

# 部品化したデータによる静的ウェブ発行システム<sup>1</sup>

3A-02

井上由文<sup>2</sup>イージャパン (株)<sup>3</sup>

## 1. はじめに

ウェブサイトにより情報を提供される場合、HTML ファイルにより公開することが一般的であり、多くの情報が HTML ファイルにより提供されている。そのため、なウェブサイトを構築、管理するためのシステムの多くは HTML ファイルを管理の基本単位としている。

ソフトウェア構成管理システム (Software Configuration Management System、以下、SCMS) である CVS [1]、VSS[2]、Perforce[3] は HTML ファイルの管理も行えるため、これらのシステムを使ってウェブ発行を管理することができる。ここでは、サーバサイドインクルード (以下、SSI) などの機能を使って HTML データを部品として取り扱っている。

コンテンツ管理システム (Content Management System、以下、CMS) Team Site[4]、Vignette,[5] Microsoft Content Management Server[6] はウェブ発行に特化システムで、ウェブサーバ上でページテンプレート等を処理することにより HTML データの取り扱いを SCMS よりも行いやすくしている。

しかし、HTML データは情報と表現が渾然一体となっており情報の二次加工が困難である。XML[7] と XSLT[8] は、情報と表現の分離を可能にし、多様な表現、コンテンツ交換 (ICE[9])、メタデータ (RDF[10]) の抽出が容易に行える。

本稿で述べるデータの部品化は XML、XSLT を用いることにより情報と表現の分離を行う方式である。本方式では、ニュースデータやリンク集などのコンテンツデータの部品化に加えて、ページの構成やナビゲーションデータの部品化も行い、ウェブ発行を部品の組み合わせによりおこなう。

以下ではデータの部品化の方法、部品からウェブ発行するための方法、そして実装システム Aqua の概要とその評価について述べ、本方式の有効性について考察する。

## 2. メタデータの部品化

ニュース記事やリンクデータなどのコンテンツデータを XML で表現し部品化することは直感的にできる作業であるが、ページのように情報を提供するためには必要ではあるが、コンテンツとして考えにくいもいわゆるメタデータの部品化を直感的に行うことは難しい。ページはデータ部品を含む複合部品として考えることができるが、図 1 の左に示す XML データのような記述になってしまうことが多い。

ページがこのような記述になりやすいのは、ページのデザインが出来上がった後で、ページの記述を行うからである。この例では左右二つのカラムに分かれたページを XML で表現している。しかし、この表現は携帯端末のように表示幅が狭く二つのカラムを表示できない場合や、紙のように多くのカラムを使うことをできる場合も表現しているとは言えない。

<sup>1</sup> Static Web Publishing System Using Modularized Data

<sup>2</sup> Yoshifum INOUE

<sup>3</sup> eJapan K.K.

HTML が情報と表現を含んでいることから、その取り扱いが難しくなったように、表示形態に依存した情報を部品にいれると、部品の再利用性が失われることになる。同様にサイト内共通要素である、ページヘッダー、ページフッター、ナビゲーションバーや、メニューも表示形態に依存したデータである。

ページヘッダーやページフッターは、ページのデザインでは上下に分かれているが、実際にはサイトの名前、著作権情報等のサイトのメタデータもしくは、ページ名、ページの作成日時名等ののページのメタデータである。ナビゲーションバーやメニューはサイト内へのリンク情報を持つデータであり、その他のリンクもやはりリンクデータであり、そのように部品化する。

ページへのコンテンツデータの組み込みもコンテンツの重要度やページが扱うカテゴリなど意味情報について記述することにより、ページそのものにデータとしての意味をもたせることが可能である。これはページデータを HTML ファイルに変換するプロセッサに多くの情報を渡すことになり、変換プロセッサにおいて出力デバイスに応じた最適化が行いやすくなる。

最適化の例としては携帯端末用の出力には優先度が低いコンテンツをリンクに代えて、ファイルサイズを小さくするとか、PC用の出力ではコンテンツがマルチメディアデータを持っている場合には、それらも HTML ファイルと同時に提供するといったようなことがある。

