DEIM Forum 2013 B10-5

Web 文書にアタッチ可能な未選択 WIX ファイルの推薦システム

牟田　拓広† 　遠山　元道 ††

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

あらまし　著者らはユーザ主体の Web 情報資源結合を実現する Web IndeX(WIX) システムを開発している．WIX システムとは, キーワードとそれに対応する URL からなるエントリを持つ WIX ファイルを用いることで Web ドキュメントに対する Web 資源結合サービスを実現するシステムである．本研究では, ユーザがアタッチした Web 文書に対し, 他にもアタッチ可能な WIX ファイルを提示, 推薦するシステムを提案する．

キーワード　WIX, Web 情報システム, Web, ユーザ支援

Recommender System for Attachable WIX file to a Web document

Takuhiro MUTA† and Motomichi TOYAMA††

† †† Department of Information and Computer Science, Keio University
Hiyoshi 3–14–1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223–8852 Japan
E-mail: †muta@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

1. はじめに

著者らは Web における利用者主導によ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 ファイルとは, キーワードとそれに対応する URL をペアとするエントリの集合であり, XML 形式で記述される．それを閲覧中の Web 文書に結合 (アタッチ) することで, Web 文書中のキーワードが対応する URL へのハイパーリンクに変換される．

この WIX ファイルはライブラリマネージャを通してライブラリ DB と WIX DB に登録され, ユーザは WIX DB マネージャを通して使いたい WIX ファイルを検索してブックマークに登録したり, 編集する．しかし, ユーザは一々WIX ライブラリにアクセスするわけではなく, 一度ある WIX ファイルをブックマークに登録したら, その他にもアタッチできそうな WIX ファイルを探すとは考え難い．また, 今後 WIX ファイルが増えていくに連れて, 本当に目的の WIX ファイルが見つからなくなる可能性もある．これはユーザにとっては非常に不便である．

そこで, 本研究ではユーザが WIX ファイルをアタッチした際に, 他にもアタッチが可能な WIX ファイルを紹介・提示し, その新たな WIX ファイルでアタッチテストをしたり, それを元に新たにブックマークを作成する機能を提案する．

WIX アタッチエンジンは Find 処理 →Select 処理 →Decide 処理 →Rewrite 処理という処理の流れを持つ．WIX アタッチエンジンでは, 全ての WIX ファイルからエントリ情報を抽出し, Aho-Chorasick 法によるオートマトン（Find インデックス）を構築し辞書式マッチングを行なっている．以下このマッチング結果を Find 結果と呼ぶ．この Find 結果からユーザのブックマークしたエントリのみを選択した結果を Select 結果とする．本手法では, Find 結果から Select 結果を除いた UnSelect 結果を用いて利用できる WIX ファイル群を抽出するユーザ支援機能を実装している．

本論文の構成は以下の通りである．2 章で WIX の概要について述べる．3 章で提案手法について説明する．4 章で WIX アタッチエンジンにおけるそれぞれの処理の概要について述べる．5 章で WIX アーキテクチャについて説明する．6 章, 7 章で評価 · まとめを行う．

2. Web IndeX(WIX)

WIX の概要は以下の通りである．

2.1 背　　景

近年, Web の普及と共に人々は検索エンジンを利用して情報検索を行うようになった．ユーザは情報を得たい単語を検索エンジンに入力し, その結果を出力した Web 文書中から必要な情報得るのが一般的である．したがって, ユーザが Web 文書中の単語に対して新たな情報を得たいという要求が生じた場合, ユーザが更にその単語を検索エンジンなどにかけなければならない．このような単語が複数存在する場合, ユーザは何回も検索エンジンを使わねければならず, かなりの負担になってしまうと考えられる．

2.2 WIX ファイル

WIX ファイルは文章中のキーワードとそれに対応する URL からなるペアを一つのエントリとしたものの集合である．実際には，図 1 のような XML 形式で記述されている．キーワードを keyword 値に，URL を target 値に格納し，それらを一つとして entry タグで囲う．

```
<WIX>
  <entry>
    <keyword>XXX</keyword>
    <target>http://www.XXX.com</target>
  </entry>
  <entry>
    <keyword>YYY</keyword>
    <target>http://www.YYY.co.jp/YYY.html</target>
  </entry>
  ...
</WIX>
```

図 1　WIX ファイル記述方式

2.3 アタッチ

WIX システムでは関係データベースの結合演算の考え方を Web の世界に持ち込むことによって，ユーザ主体の Web 情報源の結合を図っている．結合演算の関係モデルでは，それ以前のデータモデルがアドレス / ポインタによって行っていた関連付けを，値に基づいて主キー，外部キーで実現した．現在の Web におけるアンカー /URL は関係モデル以前の DB モデルと酷似している．そこで WIX ではアンカーテキストとリンクを Web ドキュメントから独立した WIX ファイルという情報源として保存し，それを適宜ドキュメントに対して結合を行うことによってハイパーリンクの生成を行う．この結合操作をアタッチと呼ぶことにする．

