

# 双方向変換に基づくウェブページ更新機構

## Web Page Updating based on Bidirectional Transformation

中野 圭介<sup>†</sup>, 森畑 明昌<sup>†</sup>, 胡 振江<sup>†</sup>, 武市 正人<sup>†</sup>

Keisuke NAKANO, Akimasa MORIHATA, Zhenjiang HU, Masato TAKEICHI

<sup>†</sup> 東京大学大学院情報理工学系研究科

Dept. of Mathematical Informatics, University of Tokyo

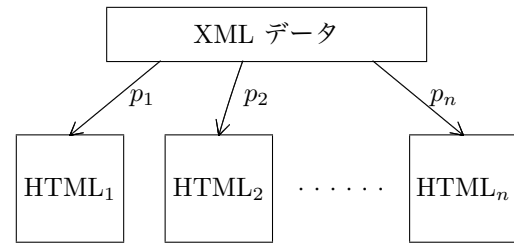
{ksk,morihata,hu,takeichi}@ipl.t.u-tokyo.ac.jp

複数のページの整合性を保ちつつウェブページを更新することは非常に困難である。これを解決する一つの方法として、共有の XML データに対する HTML 文書への変換を複数用意することが考えられる。しかしながら、この方法では XML データを直接変更する必要があるため、Wiki や blog のようにブラウザ上からウェブページを直接更新したいという要望に答えることはできていない。そこで、本稿では、HTML 文書側の編集に対応した XML データの更新の自動導出が可能な双方向変換言語 Bi-X を利用することにより、新しいウェブページ更新機構 Vu-X を提案する。本機構の実装では、Javascript を用いることによりウェブページをブラウザ上で違和感なく編集することが可能であり、その編集は自動的に XML データ側に反映される。

## 1 はじめに

インターネット上にウェブページを公開する際、情報発信者の信頼性を維持するためには、公開した情報の迅速な更新が望まれる。その一方で、情報発信者の管理しているページは一般に複数であり、それらに含まれる情報の一部は他のページと共有していることも多いため、これらの整合性を保ったまま更新することは非常に面倒な作業である。このような煩わしさを解決する方法として、一つの XML データに対し、それを HTML 文書へ変換するプログラムを複数用意する手法が考えられる (図 1)。これにより、XML データ側を更新するだけで複数のページに表示される情報をそれぞれ自動的に更新させることが可能になる。実際、XSLT [12] などに変換プログラムを作成して構築しているウェブサイトも多い。

このような構成の場合、発信している情報を更新するには XML データを直接編集しなければならないため、そのデータを保管するサーバにアクセスする必要がある。情報の更新頻度を維持するためには、任意のクライアントから標準のウェブブラウザなどを通じて更新できることが望ましい。サーバへアクセスするだけであれば、単に XML データを更新するための CGI を用意するだけでも解決できるが、更新したいウェブページの編集すべき部分が XML データ内のどの箇所に対応するかを見つけることは一般に難しい。また、情報を公開しているウェブページ



$p_1, p_2, \dots, p_n$  は HTML 文書への変換プログラム

図 1: 変換プログラムに基づくウェブサイトの構成

自体をブラウザから直接編集したいという要望も少なくない。このことは、近年 Wiki [10] や blog [5, 3] などが急激に台頭してきていることから窺い知ることができる。

ブラウザから利用できるウェブページ編集機構はこれまでもいくつか提案されているが、そのほとんどは図 1 の XML データに相当するようなデータベースを持たず、対象としているページの HTML 文書のみを作成・編集を目的としているため、上述した整合性の問題を解決することはできない。また、Wiki や blog などはデータベースを利用することで整合性を保つことを可能にしているが、データベースからウェブページとして表示するための変換プログラムは限定されており、サイト管理者の望むウェブサイトの構築は難しい。

本稿では、このような問題を解決するために双方向変換言語 Bi-X [13] を利用する。ここでいう双方向

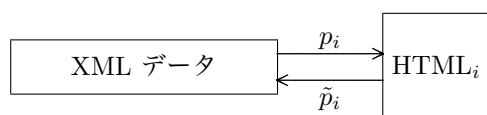


図 2: XML データとウェブページ間の双方向変換

変換とは、図 2 のように、XML データから HTML 文書への変換  $p_i$  だけでなく、その逆変換に相当する HTML 文書側の変更を XML データへ反映できるような変換  $\tilde{p}_i$  も存在することを意味している。言語 Bi-X は XML データから HTML 文書への順変換を記述するためのものであるが、Bi-X の基となる言語 X/Inv [2] の理論的枠組により上述の逆変換が自動的に導出できるという性質を持つ。そこで、我々はこのような双方向変換言語に基づく新しいウェブページ更新機構 Vu-X を提案する。Vu-X には次のような特長がある：

