

iOS プラットフォームにおける Web Index システムの提案

生田 史織[†] 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: [†]ikuta@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

あらまし Web Index とは、単語とそれに関する Web 文書の URL のペアの集合を閲覧中 Web 文書に対して結合することによって Web 文書中のキーワードに対して指定された URL へのハイパーリンクを生成し、Web 情報資源の結合を実現するシステムである。既存の WIX ではデスクトップブラウザにのみ対応した処理機構を実現していたが、WIX の利便性をさらに向上させるためには、モバイル端末において利用可能である必要がある。そこで、本研究では国内においてモバイルブラウザのシェアの 6 割以上を占める iOS プラットフォームの標準ウェブブラウザである Safari ブラウザに対応する新たなアタッチ処理機構およびそれらを用いたモバイルアプリケーションの開発を行う。

キーワード Web 情報システム, Web Index, iOS, アプリケーション

1. はじめに

近年、インターネットの普及により Web 上での情報検索のニーズが著しく増加し、ユーザは検索エンジンを使用することで必要な情報を得ることが可能となった。しかし、検索エンジンを用いた情報検索は、検索結果として得られた Web ページ内の単語の詳細情報を得たい場合には、新たに検索エンジンで再検索し、選択しなければならない。Web ページ作成者はハイパーリンクによって Web ページに関連する複数の情報を予め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込えているとは限らない。そこで、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 (WIX) システムという情報資源表現形式の提案、開発を行っている [1] [2]。これは、関係データベースの基礎となる関係モデルの考え方を Web に応用したシステムであり、新たな Web の形態の 1 つとして研究が行われている。

単語 (keyword) と URL (target) の組み合わせであるペア (WIX エントリ) の集合を XML 形式で記述したものを WIX ファイルという。WIX ファイルを用い、閲覧中の Web ページに結合 (アタッチ) することで、Web ページ内の文章に出現するキーワードをそれに対応する URL へのハイパーリンクに変換する。現在の Web 形態では、Web ページ作成者によって特定のアンカーテキストから特定のページへのリンクが関連付けられるという構造が一般的である。Web ページ内のリンクは常に既存の Web ページへしかリンクすることができず、その Web ページ作成後に作成される新たな Web ページへのリンクを作成することは、その Web ページ作成時には決してできない。WIX では、アンカーテキストとリンクを Web ページから独立した「キーワードとリンク先の集合」として扱い、任意のドキュメントに対してユーザ主導で「結合」することでドキュメント内のキーワードに対応する URL のハイパーリンクに自動的に変換する。その結果、Web ページ作成の時系列という壁を越え、古い Web ページからより新しい Web ページへのリン

クも可能となる。

現在、WIX システムはデスクトップブラウザに対応した処理機構を主軸に実現している。WIX システムの利便性をさらに向上させ、ユーザ数を拡大するためには、モバイル端末においても WIX システムが利用可能である必要がある。そこで、本研究ではモバイルウェブブラウザに対応する新たなアタッチ処理機構およびそれらを用いた WIX システムのモバイルアプリケーションの開発を行った。

本論文の構成は以下のとおりである。まず、2 章で WIX システムの概要・現状を説明し、次に 3, 4 章で本研究の設計と実装について述べる。最後に 5 章で評価について述べ、6 章でまとめを行う。

2. Web Index システム

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る。エントリには、キーワードとなる見出し語を keyword 要素として、それに対応する詳細情報を示す参照先の URL を target 要素として格納する。また 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データを格納することも可能である。記述例は図 1 のようになる。WIX ファイルは「日本語版 wikipedia の見出し語一覧 (wikipedia.ja.wix)」や「企業公式サイト (企業.wix)」や「Ameba ブログ (Ameba.wix)」などのように、内容がグルーピングされたものが存在する。

2.1 アーキテクチャ

WIX システムでは WIX ファイルの情報を WIX 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 2 に示し、それぞれについて説明していく。

```

<?xml version="1.0" encoding="utf-8"?>
<WIX>
  <header/>
  <body>
    ...
    <entry>
      <keyword>錦織圭</keyword>
      <target>http://ja.wikipedia.org/wiki/錦織圭</target>
    </entry>
    <entry>
      <keyword>ノバク・ジョコビッチ</keyword>
      <target>http://ja.wikipedia.org/wiki/ノバク・ジョコビッチ</target>
    </entry>
    <entry>
      <keyword>ロジャー・フェデラー</keyword>
      <target>http://ja.wikipedia.org/wiki/ロジャー・フェデラー</target>
    </entry>
    ...
  </body>
</WIX>