ページ部品のようなデータ部品を含む部品は複合部品はページを定義するだけでは無く、コンテンツリスト、インデックスやカタログなどのメタコンテンツを含む部品にも応用することも可能である。この場合はシステムにより自動的に生成することも可能であり、自律部品 (automated component) と呼ぶ。もちろん自律部品であっても自律して動作するための情報は与える必要がある。

<pre> &lt;page title="トップページ"&gt; &lt;header ref="header-data"/&gt; &lt;navigation ref="navigation"/&gt; &lt;left&gt; &lt;box title="ニュース"&gt; &lt;component ref="news-1"/&gt; &lt;component ref="news-2"/&gt; &lt;/box&gt; &lt;/left&gt; &lt;right&gt; &lt;box title="スポーツ"&gt; &lt;component ref="sports-1"/&gt; &lt;component ref="sports-2"/&gt; &lt;/box&gt; &lt;/right&gt; &lt;footer ref="footer-data"/&gt; &lt;/page&gt; </pre>	<pre> &lt;page title="トップページ" meta-data="site-data" &gt; &lt;category class="navigation"&gt; &lt;component ref="main-links"&gt; &lt;/category&gt; &lt;category title="ニュース" priority="1"&gt; &lt;component ref="news-1"/&gt; &lt;component ref="news-2"/&gt; &lt;/category&gt; &lt;category title="スポーツ" priority="2"&gt; &lt;component ref="sports-1"/&gt; &lt;component ref="sports-2"/&gt; &lt;/category&gt; &lt;/page&gt; </pre>
---	--

図1: ページの XML 表現

### 3. データ部品によるウェブ発行

データ部品によるウェブ発行では、部品を集めてページを構成することと、そのページを出力形式に例えば HTML やテキストに変換する、と言った二つの段階により行うことができる。

この処理は make[11] コマンドと同様の処理であり、部品がソースファイル、コンパイラが XSLT プロセッサ、HTML ファイルがオブジェクトファイル、ページ部品が makefile に相当すると考えることができる。この処理には make コマンドと同様に各部品の更新時間と HTML ファイルとの時間を比較することにより、更新が必要な部品のみを XSLT プロセッサで処理して、HTML ファイルを生成することが可能になる。

多くの場合、部品は共有されているため、更新時刻による更新が必要な HTML ファイルの自動検出は、無駄な処理を行わないために重要である。

#### 4. 実装システムとその評価

本方式はコードネーム Aqua として Windows 2000 サーバ上に実装した。データ部品は XML により表現し、SQL Server に ntext データとして格納した。データ部品へのアクセスは DCOM を用いて SQL Server へのアクセスのキャッシングとラッピングを行うことにより、データ部品を利用するプログラムが SQL Server の存在を意識することなく、プログラムを作成できるようにしている。

データ部品は websh と呼ぶ Web ベースのユーザインタフェイスか、AO XML と呼ぶスキーマに基づく XML データファイルを aql (aqua load) コマンドを使い Aqua に入力する。WebSh の入力編集フォームは XML スキーマにより自動生成されるようになっていた。そのため、サイト設計者はコンテンツ情報に適したスキーマを設計し利用することが可能になっている。

ウェブサイトへのコンテンツの発行は aqm (aqua make) コマンドが前述のように make コマンドの処理と同様に依存する部品の更新時刻に基づき発行を行う。この作業はウェブサーバでページにアクセスがあったときに行うのではなく、別サーバで行われファイルをウェブサーバに転送することによって行っている。

Aqua による本方式の実装では XSLT も部品化して取り扱うようにしている。XSLT はスキーマと出力形式により部品化されており、データ部品のスキーマと出力形式に基づき部品化したままとめて XSLT プロセッサに渡すようにしている。そのため、コンテンツスキーマの追加や変更を行った場合でも対応する XSLT データの追加変更を行うだけでよい。また、aqm は XSLT データの更新情報も管理しているため、変更があったファイルだけが発行しなおされる。