- データベースと変換プログラムによりウェブサイト構成することにより、ウェブページ同士の整合性を常に保つことができる。
- 変換プログラムに双方向変換言語 Bi-X を用いているため、XML データから HTML 文書へのレイアウトのための変換を記述するだけでよい。HTML 文書側の編集がどのように元のデータベースに反映されるかは自動的に導出される。
- 順変換および逆変換のための Bi-X の処理系などをサーバに担当させることにより、クライアント側に特別な処理系をインストールしなくてもウェブページの更新ができる。
- サーバと交信する Javascript を HTML 文書に付加したものを提供することにより、クライアント側ではブラウザ上で HTML 文書の編集ができるため、特別なアプリケーションがなくても WYSIWYG スタイルでのウェブページの更新ができる。

本稿は、この節を含め 5 つの節から構成される。まず、第 2 節において双方向変換言語 Bi-X について概略を述べる。次の第 3 節では、本研究の主体となるウェブページ更新機構 Vu-X について紹介し、その実装についても解説する。第 4 節で関連研究について触れ、第 5 節を本稿のまとめとする。

```

<xmap>
  <xif>
    <xreq><xmap><xhide/></xmap><c/></xreq>
    <xmap><dup/></xmap>
    <xid/>
  </xif>
</xmap>

```

図 3: Bi-X プログラムの例

## 2 双方向変換言語 Bi-X

本節では、我々の提案するウェブページ更新機構 Vu-X が利用している双方向変換言語 Bi-X についての概略を述べる。

### 2.1 双方向変換

あるデータベース (ソース) に対し、その中の必要な情報を抽出・整形した表示 (ビュー) を与えることを考える。ソースの集合を  $S$ 、ビューの集合を  $V$  で表すものとする。いま、ソースからビューへの変換  $FWD : S \rightarrow V$  だけでなく、ビューからソースへの変換  $BWD : V \rightarrow S$  が存在するとき、これらの変換をソースとビューの間の**双方向変換**と呼ぶ。一般に、ビューはソースよりも情報量が少ないため、ビューからソースへの変換  $BWD$  を一意に定めることはできない。しかしながら、多くの場合、更新後のビューと更新前のソースから更新後のソースを求めることは可能であるため、変換  $BWD : (V \times S) \rightarrow S$  を定義することで、この問題は解決できる。双方向変換言語 Bi-X では、このような広い意味での双方向変換を扱っている。

### 2.2 Bi-X による双方向変換

本枠組では、ソースを XML データ、ビューを XHTML 文書とするような双方向変換を考える。言語 Bi-X によるプログラムは、XML データから XML データへの変換を定めており、これは上述のソースからビューへの変換  $FWD$  に相当する。更新後のビューと更新前のソースから更新後のソースへの変換  $BWD$  は、Bi-X の基となる言語 X/Inv [2] の理論的枠組により自動導出される。

ここで、Bi-X による双方向変換の大きな特長である**入力複製**に対する双方向変換について簡単に紹介する。Bi-X では XML 変換プログラムを図 3 のように XML で記述する。本稿では Bi-X の詳しい文法については省略するが、このプログラムはルート要素

の子である *c* 要素の全ての子をそれぞれ複製するような変換を表している。複製は `<dup/>` で表現され、入力となる XML データの要素を 2 回繰り返した列を子として持つような Dup 要素を出力する。このプログラムでは、ルートの下の子の *c* 要素の全ての子がそれぞれ複製されるので、入力される XML データ

```
<root>
  <a> <b/> </a>
  <c> <d><f/></d> <e/> </c>
</root>
```

に対し、XML データ

```
<root>
  <a><b/></a>
  <c>
    <Dup> <d><f/></d> <d><f/></d> </Dup>
    <Dup> <e/> <e/> </Dup>
  </c>
</root>
```

を変換結果として出力する。`<dup/>` によって複製された要素は編集の際に同期されることが保証されている。たとえば、この変換結果において、複製された *d* 要素のうち、一方の子である *f* 要素を削除した場合を考える。このとき、Bi-X の処理系は、BWD によって得られる入力として、元の XML データの *d* 要素に対し同様の編集を行われたものを返す。更に、この入力に対して FWD による変換を再び適用するため、結果的に、編集されなかった *d* 要素に対しても同様の編集を行われたことになり、複製した要素が同期されることが可能となる。