```

図 1 WIX ファイルの記述例

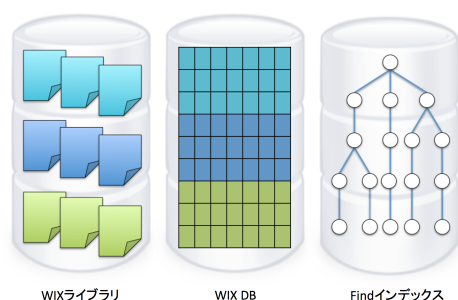


図 2 形態比較概念図

2.1.1 WIX ライブラリ

WIX ライブラリでは、全ての WIX ファイルの XML テキストをそのまま保存しており、ファイル単位での情報管理を行っている。アタッチの際には全ての WIX ファイルのエントリに対して辞書式マッチングを行うため、WIX ファイルをエントリ単位に分解し WIX DB に格納する。

2.1.2 WIX DB

WIX DB では、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 タプルとして管理する。WIX ファイルのもつエントリの情報は entry テーブルで管理される。(表 1)。エントリが属する WIX ファイルの id(wid)，エントリの id(eid)，辞書語となる keyword とそれに対応する target を属性として持つ。

表 1 entry テーブル

wid	eid	keyword	target
1	1	錦織圭	http://ja.wikipedia.org/wiki/錦織圭
1	2	フェデラー	http://ja.wikipedia.org/wiki/フェデラー
2	3	田中将大	http://ameblo.jp/tanaka-masahiro/
3	5	坂本勇人	http://www.giants.jp/G/player/prof.2756.html
:	:	:	:

2.1.3 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。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。

2.2 ハイパーリンクの生成 (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 などによって実装されている [1]。32 は Chrome Extension の例である。ユーザがブックマークボタンをクリックすると、サーバーサイドにおいて閲覧 Web ページと Find インデックスとの辞書式マッチングが行われ、リンク生成済の HTML 文書がレスポンスとして返され、元のページにはなかったハイパーリンクが処理後のページに生成される。このハイパーリンクを生成する処理をアタッチと呼ぶ。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。

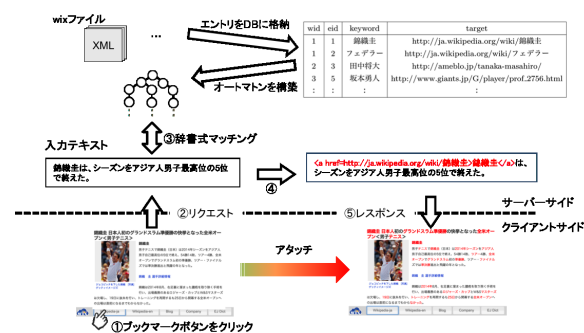


図 3 ハイパーリンクの生成 (Chrome Extension)

2.3 WIX システムの強み

一般的なウェブブラウジングでは、ユーザは閲覧している Web ページ上で気になる単語があった場合、その単語を新しく開いたタブの検索窓にコピーアンドペーストもしくは直接入力し検索することでその単語に関する情報を得ている。一方 WIX システムでは、使用する WIX ファイルをツールバーで選択しリンクが生成された単語をクリックすることでユーザが求める情報を得ることができる。このようにクリックのみでユーザが必要とする情報を取得可能であることは WIX の強みのひとつであるといえる。

2.4 WIX システムの現状

現在 WIX システムは Chrome, Safari, Firefox とデスクトップブラウザに対応した処理機構を主軸として提供している。しかしこの現状には2つの懸念点があると考えられる。1つ目は、デスクトップブラウザでは WIX の強みを活かしきれていないという点である。これは PC ブラウザにはマウスや物理キーボードが存在しており、検索したいワードのコピーアンドペースト・入力が比較的容易であるということである。2つ目は、

