XML 操作スクリプト言語による個人化手法の提案

岡本辰夫 *1　吉田奈美子 *2　國島丈生 *2　横田一正 *2

*1 岡山県立大学大学院情報系工学研究科
tatu@c.oka-pu.ac.jp
〒 719-1197 総社市窪木 111

*2 岡山県立大学 情報工学部
{yoshida,kunishi,yokota}@c.oka-pu.ac.jp
〒 719-1197 総社市窪木 111

近年, ネットワークの発達に伴い様々なマルチメディアコンテンツを統合して提供することを目的とした研究が行われてきている. また, このような電子化された情報の交換フォーマットとしてデファクトスタンダードとなっている XML を用いたコンテンツの統合はさかんに行われてきている. しかし, 従来の研究においては決められた手順で情報を受け取ることがほとんどで, その情報に対して利用者の意図を反映させることは一般的に行われていない. そこで本研究においては, コンテンツの提供者の意図と利用者の意図の差分を, われわれの考案した XML 操作スクリプト言語 xTrics を用いて記述することによりコンテンツに利用者の意図を反映させる. また第三者の xTrics を重ねてかけることで複数の人の意図を適用したコンテンツを動的に生成することもできる.

キーワード: XML, スクリプト言語, 個人化, オントロジー

Personalization of XML-Based Contents as a Transformation Process

Tatsuo Okamoto*1 Namiko Yoshida*2 Takeo Kunishima*2
Kazumasa Yokota*2

*1 Okayama Prefectural University
Graduate School of System Engineering
tatu@c.oka-pu.ac.jp
111 Kuboki, Soja, Okayama 719-1197

*2 Okayama Prefectural University
Faculty of Information Science
and System Engineering
{yoshida,kunishi,yokota}@c.oka-pu.ac.jp
111 Kuboki, Soja, Okayama 719-1197

XML has become *de facto* standard as a protocol of information exchange for not only documents but also multimedia contents. However It is difficult to personalize the same shared digital information, because digital contents strongly reflect authors' intension and it can be personalized in the framework of neither database view nor filtering techniques. In this paper, we propose a script language, called xTrics, in which we can describe a personalization or transformation process of XML-based contents. Furthermore, xTrics can be applied to ontology-based transformation and dynamic generation of user-adaptable contents.

Key words: XML, script, personalization, ontology

1　はじめに

近年, コンピュータやネットワークの発達に伴い, 様々なマルチメディアコンテンツの作成・加工が容易になり, マルチメディアコンテンツを統合して提供することを目的とした研究が行われてきている. 特に, このような電子化された情報の交換フォーマットとしてデファクトスタンダードとなっている XML を用いたコンテンツの統合はさかんに行われてきている. そして, このように統合されたコンテンツに対しては, 図書館や美術館, 博物館などのようにデジタル化された三次元空間を使用して情報提供を行うことを目指した研究や, 本や web ページといった文章にして提供する研究がさかんに行われてきている.

しかし, 従来の研究においては決められた手順で情報を受け取ることがほとんどであるため, 利用者はそれらの膨大な情報の中から自分にとって必要な情報だけを選択して視聴する要求が増加している. さらに, コンテンツの提供者も利用者に個別のコンテンツ構成やリンク情報を提供したいという要求が高まっている. 例として, 利用頻度の高い利用者に対しては他の利用者と異なった情報を提供することで差別化を図ることや, 利用者の年齢に応じて年齢の低い利用者にはより理解しやすい内容を年齢の高い利用者にはより高度な内容を提供するという要求が考えられる.

このような背景から近年では, ネットワーク上に存在する様々なデータに対して個々の利用者や利用者グループに適した表現にする「個人化」に関する研究や個々の利用者の端末に合わせて提示レイアウトを変更する「適応化」に関する研究がさかんに行われてきている [1, 2, 3].