例えば，閲覧中の Web 文書中にある英単語について英和辞書の情報を得たい場合，従来なら英単語一つずつ検索エンジンなどで検索しなければならない．これに対し WIX では閲覧中の Web 文書に英和辞書の WIX ファイルを結合することで，英単語からその英単語のページへのハイパーリンクが一括生成される．

また，同じ英単語をキーワードとする WIX ファイルでも，例えば英和辞書の他にも英英辞書，英仏辞書，または同じ英和辞書でも会社が異なる辞書まで WIX ファイルを提供すれば，利用者は必要に応じて自由に WIX ファイルを選択し，自分の閲覧している文書にアタッチすることで状況に応じた情報を簡単に切り替えることができる．これにより Web ページ作成者の意思とは独立した，ユーザ主体でのハイパーリンクの生成を行うことができる．

2.4 WIX システム (クライアント)

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．図**??**，図 3，図 4 は FireFox add-on の例である．

WIX システムでは，ユーザは予め目的に適った WIX ファイルをブックマークしておく必要がある．図 3 のようにログインするとツールバーにアタッチボタンが生成される．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．アタッチボタンをクリックすることで，図 3 の記事にはなかったハイパーリンクが図 4 の記事には生成される．これによって，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　WIX ツールバー

図 3　Firefox add-on(1)

図 4　Firefox add-on(2)

3. WixMinus: アタッチ可能な WIX ファイルの紹介及び推薦機能

3.1 背　　景

ユーザは，図 9 内の WIX DB マネージャ(図 5) にアクセスし，WIX ファイルを検索する．そして，得られる検索結果 (図 6) から目的の WIX ファイルを選び，ブックマークに追加することになる．この時，WIX ファイルを登録するブックマークは，現状のシステムでは予め用意されているものから選ぶ事しかできず，新たにブックマークを作成したい際は，予め新規ブックマークを別ページで作成した上で WIX ファイルをその新規ブックマークに登録しなければならない．また，この検索は WIX ファイルの情報でしか検索出来ず，登録されているエントリから検索することが出来ない．そのため，あるキーワードが登録されている WIX ファイルを検索しようとしても非常に困難である．そして，多くの場合ユーザはある程度有用な WIX ファイルを探してブックマークに登録し終えると，それ以降はその WIX

ファイルのみを使用するケースが多くなると想定される．この時, 同じ情報に対して新たな WIX ファイルが提供されても, 一般ユーザはそれに気付くことができない．これは, WIX ファイル提供者にとっても, ユーザにとっても損である．そこで, 本研究ではユーザが使用している WIX ファイルに関連するにも関わらず使用されていない, もしくは新たに追加された可能性のある WIX ファイルをユーザがアタッチしたタイミングで検索し, 紹介・推薦するシステムを提案した．