Aqua によるファイルの発行時間は、シングルプロセッサ Pentium III (750MHz) の PC 上で、約 5 万ファイルを約 40 分で行う。実際のウェブサイトの日々の運用では一日にせいぜい数十ファイルが更新されるのみなので、運用中に発行時間を気にすることは少ない。

Aqua を採用したサイトはいくつかあり、それぞれ数万を超えるファイルを扱い、コンテンツをほぼ毎日更新している。それぞれのサイトは一日数万アクセスあるが、静的なウェブページの発行により、ウェブサーバを少ないリソースで安定稼働させている。

#### 5. データ部品の管理

Aqua ではデータ部品の管理は、Aqua Space と呼ぶファイルシステムのような階層構造上にデータ部品を配置することによって行っている。また、この階層構造ではシンボリックリンクの利用も可能であり、データ部品を多様な視点から分類することを可能にしている。例えば、Yahoo のようにカテゴリ分類されたリンク集は、リンクデータをデータ部品として扱い、この階層構造上に自然に表現することができる。

Aqua Space は websh からアクセスする他に aqsh (aqua shell) と呼ぶ、コマンドラインツールからもアクセスできる。aqsh では ls, rm, mv などの Unix の基本コマンドと同じ名前のコマンドが利用可能で、ファイルシステムを扱っているのと同じ感覚でデータ部品を操作することができる。

Aqua Space はまた自律部品へのデータの提供も行っており、Aqua Space の階層構造を自律部品からアクセスし、コンテンツリストやカテゴリツリーのデータを作成し、それをウェブサイト発行することもできる。

まとめ

本稿ではデータを部品化してウェブ発行を行うために、ページに含まれるメタデータの部品化について述べた。本方式ではサイトをXMLとして記述することが可能であり、XSLTプロセッサ等のXMLの周辺技術を利用することによりウェブ発行が簡単に行うことができる。また、サイトを記述したXMLファイルからRDFなどの別のXML表現に変換することも可能である。

本方式を実装したシステムAquaは幾つかの数万ページを超える巨大サイトでの運用実績があり実用上問題が無いことも確かめられている。これらのサイトでは静的にページを発行しているためウェブサーバへの負荷が少なく、高付加に起因するサイトの不安定性がないことも要因の一つとして考えられる。

また、あるサイトではコンテンツデータの部品はそのまま、XSLTのみを変更することにより、サイトのデザインの変更が行うことにより、データの最良を可能にしたデータ部品化の利点を利用することができた。

なお、今回の実装では静的なウェブ発行のみを行っているが、部品に対するアクセスはDCOMを使ってプログラムに公開しているので、DCOMを利用してウェブサーバ上でユーザのリクエストに応じてHTMLデータを生成することも可能である。また、Cocoon[12]のようなXMLをベースとしたアプリケーションサーバや、Active Server Page (ASP)、または、ASP.NETのようなアプリケーションサーバと組み合わせることも可能であり、BBSのように即時性があるサービスの提供も可能である。

## 6. 参考文献

- 1) CVS, Concurrent Versioning System, <http://www.cvshome.org/>
- 2) VSS, <http://msdn.microsoft.com/ssafe/>
- 3) Perforce, <http://www.perforce.com/>
- 4) Team Site, <http://www.interwoven.com/products/teamsite/>
- 5) Vignette, <http://www.vignette.com/>
- 6) Microsoft Content Management Server, <http://www.contentmanagementserver.com/>
- 7) XML, <http://www.w3.org/XML/>
- 8) XSLT, <http://www.w3.org/XSLT/>
- 9) RDF, <http://www.w3.org/RDF/>
- 10) ICE, <http://www.w3.org/TR/NOTE-icc>
- 11) make, <http://www.gnu.org/software/make/make.html>
- 12) Cocoon, <http://xml.apache.org/>