従来の個人化・適応化に関する研究においては, 情報の提供者側で利用者の個人情報や嗜好情報を収集し, さらに行動履歴などをフィードバックして利用者個人個人に合わせたフィルタを生成し, 情報をフィルタリングして提供することがほとんどである (図 1 に参照されたい). そのため, このような個人化のアプローチでは, 利用者の受け取った情報に対してさらに利用者の意図を反映させることは行われていない. また, 提供者側が個人化を行うことを明確にする必要があり, 個人化のシステムを組み込まなければならない. そのため, 個人化に対応してないシステムや web サイトに対しては利用者の意図を反映させて閲覧することはできない.

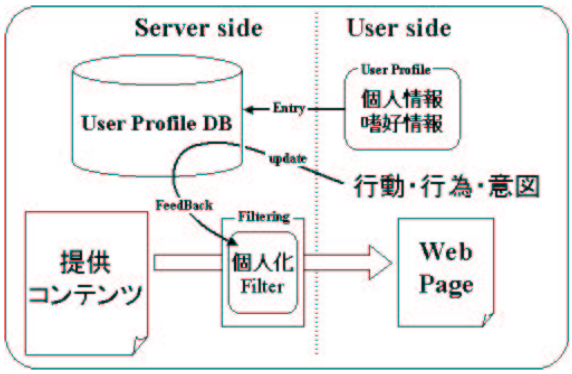

図 1: 従来手法による個人化

そこで本研究においては, 提供者側の提供したい情報や提供者の意図と利用者側の閲覧したい情報や利用者の意図との差分を取り, 差分情報をフィルタとして見ることで, ネットワーク上の情報を利用者個人個人でカスタマイズできる個人化手法を提案している. また第三者の差分情報を重ねてフィルタリングすることで, 複数の人の意図を適用した情報を動的に生成することもできる. 差分情報の記述としては, われわれの考案した XML 操作スクリプト言語 xTrics(XML TRee Information Control Script) を用いている.

本稿の構成は次のとおりである. 2 節で本研究において提案する個人化手法について述べ, 3 節で XML 操作スクリプト言語 xTrics について述べる. そして,4 節で xTrics の XML 表現について述べる. 最後に 5 節では研究のまとめを述べる.

2　個人化のアプローチ

前述したように本研究において提案する個人化手法とは, 提供者から提供された情報と利用者の意図との差分情報をスクリプト言語を用いて記述し, 提供された情報にスクリプトをかけることによって, 利用者の意図を反映させた情報にカスタマイズすることである. 利用者の意図としては, 以下の情報が挙げられる.

1) 利用者の意見などのメモ書き
2) 重要と思われる情報に対するマーキング
3) 情報の追加, 削除
4) 提示レイアウトの変更
5) 個人情報や嗜好情報による情報の選別

ここで 1,2,3) は利用者の主観的な意見などを適応するための情報であり, 4,5) は従来の個人化・適応化手法によくみられる情報であるが, 従来の手法とは違いこれらは利用者が情報を受け取ってから, 利用者側でフィルタリングするための情報である.

また本研究で考えているスクリプト言語による個人化手法は, 従来の研究によくみられる情報を提供者側でフィルタリングして提供する手法も, 提供者側の情報にスクリプトをかけて送ることで対応することが可能である. このとき利用者側に送られるのは元の XML 文書と実際にかけられたスクリプト列である. 図 2 に提案手法によるデータの流れを示す.

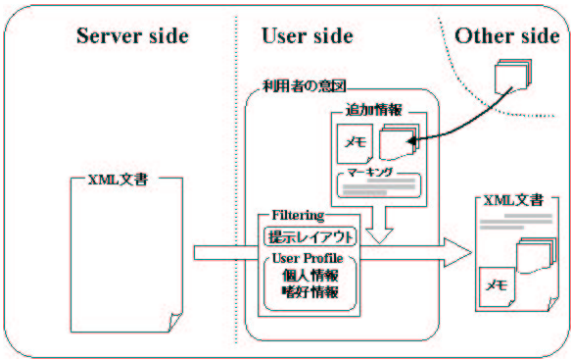

図 2: 提案手法の例

また, コミュニティや利用者グループに対して, 利用者個人個人の意図を共有化しておき, 他の利用者のスクリプトを重ねてかけることで複数の人の意図を適用したコンテンツを動的に生成することもできる. ここで共有化可能な利用者の意図は, 1,2,3) のみであり, 4,5) を共有化することはできない. このとき, どの利用者のどのスクリプトをかけるかは, 利用者 (閲覧者) が自由に選択し, 他の利用者が勝手にスクリプトをかけることはできない. 図 3 は図 2 にコミュニティを追加した提案手法の例である.