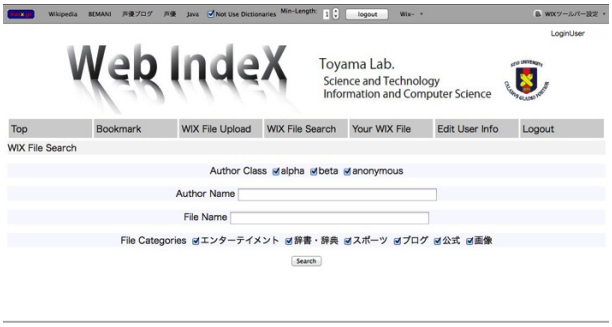

図 5　WIX DB マネージャ

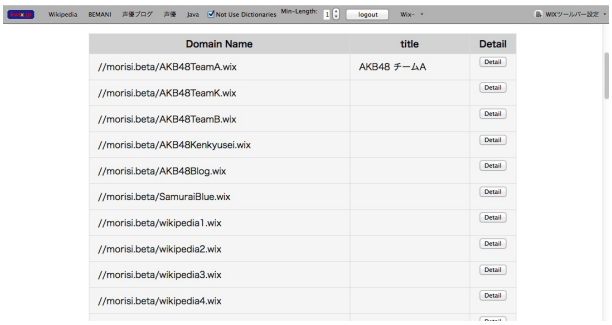

図 6　WIX ファイル検索結果

3.2　システム概要

3.2.1　アタッチ時

ユーザがアタッチすると, アタッチした Web ページに対する Find 結果の内, アタッチするブックマークに対応した WIX ファイル群に含まれるエントリが Select 結果となる．そこで, Find 結果から Select 結果を除いたものは, ユーザが使用していない, もしくは新たに追加された WIX ファイルのエントリである可能性が高い．これを UnSelect 結果と定義する．ここで, Find 結果, Select 結果, UnSelect 結果は (wid, eid, start, end) の 4 属性を持ったオブジェクトとして保持される．start, end はそれぞれ Web 文書でのキーワードの出現位置の始点及び終点を指す．この UnSelect 結果の wid を集計し, wid に対応したエントリ数を count 属性として (wid, count) というオブジェクトで取得する．そしてこの count 値の上位 10 件 (結果が 10 件に満たない場合はその件数) について, WIX ファイルの情報を取り出し, XML 形式 (図 7) でサーバに一時保存する．w_domain 属性はアップロードされた元 WIX ファイルのファイル名及びファイルアップロード者の情報で, ontology 属性は, 先の WIX DB マネージャで検索する際の File Category に当たる．

```
<wixdatas>
    <wixdata>
        <wid>19</wid>
        <title>LongmanEnglishDictionary.wix</title>
        <w_domain>//sitow.beta/LongmanEnglishDictionary.wix</w_domain>
        <count>11189</count>
        <ontology>2</ontology>
    </wixdata>
...
    <wixdata>
        <wid>156</wid>
        <title>JavaAPIクラス</title>
        <w_domain>//muta.beta/javaclass.wix</w_domain>
        <count>393</count>
        <ontology>2</ontology>
    </wixdata>
...
</wixdatas>
```

図 7　WIX ファイル情報形式

3.2.2　Wix-ボタン

ユーザはツールバー (図 2) の Wix-ボタンをクリックすることで, サーバーからこのファイルを読み込み, アタッチ可能な WIX ファイル候補の情報を一覧としてツールバーに表示する (図 8)．ここから, ユーザは目的の WIX ファイルに対して以下の 2 通りの操作ができる．

（1）　テンポラリアタッチボタン:選択した WIX ファイルを利用して一時的にアタッチが可能

（2）　新規ブックマークを作成, 追加

ユーザは (1) の操作でアタッチテストをしてみて, 気に入ったら (2) の操作で新たにブックマークを作成するといった事が可能になる．

図 8　Wix-ボタン

3.3　辞書系 WIX ファイルの扱い

システムの動作については前述の通りである．ここで問題になるのが, 一般の WIX ファイルに比べてエントリ数が圧倒的に多い辞書系 WIX ファイル (Wikipedia, LongmanEnglishLibrary etc...) の存在である．紹介する WIX ファイルの一覧が count 値でソートされていることは既に述べた．しかし, 辞書系 WIX ファイルに登録されているエントリ数が多い以上, UnSelect 結果に含まれる辞書系 WIX ファイルのエントリも格段に増えてしまう．すると count 値でソートされた結果辞書系 WIX ファイルばかりが一覧に表示されてしまい, 目的の WIX ファイルが下位の方に表示されてしまう, もしくは表示されすらしない危険性がある．そこで, WixMinus のオプションとして NotUseDic 属性を付与した．この値が true であると辞書系 WIX ファイルを除いた状態で WIX ファイル候補が一覧に表示される．

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．尚，提案手法においてはこの時絞り込みによって排除されたエントリ群を UnSelect 結果としている．

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. WIX アーキテクチャ

現在の WIX システムのアーキテクチャを図 9 に示す．

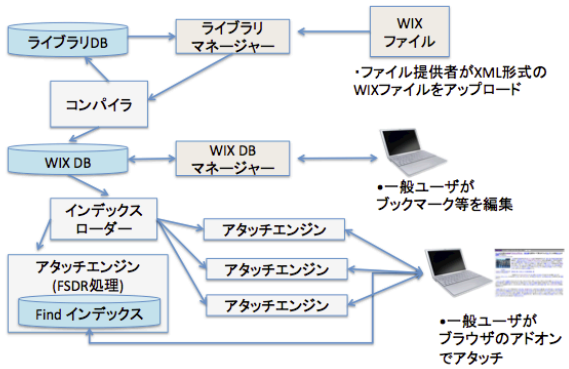

図 9 WIX アーキテクチャ

また，WIX システムでは WIX ファイルの情報をライブラリ，WIX DB，Find インデックスの 3 つの異なる形態で管理する．これら 3 形態の比較概念図を図 10 に示す．

5.1 WIX ライブラリ

ライブラリでは，全ての WIX ファイルを過去のヴァージョンも含めて XML テキストそのまま保存する (図 10 左)．即ち，ライブラリではファイル単位での情報管理となっている．しかしながらアタッチにおいては全ての WIX ファイルのエントリに対して，辞書式マッチングを行わなければならない．故に WIX ファイルをエントリ単位に分解し格納する WIX DB が必要となる．