近年のウェブ技術の発展によりウェブ利用形態はデスクトップ環境からモバイル環境へ広がっているという点である。モバイル端末上でのウェブの利用形態は PC 上でのものと大きく異なっているため、デスクトップ環境のみへの対応ではユーザーのニーズに応えきれていないといえる。この懸念点2点は、WIX をシステムとして提供し、ユーザーの規模を拡大するのに足かせになるといえる。

2.4.1 WIX システムとモバイル端末

一般的にモバイル端末はデスクトップ PC に比べて画面が小さく物理キーボードが無いため文字が入力しづらい。そのため、デスクトップ PC でのウェブ検索と比べて、モバイル端末で行われるウェブ検索のクエリは短く・また曖昧なものが使用される傾向があるといえ、このことに注目したモバイル検索エンジンの研究などが実際に行われている [4]。このことから、モバイル機器の画面の小ささと文字の打ちづらはモバイル検索の課題となっているといえる。また、ウェブページ上に検索したい文字列があった場合、PC 上のマウス操作では容易であった文字列の選択がモバイル端末では指による操作ため困難であるといった問題点がある。そこで、モバイル端末において WIX システムが利用可能となれば、クリックのみでウェブページを遷移できるため、モバイル端末上でのウェブブラウジングに伴う様々なコストを削減でき、WIX のユーザー規模の拡大にもつながるといえる。

現在、androidOS で使用可能な専用ブラウザを用いた WIX 処理機構が実現している [5]。しかし、モバイルウェブブラウザは主に上位 3 つのシェアが全体の約 97 %を占めており、新しくその他の専用ブラウザを使用するユーザーは獲得できないと言える。そのため、シェアの大半を獲得するためには、iOS プラットフォームの標準ブラウザである Safari ブラウザに対応することは不可欠であるといえる。

3. 提案設計

iOS プラットフォームの標準ブラウザである Safari ブラウザに WIX システムを対応させる本研究の提案設計では、2014 年 9 月にリリースされた iOS 8 より追加された AppExtension [6] という機能を利用する。この機能により、iOS プラットフォームにおける WIX システム専用のブラウザを構築するのではなく、iOS の safari ブラウザをホストアプリケーションとし、WIX システムを拡張機能のアプリケーションとすることで、ユーザーが新たに使用するブラウザを変更することなく iOS プラットフォームにおいて WIX システムを利用することが可能となった。AppExtension とは、起動中のアプリケーションから別のアプリケーションを立ち上げる拡張機能である。実際の挙動のイメージは図 4 のようになっている。Safari が AppExtension を通して拡張機能である WIX アプリケーションをリクエストし、WIX アプリケーションがリソースを ReadWrite しレスポンスを AppExtension を通して Safari に返す。

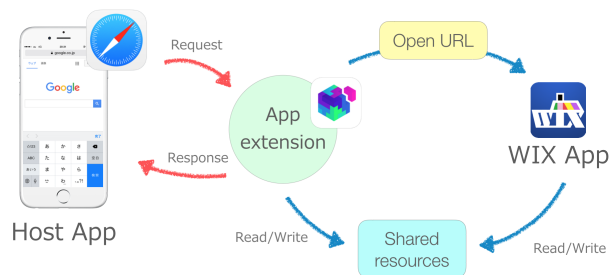


図 4 WIX アプリケーションのイメージ

ここで、さらに具体的に拡張機能としての WIX アプリケーションの挙動を決定すべく、AppExtension の機能について詳しく説明する。AppExtension では様々な拡張ポイントが提供されているが、本研究では Action というホストアプリケーションで表示されるコンテンツを操作または表示する拡張ポイントというものを使用する。実際に Action を使用しているアプリケーションとしては、パスワード管理アプリの 1Password [7] や iOS ブラウザ上でのソースコードを表示する View Source [8] などがあげられる。

実際に 1Password の挙動の流れを例示する。図 5 のように Twitter のログイン画面を Safari ブラウザで開いた場合、拡張機能アイコンから 1Password のアプリケーションを呼び出し、立ち上がった 1Password のアプリケーション内で twitter アカウントを選択をし、その結果が反映された Safari ブラウザが自動的に返されるといった流れになっている。これがホストアプリケーションで表示されるコンテンツを操作または表示するという Action 拡張ポイントの機能を使用した例である。