本研究において使用するスクリプト言語は, 本研究室で開発中の XML の構造を容易に操作できる XML 操作スクリプト言語 xTrics を使用している. 次節で xTrics の概要を述べる.

3 xTrics の概要

現在 XML は, 半構造化データの記述言語として広く利用されており, 電子化された情報の交換フォーマットとしてデファクトスタンダードとなっている.

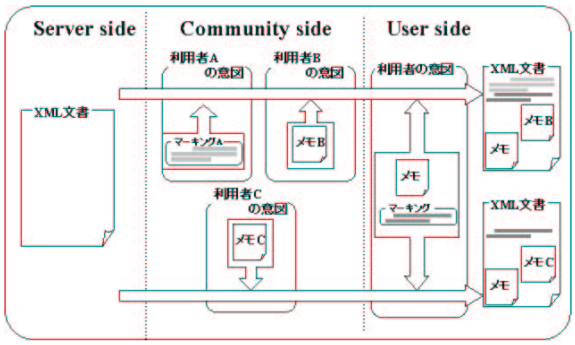

図 3: 提案手法の例 (Community)

xTrics は,XML が持っている木構造の各ノードに対して, 追加や削除, 変更などの操作を行うことを目的として本研究室で開発しているスクリプト言語である. その主な特徴として以下のことなどが挙げられる.

- XML の文章としての連続性の保証
- 木構造に対する操作
- 分岐や繰返しなどの制御フロー
- 文字列に対する強力な操作

3.1 文章としての連続性

前述したように xTrics は木の操作を行うことを目的として開発されている. しかし,XML はもともと構造化文書の記述言語として作られているため, SVG や SMIL などのように半構造化データの記述言語としての利用が増加したとはいえ,「文書」としての特質を持つ XML もたくさんある. そのため「文書」であるような XML に対して操作を行う場合, 文章としての連続性を損なわないことを保証してやる必要がある.

例えば図 4 ような XML 文章において, XML を木構造としてみた場合, section タグに 2 つの para タグがあり, para タグは子供として文章を持っている. また,marker タグは para タグの子供であることがわかる.

一方この XML を文章としてみた場合, 1 つの小節の中に段落分けされた文章が入っていてその一部分にマーキングしてあることがわかる. ここで, この XML から任意の一部分を切り出す操作を考える (後述するスライス操作の適用).

例えば,“築城などの記載はまったくな

```
<section>
  <para>
    鬼ノ城は,663 年の白村江の戦いで唐・新
    羅の連合軍に大敗し, その連合軍が日本に
    侵攻する危機感から, 国土防衛のため, 北
    九州〜瀬戸内沿岸〜畿内にいたる西日本
    各地に築いた, 古代山城の一つと考えられ
    ています. しかし,<marker color="red">
    『日本書紀』などに築城などの記載はま
    ったくなく</marker>, これまで謎の城と
    いわれてきました.
  </para>
  <para>
    国指定史跡である鬼ノ城を将来的には整
    備をしていくため, いまだ不明の点が多い
    ことからまず基礎データを得るための発
    掘調査を実施し, その成果をもとに整備計
    画を進めていく方針で, これまでに 3 回に
    わたる発掘調査を実施してきました.
  </para>
</section>
```

図 4: XML の例

く</marker>, これまで謎の城といわれてきました.” を切り出した場合,

1) <marker color="red">築城などの記載はまったくなく</marker>
2) , これまで謎の城といわれてきました.

の 2 つの木構造が抽出される. このとき,1) と 2) は順序が決定されており, 意識的に順序を入れ替えない限り 2),1) の順に並ぶことはない. そのため切り出した林から文章のみを取り出すと, “築城などの記載はまったくなく, これまで謎の城といわれてきました.” という文章が抽出できる.