このような複製は、XML データから (X)HTML 文書への変換において、タイトル (*title* 要素) と見出し (*h1* 要素) を同期したいときなどに使われる。

### 3 ウェブページ更新機構 Vu-X

我々の提案するウェブページ更新機構 Vu-X は、第 1 節で述べたような多くの特長を持っている。本節では、これらの特長を実現するためのウェブサイトの構成とそれを更新する機構について解説する。

#### 3.1 対象とするウェブサイトの構成

本枠組では、ページ同士の整合性を保ったウェブサイトを実現するために、図 1 に示したような XML データと変換プログラムに基づく構成をする必要がある。ここでは、変換プログラムとして Bi-X 言語を用いるものとする。どのような XML をデータベ

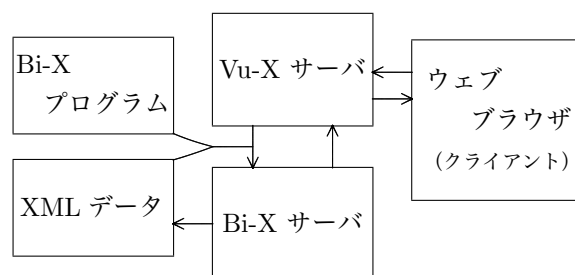


図 4: Vu-X の構成

スとして利用するかは設計者次第であるが、HTML 文書への変換プログラムに関しては Bi-X で記述する必要がある。Bi-X は少数の基本変換の組み合わせで定義するため、一般に大きなプログラムを書くのは困難であるが、XQuery (Core) [11] から Bi-X への自動変換が提供されており [1]、ユーザは XQuery によって変換を記述することもできる。また、ソースとなる XML データは BWD による更新ができるように WebDAV でアクセス可能なファイルサーバに存在するものとする。

#### 3.2 Vu-X の仕組み

我々の提案するウェブサイトの構成では、各ウェブページは XML データベースと Bi-X プログラムによって与えられている。このウェブページは、図 4 のように構成された機構に基づいて編集が行われる。この機構では、任意のクライアントのブラウザからウェブページを更新できるように、サーバ・クライアントモデルを採用している。このため、編集をしたいクライアントに特別な処理系をインストールする必要はない。ここでは、XML データや変換プログラムを保管する通常のファイルサーバの他に、双方向変換を提供する Bi-X サーバ [14] とウェブページの編集を補助する Vu-X サーバ の 2 種類のサーバを利用している。これらのサーバの連携により、通常のブラウザからのウェブページの更新が可能になる。以下では、これらのサーバの役割について解説する。

##### 3.2.1 Bi-X サーバ

このサーバには言語 Bi-X の処理系がインストールされており、Bi-X プログラムに対応した FWD および BWD による変換を提供する役割を担っている。FWD による変換では、クライアントから要求された Bi-X プログラムの URI とソースとなる XML デ

タの URI を受け取り, 変換結果である XHTML 文書  
を出力することが可能である. また, BWD による変  
換では, クライアントから要求された Bi-X プログラ  
ムの URI と変更されたビューである XHTML 文書  
と元のソースの XML の URI を受け取り, ビュー側  
の変更に従ってソースを WebDAV 経由で更新する.  
我々の提案する機構 Vu-X では, クライアントとし  
てウェブブラウザを想定しているため, サーバ・ク  
ライアント間のファイルや URI のやりとりは, HTTP  
の GET や POST を通じて行われる.

### 3.2.2 Vu-X サーバ

ソースの XML とそれに対するビューの XHTML  
文書への変換プログラムが Bi-X で与えられたとき,  
Bi-X の処理系による FWD の変換結果はウェブペ  
ージとして表示されるべき XHTML ビューであるが,  
このままではブラウザから編集できない. そこで,  
Vu-X サーバは, Bi-X サーバから FWD の変換結  
果である XHTML ビューを得た後, その XHTML  
ビューに編集機能を追加する Javascript プログラム  
の読み込むように変更して, それをクライアントに  
渡す. この Javascript により, クライアントが受け  
取った XHTML ビューはブラウザ上から直接編集す  
ることができる.

