

XML を用いたデータベース集約型 Web 環境の提案

嵐 大亮 † 遠山 元道 ‡

† 慶應義塾大学大学院 理工学研究科 開放環境科学専攻

‡ 慶應義塾大学 理工学部 情報工学科

E-mail: † arashi@db.ics.keio.ac.jp, ‡ toyama@ics.keio.ac.jp

ショッピングサイトなどの Web サイトでは、クライアントのリクエストに応じた出力結果をレスポンスとして返すために、Web サーバ内で動的に Web ドキュメントを生成している。このような Web サイトを作成するためには、従来では CGI/Perl や PHP 等の Web アプリケーション用のスクリプト言語を用いて、クライアントのリクエストに応じたページ生成を行っていた。しかし、(1) 配信コストが大きい、(2) メンテナンスが困難である、(3) 作成するために高度な技術が必要であるといった問題が挙げられる。本研究では、動的 Web サイトで用いるデータを Web サーバ内部に XML で保持しておくことによって、配信コストの低減、スキーマの柔軟な変更を可能にしたデータベース集約型 Web 環境を提案する。

キーワード : Web、XML

The proposal of the web environment unified with database using XML

Daisuke ARASHI † Motomichi TOYAMA ‡

†School of Science for OPEN and Environmental Systems,
Faculty of Science and Technology, Keio University.

‡Department of Information and Computer Science, Faculty of Science and Technology,
Keio University.

E-mail : †arashi@db.ics.keio.ac.jp ‡toyama@ics.keio.ac.jp

In many websites, such as a shopping site, the web document is dynamically generated within a server in order to return the output result according to the request of a client. We made the website generating the page according to the request of a client by using the script language for web applications. However, there are some weak points in such technologies. First, it has large distribution cost. And a maintenance is difficult in this system because of a database and application are used together. Finally, advanced technology, such as a script language and database management, is required. In this paper, we propose the web environment unified with database in order to reduce the distribution cost and to enable flexible change of a data schema by holding contributed data in the XML format.

keyword : Web , XML

1 背景

近年ではインターネットを利用したショッピング、情報検索、翻訳など様々なサービスが展開されている。それらのサービスは主に Web 上で行われており、クライアントからの入力に対して、動的にレスポンスを返すような動的 Web サイトにより実現されている。現在ではインターネット上において、もっとも身近なデータベースのインターフェースは Web であり、私達は動的 Web サイトを通じて Web 上から様々な情報を探索、追加、更新することが出来るのである。そのような動的 Web サイトは HTTPd である Web サーバと、RDB への探索やドキュメントの整形を行う Web アプリケーションで構成されている。しかし、動的 Web サイトでは、RDB への探索や各種ファイルへの I/O を行いドキュメントを生成するため、配信コストが大きい。また、動的 Web サイトでは、一つの Web ページを作成するために、RDB、Web アプリケーション用のスクリプト、そして場合によっては別途 HTML を用意する必要がある。ここで、もしページのレイアウトの変更や、スキーマの変更を行おうとした場合、RDB のスキーマの変更だけでなく、スクリプトまでも変更しなければならない。このような動的 Web サイトを作成するためには、Web の知識だけではなく、HTML 等の Web ドキュメントのマークアップ言語、Web アプリケーション用のスクリプト言語、そしてデータベースの知識が必要となる。

最近では Web サイトの低寿命化に伴い、動的 Web サイトの製作サイクルも短くなりつつある。従来の RDB を利用した動的 Web サイトでは、データベースのスキーマの観点から Web サイトを作成してゆくのが主な手法であったが、Web サイトによってはデータの入出力から動的 Web サイトを構築した方が有利な場合もある。例えばアンケートのページなどは、アンケートの内容を変更する際に、データベースのスキーマを変更し、HTML のフォーム、スクリプト共に変更しなければならない。そこで本研究では、XML を用いたデータベース統合型 Web 環境を提案する。Web サーバがデータ管理機能を有することによって、従来の動的 Web サイトで必要であった Web アプリケーションを省略し、XML を利用することで動的 Web サイトで扱われるデータの柔軟なスキーマの変更を可能にし、クライアン

トへの配信コストを低減するのが目的である。

以下、??章で関連研究を紹介し、??章で本研究の背景である動的 Web サイトについて述べ、??章で本研究の目的について説明し、??章で本研究で提案するシステムについて述べる。その後 ??章で今後の課題を示す。

2 関連研究

Anindya らは動的 Web サイトの高速配信として、Proxy サーバを利用した分散配信を提案した [?]。Proxy サーバによる分散配信を可能とするため、Web ドキュメントに独自のマークアップを施し、クラスタリングを行い、そこで作成された HTML フラグメントを Web サーバ側で一括管理している。Ricca らは HTML 等によってのみ構成されている静的 Web サイトを、動的 Web サイトに変換する手法を提案した [?]。この研究の特徴としては動的 Web サイトへマイグレーションを行うだけでなく、その変換プロセスの過程で、複数の言語で記述された Web ドキュメントから成る Web サイトを一つの動的 Web サイトに集約することも可能であった。Titchkosky らは動的 Web サイトを実装する際に主に用いられる各技術について様々な条件下で比較検討を行った [?]。比較検討したのは、CGI/Perl, PHP (apache1.3, apache2.0), Server side java (Resin, Jetty, Tomcat) である。これらの動的 Web サイトの技術は、状況によってパフォーマンスが大きく異なることを示した。Mignet らは、現在 Web 上で用いられている XML についてサーベイを行った [?]。現在、Web 上でどのような形で XML が用いられているのか、そしてその XML データの形式について定量的に分析を行った。