このように xTrics においては, XML を木構造として捕らえながらも文章の連続性を保つために各ノードがシーケンシャルに一列に並べられるように考えている.

3.2 XML のデータモデル

XML を操作するスクリプト言語を考える場合, XML のデータモデルをどのように設計するかが非常に重要である. 一般的に XML のデータモデルは, タグの入れ子構造を木として捉える. xTrics においても同様であるが, 前節で述べたように XML の文章としての連続性を考慮する必要がある.

XML のひとつのタグに注目すると, その子供には必ずタグか文字列が来て, それらの要素間に明確な順序があるためにシーケンスで表現できる. また, 文字列も 1 文字のデータが順序よく並んだシーケンスであるため, XML の構造はシーケンスの入れ子となる. xTrics においてはシーケンスを [a,b,c] のように角括弧とカンマで区切られた要素により記述する. つまり図 4 は, [["鬼ノ城…かし,",["『日本…なく"],", これ…した."],["国指…した."]] となる. このとき"鬼ノ城…かし,"は文字列であり, [‘鬼’,‘ノ’,‘城’,…,‘か’,‘し’,‘,’] と同値である. しかし, このように XML を記述するとタグ名や属性の情報が欠落している. そこで XML を, Element@Att1="val1"…@Att$n$="val$n$" [value1,…,value$m$] のようにタグ名のあとに@で属性を列挙し, 角括弧とカンマで区切られた要素で記述する. XML において [value1,…,value$m$] の部分を省略した場合は空要素タグとして扱う.

そのため,xTrics における XML のデータモデルはラベルつき木構造であると考えることができる. また, 文章としてみた場合そのデータは行きがけ順にシーケンシャルに並んでいると考える (図 5 に参照されたい).

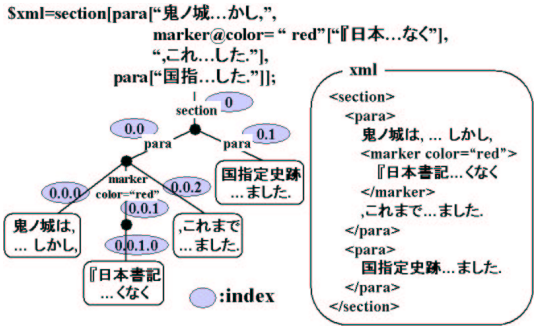

図 5: XML のデータモデル

3.3 インデクシングとスライス

XML データモデルの基本となるシーケンスは, 順序を持つデータ列であり各要素を指し示す index を持っている. (図 6 に参照されたい.) シーケンスの最初の要素の index は 0 である. また,index は負の数でもよく, このときは右から数える. つまり最後の要素が −1 である.

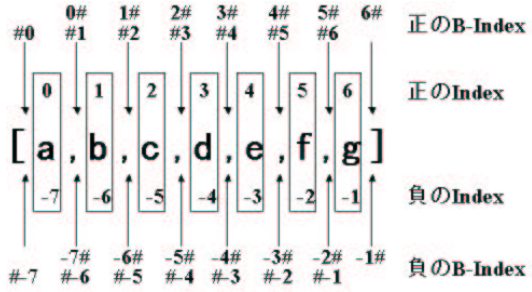

図 6: index の例

ここで, シーケンス型の変数に対して index を指定して, 指定の要素を抽出する演算をインデクシングと呼び, 変数名 [index] で記述する. xTrics では変数を$value のように$のあとに任意の英数字の列で記述する.

表 1 はインデクシングの例である. ここで 4) は,index の範囲を超えるためエラーとなる. また 5) は,$seq[-2] の結果がシーケンスであるため, さらにインデクシングを行える. しかし 6) は,$seq[2] の結果がシーケンスではないためエラーとなっている.

$seq=[1,‘x’,2.4,“xTrics”,‘a’];	
1) $seq[2];	2.4
2) $seq[-2];	“xTrics”
3) $seq[0];	1
4) $seq[-10];	error
5) $seq[-2][-5];	‘T’
6) $seq[2][4];	error

表 1: インデクシングの例

また, シーケンス型の変数に対して範囲を指定することで, 部分シーケンスを抽出する演算をスライスと呼び, 変数名 [index1:index2] で記述する.

