mailto 記法)