

多チャンネル対応コンテンツ生成方式の開発 (2)

6Z-03

— マルチテンプレート管理システム —

石田 和生 矢野尾 一男 小川 隆一
NEC インターネットシステム研究所

1 はじめに

放送のデジタル化や携帯電話の普及とともにコンテンツを配信するチャンネルの多様化が進んでいる。その結果、コンテンツへのアクセス性を高めるために同一コンテンツを複数チャンネルで配信することが望まれている。しかし、各チャンネル毎にコンテンツを用意していたのではコンテンツの制作、管理コストが増大するため、ひとつのソースデータをスタイルシートやテンプレートなどで各チャンネル向けに変換する手法が利用されている [1][2]。

前記手法はソースデータはひとつであるがテンプレート自体は各チャンネル毎に必要である。このため、コンテンツのスタイルや構成を変更する際には個々のテンプレートを個別に修正することになるが、これは一般に高度のスキルと多くのコストを必要とする。そこで、筆者らが現在開発中の多チャンネルコンテンツ変換サーバ [3] では、複数のテンプレートを一括管理するマルチテンプレート管理方式を検討中である。本方式はひとつのテンプレートの変更点を他のテンプレートに自動的に反映させることで、個別にテンプレートを修正する手間を省くことが出来、また、修正ミスや修正洩れを防ぐという効果がある。さらに、特定のテンプレートを修正するだけで他のテンプレートも変更されるため、全てのチャンネル用のテンプレートを理解していない者でも修正作業を行うことが可能となる。

2 マルチテンプレート管理方式

本章では、XML 形式のソースデータと XSLT で記述されたテンプレートを対象に、本方式の処理内容を説明する。

あるひとつのテンプレートの修正を他のテンプレートに反映させるためには次のような機能が必要となる (図 1)。

1. テンプレートの変更点の抽出
2. 他のテンプレート内での反映場所の探索
3. 他のテンプレートへの変更点の適用

以下では上記 3 つの内容と、それぞれを実行するために必要なツリー変換の手法について説明する。

2.1 ツリー変換と変更点の抽出

あるひとつのテンプレートが変更された時に全てのテンプレートを一括修正するためには、まず、

Development of Multi-channel Content Authoring Method (2)
— Multi Template Management System —
Kazuo ISHIDA, Kazuo YANOO, and Ryuichi OGAWA
Internet Systems Research Laboratories, NEC Corp.

その変更点を抽出する必要がある。このための方法としては大きく分けて、変更前後のテンプレートを直接比較する方法と、別形式に変換して比較する方法の 2 つが考えられる。後者の方式の方がロバスト性を高くすることが可能なので、ここではテンプレートからキー情報を抽出してツリー構造へと変換して比較を行う。キー情報としては、ソースデータとシステムが出力するコンテンツデータに強く依存した情報である

1. ソースデータに含まれるタグ名、属性名
2. 出力データに含まれるタグ名、属性名

を選択する。また、ツリーへの変換は、基本的に、元のテンプレート上での親子、兄弟関係と矛盾しないようにキー情報をツリーに構成することで行う。例えば、図 2 に示されるような HTML 形式のコンテンツデータを生成する XSLT テンプレートは図 3 に示されるようなツリーに変換される。

変更点の抽出処理は変換されたツリーを用いて行う。すなわち、変更前後のテンプレートそれぞれを変換したツリー同士で共通する部分木を削除し、残った部分木を変更点として抽出する。例えば、図 2 のテンプレートの点線で囲まれた部分に変更によって追加された部分であるとすると、図 3 の点線部分がテンプレートの変更点として抽出される。

2.2 反映場所の探索と変更点の適用

変更点が抽出できると、次はこの変更を反映させたい対象のテンプレートに対して、変更点を反映させるべき場所の探索を行う。この探索も、前節

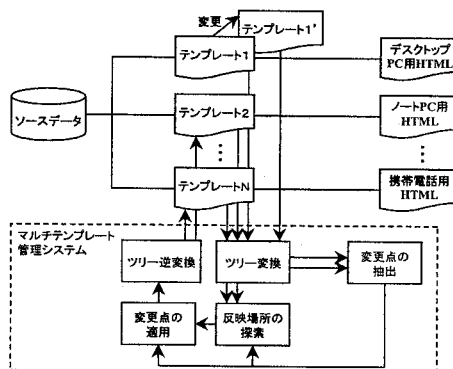


図 1: システム構成

```

<xsl:template match="link">
  <P><xsl:value-of select="description"/><BR>
  <A>
    <xsl:attribute name="HREF">
      <xsl:value-of select="data/target"/>
    </xsl:attribute>
    <xsl:value-of select="data/name"/>
    <IMG>
      <xsl:attribute name="SRC">
        <xsl:value-of select="data/image"/>
      </xsl:attribute>
    </IMG>
  </A>
</P>
</xsl:template>