ここで index1 と index2 の関係は index1 ≤index2 となる必要があり, この時の戻り値は指定した index を含むシーケンスとなる. もし index1 >index2 のときはエラーとなる. ただし負の index の場合は正の index に変換して比較する.

指定する index には, 要素と要素のあいだを指し示す B-index(Between index) を使用することもできる.

表 2 はスライスの例である. ここで 2) は 8,>5 であるためエラーとなる. また 4) は,index の範囲を超えているためエラーとなる.

$str=”xTrics はエックストリックスと読む”;	
1) $str[0:5];	“xTrics”
2) $str[8:5];	error
3) $str[6#:-4];	“エックストリックス”
4) $str[20:25];	error
5) $str[6#:#7];	[ ]
6) $str[6:#7];	“は”

表 2: スライスの例

次に XML のインデクシングとスライスについて述べる. XML は, 木構造であるので, 親 index. 子 index. 孫 index.· · · のように ‘.’ で連結された path で index を記述する. また,XML はラベルつきシーケンスでもあるため, index の変わりにラベルを指定しても良い. このとき指定するラベルは, タグ名, 属性名, 属性名+属性値の任意の組み合わせでよく, ラベルとマッチする要素が選ばれる. もしマッチするラベルが複数あるときは, ラベル (index) で指定する. ここで, 番号を省略すると最初にマッチした要素が選ばれる. 例として, 図 7 において,$XML[0.1] と $XML[label0.label1], $XML[0.label1] はすべて同値となる. また,XML はシーケンスの入れ子構造でもあるため, $XML[0.1.1] は$XML[0][1][1] と同値となる.

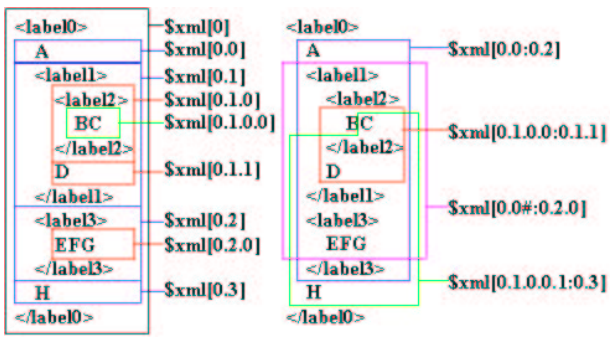

図 7: XML 型のインデクシングとスライス

XML のスライスにおいては, 抽出された値が林となる場合がある. しかし xTrics における XML には明確な順序があるため, 抽出された値は XML を要素とするシーケンスになる. またスライスで抽出された XML は, 一見するとウェルフォームドになってないように見えるが, xTrics において XML は木構造で処理するため, 抽出された XML は必ずウェルフォームドとなる.

3.4 XML の操作

xTrics における XML の操作としては, 前述したインデクシングとスライスのほかに XML の木構造に対する挿入, 削除の操作と各ノードの持つタグ名の変更や属性値の追加, 変更, 削除がある.

3.4.1 挿入操作

挿入操作は組み込み関数 insert() を使用して行う. insert() は引数を 3 つ持ち, 第 1 引数として挿入操作を行う対象となる XML を指定する. 第 2 引数としては B-index をとり, その場所にデータを挿入する. 第 3 引数として挿入するデータを指定する.

以下に図 5 の XML データに対して, 挿入操作を行う例を示す.

1) $xml=insert($xml,#0.0,title[“鬼ノ城”]);

```
<section>
 <title>鬼ノ城</title>
 <para>
  鬼ノ城は,663 年の白村…
  ……
  …実施してきました.
 </para>
</section>
```

また挿入操作は,XML だけでなくシーケンスに対しても挿入することができる.

3.4.2 削除操作

削除操作は組み込み関数 delete() を使用して行う. delete() は引数を 2 つ持ち, 第 1 引数として削除操作を行う対象となる XML を指定する. 第 2 引数としては index か範囲指定をとり, その場所のデータを削除する. 範囲指定は 3.3 節のスライスと同様に start:end で表記する. ここで start と end は index か B-index をとる.