3 動的 Web サイト

通常、Web サーバはクライアントのリクエストに応じて静的なドキュメントを配信するだけである。HTML 等の Web ドキュメント内に用意されたフォームから、クライアントが Web サーバへデータを投稿した場合、HTTPd 単体ではそれを処理することが出来ない。そこで、現在の動的 Web サイトのように、クライアントの入力に従って出力を動

的に生成するような工夫がなされてきた。

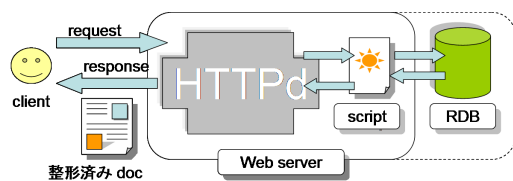


図 1: 動的 Web サイトの概要

通常、動的 Web サイトは図 ?? に示すようなシステムとなる。図 ?? のスクリプトの箇所はシステムによって異なる。

3.1 CGI

古くから用いられてきた動的 Web サイトの手段として、CGI (Common Gateway Interface) がある。その仕組みは、クライアントの投稿データおよびクライアントの環境変数を、要求されたリソースへ標準入力を通して渡すというものである。ここでそのリソースが実行可能なプログラムであったら、別プロセスで実行され、標準出力の内容を HTTPd へ渡し、Socket を通じてクライアントへ返す。主に Perl, C 言語が用いられるが、どのような言語、プログラムでも構わない。

CGI を利用して RDB へ探索を行うような動的 Web サイトを作成する場合、このスクリプトが RDB への探索、クライアントへ返すための Web ドキュメントの生成を行う。動的 Web サイトのシステムとしては非常に原始的ではあるが、クライアントへの出力は自由であり、またどのようなプログラムでも動かせることから、柔軟な動的 Web サイトを作成することが可能である。

3.2 PHP, ASP, JSP

CGI の仕組みは、リクエストがあったリソースを実行して、その結果をクライアントへ返すというものであり、そもそも動的 Web サイトのためのシステムではなかった。そこで次に動的 Web サイトの技術として台頭してきたのが JSP (Java Server Pages), PHP (Hypertext Preprocessor), ASP (Active Server Pages) 等の技術である。これらの技術は、Web ドキュメント内にプログラムコードを記述し、クライアントへそのドキュメントを配信する

際に、コードを実行し、実行結果とプログラムコードを置き換えることで動的 Web サイトを実現している。そのため、CGI に比べて動的 Web サイトの作成が容易である。各技術毎にデータベースを利用するモジュール等が実装されている。

4 研究目的

インターネット上で情報探索を行うには、クライアントのリクエストを受けて、Web アプリケーションがデータベースへ問い合わせを行い、その結果を Web ドキュメントに成型してクライアントへ返す。このような動的 Web サイトには、幾つかの欠点が挙げられる。

1. Web サーバ側の配信コストが大きい
2. ドキュメントのレイアウト変更やメンテナンスが困難となる
3. 動的 Web サイトの構築は、様々な技術を考慮しなければならないため、高度な技術が必要である

そのため、従来の動的 Web サイトの環境では、アンケートのページ等を生成する際、アンケート毎にスキーマを設定し、入力フォームを作り、それを処理するスクリプトを書かなければならなかった。Web 上では予めフォームが用意されている場合が殆どで、ある程度は決まった形のデータがフォームから送信され、それをスクリプトを介してデータベースへ格納している。その Web 上でのデータの入出力に着目し、Web サーバとデータベースを統合することで、作成およびメンテナンスが容易で高速な動的 Web サイトの環境を提案する。

5 データベース集約型 Web 環境

本研究では HTTPd とデータベースの統合環境を提案することによって、?? 章で述べた目的を果す。図 ?? に本システムの概要を示す。

従来の動的 Web サイトのアーキテクチャーとの大きな違いは、スクリプトを使わないことである。Web サーバにデータを管理する機能を予め備えておくことによって、スクリプトを使わずに動的 Web サイトを実現できる。データの管理方法として、XML

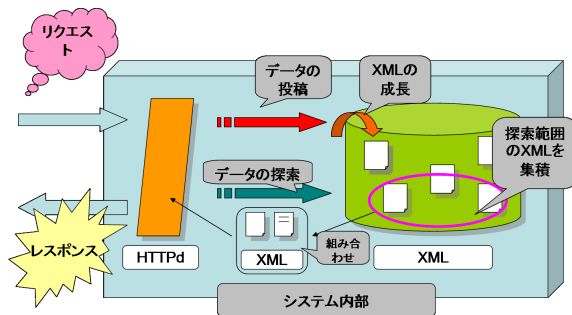


図 2: システムの概要

を利用する。その理由としては、Web ドキュメントに強い親和性があること。そして、データの交換用フォーマットとして定着していること、スキーマの変更が柔軟であることである。