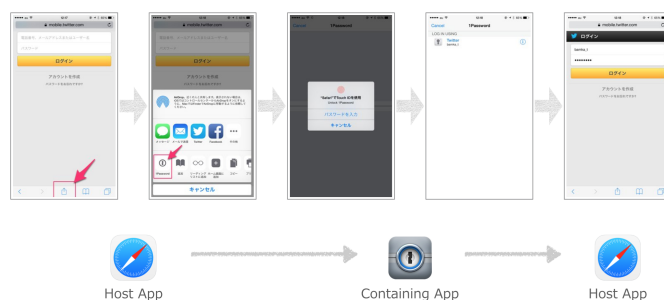


図 5 1Password の挙動例

4. 実装

続いて、拡張機能としての WIX アプリケーションの実際のクライアントサイドでの挙動を決定していく。ここでは、考えられる 3 つの実装例を提示する。本研究ではこれら 3 つを全て実装し、評価実験を行う。

4.1 実装例 1

1つ目の実装例は図6のようにになっている。これはPCブラウザ上提供されているWIXシステムの挙動に類似している。リンクを生成したいウェブ文書がある場合、拡張機能としてWIXアプリケーションを立ち上げ、そのアプリ内でリンク先を生成するWIXファイルを選択しリンクが生成されたWeb文書がSafari上で返される。



図6 実装例1

4.2 実装例 2

2つ目の実装例は図7のようにになっている。拡張機能としてWIXアプリケーションを呼び出した際、WIXアプリケーション内でWIXファイルを選択することなく自動的にアタッチされたSafariのウェブ文書が返される。これは1つ目の実装例で懸念されるWIXアプリケーションを立ち上げ、WIXファイルを選択するというユーザの動作コストを解消する実装例となる。これまでのWIXシステムのようにアタッチの際にリンク先を決定するのではなく、全てのリンク先が文書上のキーワードにアタッチされ、その中からユーザが必要なリンク先を選択する。



図7 実装例2

4.3 実装例 3

3つ目の実装例は図8のようにになっている。従来のWIXシステムとは異なった形の新しい実装例である。拡張機能として呼び出したWIXアプリケーション上で、ブラウザ上に表示されていたWeb文書のキーワードとリンク先のリストを表示さ

せ、そこからユーザが必要なリンクをクリックする。すなわち、上記の2つのものと違いWeb文書上にリンクを生成せず、リンク先のリストを表示するため、表示するリストの順序やその数をアルゴリズムによって決定することでユーザの利便性を向上できると考えられる。



図8 実装例3

5. 評価

クライアントサイドにおけるユーザビリティ、サーバーサイドにおける通信量などの観点から3つの実装例に対する評価実験を行う。また、更なるユーザビリティの向上を目指すべく、ユーザのクリック履歴やGPSによる位置情報を利用した機能の構築、それによって向上したユーザビリティを評価する。

6. まとめ

近年のウェブ技術の発展によりウェブ利用形態はデスクトップ環境からモバイル環境へ広がっている。モバイル端末上でのウェブの利用形態はPC上でのものと大きく異なり、デスクトップ環境のみへの対応ではユーザのニーズに応えきれていない。モバイル端末においてWIXシステムが利用可能となれば、クリックのみでウェブページを遷移できるため、モバイル端末上でのウェブブラウジングに伴う様々なコストを削減でき、WIXのユーザ規模の拡大にもつながる。そこで、本研究ではiOSプラットフォームの標準ウェブブラウザであるSafariブラウザに対応する新たなアタッチ処理機構を備えたモバイルアプリをAppExtension機能を利用し開発を行う。

文献

- [1] 林昌弘, 青山峻, 朱成敏, 遠山元道. KeioWIX 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] 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.

- [4] Leung, Kenneth Wai-Ting, Dik Lun Lee, and Wang-Chien Lee. PMSE: A personalized mobile search engine. Knowledge and Data Engineering, IEEE Transactions on 25.4 (2013): 820-834.
- [5] 藤井洋太郎, 遠山元道. Web Index クライアントサイドにおけるクロスプラットフォーム化. 慶應義塾大学大学院修士論文, 2013.
- [6] <https://developer.apple.com/jp/documentation/ExtensibilityPG.pdf>
- [7] <https://agilebits.com/onepassword>
- [8] <https://itunes.apple.com/jp/app/view-source-html-javascript/id917660039?mt=8&ign-mpt=uo%3D4>