編集のための Javascript を読み込んだ XHTML  
ビューでは, body 要素にイベントハンドラが登録さ  
れており, あるノードをダブルクリックするとその  
ノードの部分が変更可能な form 要素に置換される  
(図 5). この form 要素は, 1 つのテキストボックス  
と「Modify」「Remove」「Insert」「Cancel」という 4  
つのボタンから成る. テキストボックスには, 初期  
値としてそのノードに対応する XHTML 文書の断片  
が表示されており, その断片を自由に編集するこ  
とができる.「Modify」はノードの編集,「Remove」は  
ノードの削除,「Insert」はノードの兄の追加を表し,  
それぞれのボタンをクリックすることにより, 更新  
後のビューがサーバに送信される. この送信された  
ビューは, 3.2.1 節で説明したように, Bi-X 処理系  
の BWD によるソースへの変更の反映に利用される.  
また,「Cancel」により, ノードの編集を中止して編  
集前の状態に戻ることができる.

Vu-X サーバは, BWD によって得られた新しい  
ソースに対し, Bi-X による複製などを解決するた  
めに再び FWD が適用される. 更に, FWD によって得  
られた XHTML ビューに対し, 上述の Javascript の



図 5: ブラウザ上でのウェブページの編集

付加を行いクライアントに返す. これによって, ウェ  
ブページの編集が引き続き行うことが可能になる.

### 3.3 サーバの実装

Bi-X サーバは [14] で提案・実装されているもの  
を利用する. ここでは Vu-X サーバの実装について  
解説する. このサーバは比較的小さなプログラムで  
あり, およそ 200 行の Ruby プログラムと編集操作  
を補助するためのおよそ 100 行の Javascript プログ  
ラムから成る.

Vu-X サーバの役割は主に以下の 2 点である. 一つ  
は Bi-X サーバを用いて編集可能な XHTML ビュー  
をユーザに提供することであり, もう一つはユーザ  
の編集操作に対応して Bi-X サーバと適切に通信し  
ビューを更新することである. 以下順を追って説明  
する.

#### 3.3.1 編集可能な XHTML ビューの生成

Vu-X サーバは編集可能な XHTML ビューを以下  
の手順で生成する. まず, ユーザからソースの XML  
データの URI および Bi-X プログラムの URI を受  
け取り, Bi-X サーバに FWD による変換を要求する  
ことによって XHTML ビューを得る. 次に, 得られ  
たビューに編集操作を行うための Javascript プログ  
ラムを組み込む. さらに, 編集操作の行われた箇所  
を特定するために, 各ノードへそれぞれ一意な識別  
子を割り振る. 最後に, できあがったビューを表示

する。この処理はサーバへの GET メッセージに対して行われる。

組み込まれた Javascript プログラムには次の 3 つの役割がある。

- ユーザのダブルクリックに対し、ダブルクリックされた要素を特定し、その箇所に編集のための form を表示する。
- form 内の「Modify」「Remove」「Insert」の各ボタンが押された際に、押されたボタンに対応して編集動作の情報を Vu-X サーバに送信する。具体的には押されたボタンの種類、テキストボックスの内容、そして編集された要素の識別子が POST によってサーバに送信される。
- 上記 POST メッセージに対する Vu-X サーバの返信 (後述) を用いてビューの更新を行う。

Javascript プログラムは Ajax の機構を用いており、以上の処理は動的かつ非同期的な HTML DOM の書き換えによって実現されている。また、ライブラリとして Prototype 1.4.0 [8] を用いた。

### 3.3.2 Bi-X サーバとの通信によるビューの更新

Vu-X サーバは、編集操作の確定によって送信される Javascript プログラムからの POST メッセージに対し、以下の順で処理を行う。まず、メッセージに含まれる情報を用いて古い XHTML ビューを編集する。次に Bi-X サーバに BWD および FWD による変換を要求することにより、新しいビューを得る。このとき、ソースの XML データは Bi-X サーバによって自動的に更新される。最後に、新しいビューを Javascript プログラムへ返信する。以上の処理の結果、ブラウザ上のビューは適切に更新される。

我々は、Vu-X サーバの動作を以下の環境で確認した。いずれの環境においてもプログラムはほぼ期待通りの動作を行い、ブラウザからウェブページを編集することができた。

- Windows XP (SP2) :
  - Internet Explorer 6.0
  - Firefox 1.5.0.5
  - Opera 9.00
- Mac OS X 10.4.7 :