以下に図 5 の XML データに対して, 削除操作を行う例を示す.

1) $xml=delete($xml,0.0.2.-5:#0.1.0.-5);

```
<section>
 <para>
  鬼ノ城は,663 年の白村…
  ……
  …といわれて
 </para>
 <para>
  きました.
 </para>
</section>
```

削除も挿入と同様にシーケンスに対して適用可能である.

3.4.3 タグ名や属性値の操作

XML の各ノードにおいてタグ名は必ず 1 つである. そのため, タグ名に関する操作は変更のみである. タグ名の変更は組み込み関数 ChangeElement() を使用する. ChangeElement() は引数を 3 つ持ち, 第 1 引数として操作を行う対象となる XML を指定する. 第 2 引数としては目標のノードを指定する index をとる. 第 3 引数として変更するタグ名を文字列で指定する. 以下に図 5 の XML データに対して, タグ名の操作を行う例を示す.

1) $xml=ChengeElement($xml,0,“鬼ノ城”);

```
<鬼ノ城>
 <para>
  鬼ノ城は,663 年の白村…
  ……
  …実施してきました.
 </para>
</鬼ノ城>
```

また, 各ノードにおける属性は複数存在するので, 属性に関する操作は追加, 変更, 削除がある. 属性の追加, 変更は組み込み関数 ChangeAtt() を使用し, 属性の削除は組み込み関数 DeleteAtt() を使用する.

ChangeAtt() は引数を 4 つ持ち, 第 1 引数と第 2 引数は ChangeElement() と同じデータをとる. 第 3 引数として属性名を文字列で指定する. もし対象のノードに指定した属性名が存在しなければ, 新しい属性を追加し, 存在すれば属性値を変更する. 第 4 引数としては, 属性値を文字列で指定する. 以下に属性の追加・更新操作を行う例を示す.

2) $xml=ChengeAtt($xml,0,“type”,“史跡”);

```
<鬼ノ城 type="史跡">
 <para>
  鬼ノ城は,663 年の白村…
  ……
  …実施してきました.
 </para>
</鬼ノ城>
```

3) $xml=ChengeAtt($xml,0.0.1,“color”,“blue”);

```
<鬼ノ城 type="史跡">
 <para>
  鬼ノ城は,663 年の白村…
  …られています. しかし,
  <marker color="blue">
   『日本書…たくなく
  </marker>
  , これまで…てきました.
 </para>
 <para>
  国指定史跡である…
  …実施してきました.
 </para>
</鬼ノ城>
```

DeleteAtt() は引数を 3 つ持ち, 第 1 引数, 第 2 引数, 第 3 引数は ChangeAtt() と同じデータをとる. ここで第 3 引数は削除する属性名である. 以下に属性の削除操作を行う例を示す.

4) $xml=DeleteAtt($xml,0,“type”);

```
<鬼ノ城>
 <para>
  鬼ノ城は,663 年の白村…
  ……
  …実施してきました.
 </para>
</鬼ノ城>
```

3.5 制御フロー

xTrics は, ほかの言語で知られている普通の制御フロー文を備えている. この節では, その中でもっとも一般的な分岐と繰返しについて述べる.

3.5.1 分岐

xTrics における分岐の制御フローは if 文のみを備えている. if 文は以下のような構文で定義されている.

$$
\begin{aligned}
\textit{if}\ \text{文} \quad &:= \quad \textit{if}\ \text{条件式}\ (\textit{then} \mid :)\ \text{式} \\
&\qquad [\textit{elif}\ \text{条件式}\ (\textit{then} \mid :)\ \text{式}] * \\
&\qquad [\textit{else}\ \text{式}] \\
\text{式} \quad &:= \quad \text{文}; \mid \textit{begin}\ (\text{式}) *\ \textit{end}; \mid \{\ (\text{式}) *\ \}
\end{aligned}
$$

このように if 文には 0 個以上の elif 部があってもよく,else 部を付けてもよい. if···elif···elif··· 列は, ほかの言語で見られる switch 文や case 文の代用品として使える.