```

図 2: 修正するテンプレート

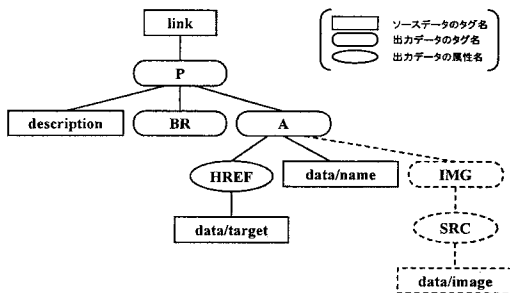


図 3: ツリー変換結果

と同様、テンプレートを直接探索対象とする方法と、変換した別形式を探索対象とする方法の2つが考えられる。しかし、前者の方法は、管理している全てのテンプレートの形式が同一である場合には問題ないが、形式が異なる(例えば、XSLTとJSPなど)テンプレートが混在している場合には、両者の対応をとることが難しく、探索は容易ではなくなる。そこで、本方式では、前節で変換したツリーをベースに、変更点として抽出された部分木の探索を行うことで反映場所の探索を実現する。

図2のテンプレートを修正して、点線部分が追加された場合を例として手順を説明する。このとき、変更を反映させたいテンプレートは図4のようなものであるとする。まず、図4のテンプレートを変換し(図5の実線部分)、次に、変更点として抽出された図3の点線部分の親ノード(A)と親ノード以下のノード(HREF, data/target, data/name)からなる部分木に注目する。この部分木にマッチする変更対象ツリー(図5の実線部分)内の部分木が変更点を反映させるべき場所となる。マッチする部分木が存在しない場合は、反映させるべき場所がないと判断し変更は行わない。最後に、探索された場所に変更点として抽出された部分木を追加して、最終的に図5のようなツリーが

```

<xsl:template match="link">
  <A>
    <xsl:attribute name="HREF">
      <xsl:value-of select="data/target"/>
    </xsl:attribute>
    <xsl:value-of select="data/name"/>
  </A>
<BR>
</xsl:template>

```

図 4: 修正を反映するテンプレート

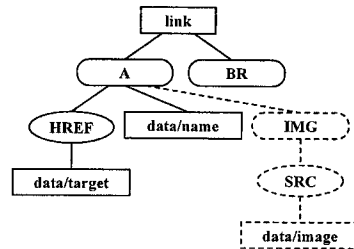


図 5: ツリーへの修正の反映

得られる。後は、ツリー変換の逆手順によって、このツリーからテンプレートを生成すれば目的とするテンプレートの修正作業が完了することになる。

3 おわりに

本報告では、テンプレートを用いたコンテンツ生成の問題点であるテンプレートの修正、管理コストを削減するマルチテンプレート管理方式を提案し、本方式を実現するために必要な機能である、変更点の抽出、反映場所の探索、変更点の適用の3つの処理方法について述べた。本方式は複数のテンプレートを一括して管理し、テンプレートに修正が必要となった場合にはあるひとつのテンプレートを変更するだけで、全てのテンプレートにその変更点が反映される。これにより、修正コストの削減が可能になる他、システムで一括して全てのテンプレートを変更するため、テンプレートの修正洩れや修正ミスが発生する可能性を低くすることも可能となる。今後は、本方式に基づいたシステムの試作と実証を行うとともに、XSLT以外の形式のテンプレートを対象とする手法について検討する予定である。

参考文献

- [1] NEC, モアレッシモ, <http://sw.sie.mt.nec.co.jp/cced/moirissimo/>, 2001.
- [2] Oracle, Portal-to-Go, <http://www.oracle.co.jp/ptg/top.html>, 2000.
- [3] 小川 他, 多チャンネル対応コンテンツ生成方式の開発(2) — コンテンツ変換サーバのアーキテクチャ —, 第64回情処全大, 6Z-02, 2002.