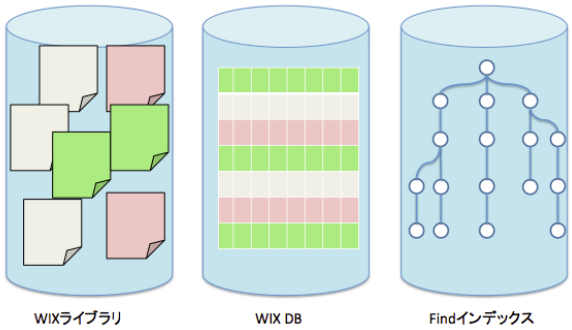

図 10 形態比較概念図

5.2 WIX DB

WIX DB においては，ライブラリで管理している WIX ファイルをエントリ単位に分解し，RDB にタプルとして管理する (図 10 中央)．即ち，WIX DB ではエントリ単位での情報管理となっている．ここで，WIX ファイルの情報を扱う entry テーブルについて述べる．

entry テーブルでは WIX ファイルのもつエントリ情報の管理を行う (表 1)．そのエントリが所属する WIX ファイルの wid，エントリの id である eid，辞書語となるキーワードの keyword とそれに対応する URL である target を属性として持つ．

表 1 entry テーブル

wid	eid	keyword	target
1	1	細貝萌	http://ja.wikipedia···
1	2	川島永嗣	http://ja.wikipedia···
1	3	アウクスブルク	http://ja.wikipedia···
2	1	イチロー	http://ja.51channel.tv···
3	1	細貝萌	http://samuraiblue.jp/···
3	2	川島永嗣	http://samuraiblue.jp/···
···			

5.3 Find インデックス

Find インデックスでは，WIX DB の entry テーブルからエントリ情報をメモリ上に展開する (図 10 右)．WIX システムでは Aho-Corasick 法に基づくオートマトンを構築し，辞書式マッチングを行う．

6. 実験・評価

本システムの有用性を評価するため，評価実験を行う．

6.1 評 価 方 法

本システムを利用した場合と利用しない場合の WIX ファイルの検索時間，ブックマークへの登録時間の差を測定した．実験は，WIX システムを利用したことのない被験者 10 名に参加

してもらい, 本システムを用いて任意のページに Wikipedia でアタッチ, Wix-ボタン操作を行なってもらい, 任意の WIX ファイルのブックマークへの登録を行なってもらった. そして, 本システムが使いやすいかについてアンケートを行った. また, 内 10 名の内 5 名には更に従来方式でのブックマークの追加も行なってもらい, 各作業にかかった時間の測定を行った.

6.2 結　　果

被験者 10 人に対するアンケート結果を図 11 に示す. どのユーザからも否定的な回答は出なかった. よって, 本手法の有用性が示せていると言える. しかし, 「どちらとも言えない」という回答をしたユーザに関して理由を聞いた所, 「既存のブックマークへの追加もできた方が良い」「ブックマーク名も任意で設定したい」といった意見が挙げられた. この改善が今後の課題と言える.

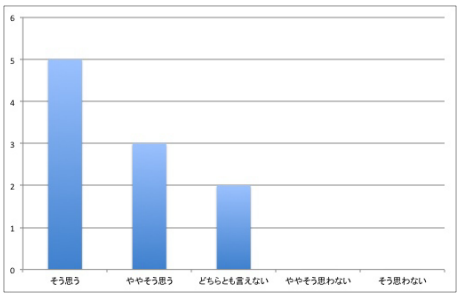

図 11　WIX-が使いやすいか

また, 被験者 5 人の提案手法, 従来手法のそれぞれの作業にかかった時間の一覧を表 2 に示す. どの被験者も提案手法によって作業の時間が大幅に短縮できている事が分かる.

表 2　実験 2 結果

被験者	提案手法 (s)	従来方式 (s)
A	87	312
B	73	196
C	103	327
D	64	259
E	71	275

7. まとめ・今後の課題

本研究では, アタッチ時にアタッチした WIX ファイルに含まれなかったエントリ群を用いる事で, ユーザが登録していない WIX ファイルを紹介, 推薦し, ユーザがその場でブックマークに登録できるシステムを提案し実装した. 辞書系 WIX ファイルによって必要な WIX ファイルが正しく表示されない危険性を考慮し, 辞書系 WIX ファイルを排除できるオプションを実装した. これによって, ユーザのブックマーク登録の手間が大幅に削減できたと言える.

とは言え, 本システムによる WIX ファイルリストへの表示の精度や, 本システムがサーバにかける通信量等の問題がまだ多くあるので, そういった要素を更に研究し, 本システムの精度向上に努めていく.

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] 市東　隼人, 分部　亮太, 朱　成敏, 遠山　元道. Keio WIX システム (3) コンテンツ作成. データ工学ワークショップ, DEIM2011. 2011.
[4] 藤井　洋太郎, 遠山　元道. WIX システムにおけるコンテンツ作成支援システム. データ工学ワークショップ, DEIM2012. 2012.
[5] 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.