3.5.2 繰返し

xTrics における繰返しの制御フローには while 文と for 文を備えている. また, 繰返し内のみ有効な制御フローとして break 文,next 文,redo 文がある. これらの制御フローは以下のような構文で定義されている.

$$
\begin{aligned}
\textit{while}\ \text{文} \quad &:= \quad \textit{while}\ \text{条件式}\ (\textit{do} \mid :)\ \text{式} \\
\textit{for}\ \text{文} \quad &:= \quad \textit{for}\ \text{変数}\ \textit{in}\ \text{シーケンス}\ (\textit{do} \mid :)\ \text{式} \\
\textit{break}\ \text{文} \quad &:= \quad \textit{break}; \\
\textit{next}\ \text{文} \quad &:= \quad \textit{next}; \\
\textit{redo}\ \text{文} \quad &:= \quad \textit{redo};
\end{aligned}
$$

ここで,while 文は条件式が真のあいだ式を実行していき, for 文はシーケンス内の各要素に対して, それらがシーケンスに現れる順序で繰返しをする. break 文はもっとも内側のループを脱出する. next 文はもっとも内側のループの次の繰り返しにジャンプする. redo 文はループ条件のチェックを行なわずに, 現在の繰り返しをやり直す.

3.6 操作の論理単位

xTrics におけるすべての文は, 処理後に成否が判定される. ここで成功していれば, その文で行った処理の結果をデータに反映し, もし失敗していれば, その文で変更されたデータはすべて破棄されて, 文実行前のデータに戻される.

また xTrics では成否にかかわらずプログラム終了まですべての文を処理する. つまり, 途中で失敗した文が現れてもその文を無効にするだけで, その後の文も順番に処理する. これは xTrics において各操作は独立していると考えるためである. 例えば, 図 5 の XML データに対して

1) $xml=insert($xml,#0.para,title[“鬼ノ城”]);
2) $xml=delete($xml,0.para(-1));

のような xTrics があったとき, 1) は “鬼ノ城” というタイトルを入れる操作であり, 2) は最後のパラグラフを消す操作である. この 2 つに明確な関係はなくお互いの操作は独立しているといえる.

しかし, 例えばの次のようにある範囲にタグを追加する xTrics を考える.

1) $str=$xml[0.para(-1).0.28:0.para(-1).0.61];
2) $xml=delete($xml,
   0.para(-1).0.28:0.para(-1).0.61);
3) $xml=insert($xml,#0.para(-1).0.28,
   marker@color=“blue”[$str]);

このとき, もし 1),2) が成功して 3) が失敗すると指定範囲の文字列が消えてしまう. このため明確な関係のある操作を 1 つの論理単位として, すべて成功するか試行する処理が必要となる. このため xTrics においては, 試行をする try 文を以下のように定義している. また,try 文内で明示的に例外を投げる exception 文も定義している.

try 文 := *try* 式

exception 文 := *exception*;

ここで try 文は, 式内のいずれかの文が失敗するか明示的に例外が投げられると, この試行は失敗として, 試行前の状態に戻す. つまり前例において try{1)2)3)} としておけば, もし 3) で失敗しても文字列が消えてしまうことがなく, すべて成功すれば範囲にタグを追加することができる xTrics となる.

4 xTrics の XML 表現

3 節で述べた xTrics は, プログラミング言語であり, 複数の XML ファイルに対して操作を行うことに関しては強力な言語であるが, 2 節で述べた提案手法のように XML 文書に対して, スクリプトを付加したり, 特定の文書のみに利用者の意図を反映する場合には, 多少冗長な部分もある.

そこでこの節においては,xTrics に対応した XML のタグセットを定義して, XML 文書中に埋め込んで xTrics を利用できるようにする xTrics の XML 表現 xTrix(xTrics XML) を提案する. この xTrix タグセットにおいては, 特定の XML 文書を対象に操作を行うため, 対象の XML 文書の URL を指定する. つまり,<xTrix target="URL"></xTrix>で xTrix を定義する. このとき target 属性は省略可であり, 省略した場合はこの xTrix を埋め込んだ XML 文書を対象に操作を行う. そして xTrix の要素値として, 以下のような操作タグを入れる.