- Safari 2.0.4
- Firefox 1.5.0.5
- Opera 9.00

## 4 関連研究

ウェブページを WYSIWYG で編集するツール数多く存在するが、そのほとんどは特別な処理系やアプリケーションのインストールを必要としており、必ずしも任意のクライアントから実行できるとは限らない。

このことを解決するために、ブラウザ上で HTML 文書を編集できるようにしたものいくつか提案されている。Active Square [6] はブラウザ上で WYSIWYG で編集可能なシステムを提供しているが、実装に ActiveX を利用しているため任意のブラウザでは実行できない。また、net2ftp [7] ではブラウザ上からファイルの編集ができるためウェブページの更新も可能であるが、HTML 文書の直接の編集を行う必要があり、編集したい箇所を見つけるのに無駄な時間がかかってしまう。一方、Javascript を用いた WYSIWYG の HTML エディタとして Tinymce [9]、FCKeditor [4] などが挙げられるが、いずれも単独のページの編集を目的とするため、整合性のあるウェブページの更新には困難を伴う。

Wiki [10] や blog [5, 3] などは、WYSIWYG ではないものの、データベースを利用しているため整合性を保つことが可能で、なおかつブラウザ上からの簡単な更新を可能にしている。しかしながら、我々の提案した Vu-X とは異なり、決められたレイアウトやテーマに沿った表示に限られているため、データベースからの HTML 文書へ変換するプログラムに対する自由度は低い。このため、既存の Wiki や blog を利用してウェブサイトの管理者が望むように設計するのは難しい。

## 5 おわりに

本稿では、整合性のあるウェブページの更新がブラウザ上から可能であるような新しいウェブページ更新機構 Vu-X を提案した。一つのデータベースを共有した複数の変換プログラムを用意することでウェブページ同士の整合性を保ち、各ページをブラウザ上から更新できるようにするために双方向変換言語 Bi-X を利用した。本実装では、Javascript を利用することでブラウザ上から HTML 文書を編集する機

構を実現した。

今回の実装では、双方向変換言語として Bi-X を利用したが、一般のウェブサイトの設計者には Bi-X を直接記述することは難しいと思われる。現行の Bi-X には XQuery (Core) [11] から自動導出する機構 [1] も提供されており、これを利用することも可能であるが、XML スキーマの情報が必要とするため、希望する Bi-X プログラムを容易に記述する環境はまだ整っているとは言い難い。双方向変換プログラムの記述しやすい環境の追究は今後の課題の一つであり、その成果によってより使いやすいウェブページ更新機構が提供できることが期待される。

## 謝辞

本研究は文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託を受けた東京大学において実施したものである。

## 参考文献

- [1] Dongxi Liu, Zhenjiang Hu, and Masato Takeichi. Bidirectionalizing XQuery – updating XML through materialized XQuery view. Technical Report METR 2006-25, Department of Mathematical Informatics, University of Tokyo, 2006.
- [2] Shin-Cheng Mu, Zhenjiang Hu, and Masato Takeichi. An algebraic approach to bi-directional updating. In *The 2nd ASIAN Symposium on Programming Languages and Systems*, Vol. 3302 of *Lecture Notes in Computer Science*, pp. 2–20, 2004.
- [3] bBlog: PHP blogging software. <http://www.bblog.com/>.
- [4] FCKeditor: The text editor for internet. <http://www.fckeditor.net/>.
- [5] MovableType: publishing platform. <http://www.sixapart.jp/movabletype/>.
- [6] Namo Active Square. <http://www.namo.com/jp/activesquare/>.
- [7] net2ftp: a web based FTP client. <http://www.net2ftp.com/>.
- [8] Prototype: JavaScript framework. <http://prototype.conio.net/>.
- [9] TinyMce: Javascript wysiwyg editor. <http://tinymce.moxiecode.com/>.
- [10] WikiWikiWeb. <http://c2.com/cgi/wiki>.
- [11] XQuery 1.0 formal semantics. <http://www.w3.org/TR/xquery-semantics/>.
- [12] XSL transformations (XSLT). <http://www.w3c.org/TR/xslt/>.
- [13] 劉東喜, 笈一彦, 胡振江, 武市正人, 王浩. A Java library for bidirectional XML transformation. 日本ソフトウェア科学会第 22 回大会論文集, 2005.
- [14] 林康史, 劉東喜, 江本健斗, 松田一孝, 胡振江, 武市正人. A web service architecture for bidirectional updating. 日本ソフトウェア科学会第 23 回大会論文集, 2006.