- <indexing value="index"
  variable="変数名"/>
- <slice start="index" end="index"
  variable="変数名"/>
- <insert place="B-index">
  挿入する情報
  </insert>
- <delete place="index"/>
  <delete place="start:end"/>
- <changeElement place="index"
  name="タグ名"/>
- <changeAtt place="index"
  name="属性名" value="属性値"/>
- <deleteAtt place="index"
  name="属性名"/>

操作タグは基本的に空要素タグであり, 操作

タグを列挙することによりスクリプトを実現する． ここで,insert タグだけは挿入するデータを要素値として持つ． このデータとしては文字列,XML 文章があり，その他に外部の XML 文書を読込む<loadXML file="URL"/> と変数を読込む<variable name="変数名"/>がある．

この他に制御フローとして以下のタグがある．

- <if condition="条件式"></if>
- <while condition="条件式"></while>
- <break/>
- <next/>
- <redo/>
- <try></try>
- <exception/>

分岐の制御フロータグは if タグのみで，繰返しは while タグのみである．

以下に xTrics を埋め込んだ XML の例を示す．

```
<section>
 <para>
  鬼ノ城は,663 年の白村…
  …られています．しかし，
  <marker color="red">
    『日本書…たくなく
  </marker>
  ，これまで…てきました．
 </para>
 <para>
  国指定史跡である…
  …実施してきました．
 </para>
</section>
<xTrix>
 <indexing value="0.0.0" variable="x1"/>
 <indexing value="0.0.1.0" variable="x2"/>
 <indexing value="0.0.2" variable="x3"/>
 <indexing value="0.1.0" variable="x4"/>
 <delete place="0"/>
 <insert place="#0">
  <鬼ノ城>
   <variable name="x1"/>
   <variable name="x2"/>
   <variable name="x3"/>
   <variable name="x4"/>
  </鬼ノ城>
 </insert>
</xTrix>
```

この XML 文章の xTrix を適応すると次のような XML 文章になる．

```
<鬼ノ城>
 鬼ノ城は,663 年の白村江の戦いで唐・新羅の連合
 軍に大敗し，その連合軍が日本に侵攻する危機感
 から，国土防衛のため，北九州～瀬戸内沿岸～畿
 内にいたる西日本各地に築いた，古代山城の一つ
 と考えられています．しかし，
 『日本書紀』などに築城などの記載はまったくなく
 ，これまで謎の城といわれてきました．
 国指定史跡である鬼ノ城を将来的には整備をしていくため，いまだ不明の点が多いことからまず基礎データを得るための発掘調査を実施し，その成果をもとに整備計画を進めていく方針で，これまでに 3 回にわたる発掘調査を実施してきました．
</鬼ノ城>
```

5 おわりに

本稿では, 従来の個人化手法で用いられている, 提供者側で情報を個人向けにカスタマイズする方法ではなく, 利用者側で提供者の意図と利用者の意図との差分を取り, 差分情報をフィルタとして見ることで, ネットワーク上の情報を利用者個人個人でカスタマイズできる個人化手法を提案した. 差分情報の記述として, われわれの考案した xTrics を用いた. xTrics は XML を木構造として操作することに特化したスクリプト言語であるが, 「文書」としての XML を扱うことも考慮している．

また本稿で提案する手法を用いれば, 第三者の差分情報を重ねてフィルタリングすることで, 複数の人の意図を適用した情報を動的に生成することもできる．

参考文献

[1] 清光英成, 竹内淳, “Web データの個人化と環境適応”, DBWeb2000,pp.157-164, 東京,2000 年 12 月 6-8 日.

[2] 竹内淳記, 清光英成, 田中克己, “アクティブルールに基づく Web 個人化・環境適応システム ActiveWeb の実装”, 電子情報通信学会データ工学ワークショップ, 5B-1, 伊豆熱川,2001 年 3 月 8-10 日.

[3] 前田葉子, 遠山元道, “ACTIVIEW:SuperSQL を利用した適応型表示ビューの実現”, 電子情報通信学会データ工学ワークショップ, 5B-7, 伊豆熱川,2001 年 3 月 8-10 日.