5.1 クライアントからの投稿処理

従来の動的 Web サイトでは、クライアントからの投稿先が実行されるスクリプトであり、そのスクリプトへデータを渡していた。しかし、本研究ではそのようなスクリプトを使用しないため、データベースを統合した Web サーバとして、以下の URI を次の表 ?? のように再定義する。

<http://www.db.ics.keio.ac.jp/arashi/paper.cgi>

表 1: URI による DB の指定

URI の要素	例	指定先
ホスト名	www.db.ics.keio.ac.jp	DB サーバ名
path 中の dir	arashi/	DB の指定
ファイル名	paper.cgi	table の指定

クライアントからの投稿を、スクリプトを使わずにデータとして格納できるように、リクエストの URI をこのように見立てる。これにより、クライアントから本システムを見た際に、従来の動的 Web サイトと変わらずにデータの投稿、探索を行うことが出来る。

各ディレクトリをデータベース、各ファイルをテーブルだとすると、スキーマの定義と受け取ったデータのマークアップの定義が必要である。本研究ではデータを XML で管理するため、XML のスキーマは DTD で定義し、マークアップの定義はディレクトリに独自の設定ファイルを置き、システムはそれ

を読み込んで記述されたとおりに投稿データを格納する。ここで、例として健康に関するアンケートの Web サイトを採り上げ、図 ?? にスキーマ定義 (DTD) を示し、マークアップ定義のフォーマットを図 ?? に示す。(この Web サイトは後に動作サンプルとして再び紹介する。)

<!-- 健康アンケート -->

```
<!DOCTYPE 健康 [
  <!ELEMENT 健康 (member*)>
  <!ELEMENT member ( 名前, 性別, 喫煙, 睡眠時間 )>
  <!ELEMENT 名前 (#PCDATA)>
  <!ELEMENT 性別 (#PCDATA)>
  <!ELEMENT 喫煙 (#PCDATA)>
  <!ELEMENT 睡眠時間 (#PCDATA)>
```

]>

図 3: アンケートページの DTD

```
%q1.xml
<member>
  <名前>#name</名前>
  <性別>#sex</性別>
  <喫煙>#smoke</喫煙>
  <睡眠時間>#sleep</睡眠時間>
</member>
%q1.xml
```

図 4: マークアップ定義

フォームからの投稿は、定義されたマークアップフォーマットで、DTD に従った形で利用される。

5.2 クライアントからの探索リクエスト

データの投稿処理については上記のとおりであるが、クライアントからのリクエストがデータの投稿ではなく、データの探索であった場合について説明する。

動的 Web サイトを表現するために、本システム上でのサイト作成者は XML を用いて動的 Web サイトを作成する。投稿されたデータは、設定されたディレクトリに保存され、そのデータを外部実体参照として取り扱う。

現段階で実際に動作しているサンプルについて説明してゆく。

5.2.1 アンケートページ

Web 上には多量のアンケートが設けられている。Web ではクライアントからの入力が簡便であること、投稿内容をそのままデータとして保存できることなど、メリットは大きい。そのようなアンケートのページを作る際、従来の Web サイトであれば、

1. RDB のスキーマ設計
2. スキーマに基づいた (HTML) フォームの作成
3. RDB への探索、更新、そして Web ドキュメントを整形するためのスクリプトの記述

手順を踏んで作らなければならないが、提案する環境では、

1. スキーマ設計 (DTD) と マークアップ定義
2. スキーマに基づいた XML と その XML をフォーム付き Web ドキュメントへ整形するための XSLT の作成

だけで作成することが出来るため、サイトの製作コストが非常に低く、Web アプリケーションのプログラムが動作せずに配信できるため、高速である。そしてスキーマの頻繁な変更にも柔軟に対応出来る。本システム上で動作しているアンケートページの一例のスクリーンショットを図 ?? に示す。

このように、サーバ側の環境は従来のシステムとは異なるが、クライアントから見れば通常の動的 Web サイトと同様に振舞うのである。配信におけるシステムでの内部処理を図 ?? に示す。投稿されたデータは、指定されたマークアップを施されて XML として格納される。現段階では、保存されている XML は一定のファイルサイズに分割されて保存されている。その分割された XML の何処に欲しいデータが格納されているのかをサイト作成者が認識することは困難であるため、投稿データを保存している XML と、サイト作成者が用意するクライアントへ配信する XML との間を本システムが仲介する。サイト作成者は、クライアントへ配信する XML を作成し、そこに外部実体参照として本シ

健康アンケート

お名前	<input type="text"/>
性別	<input type="radio"/> 男性 <input type="radio"/> 女性
喫煙	<input type="checkbox"/> 吸う
睡眠時間	5時間以下 <input type="button" value="↓"/>
<input type="button" value="送信"/> <input type="button" value="リセット"/>	

アンケート結果

お名前	性別	喫煙	睡眠時間
嵐 大亮	男性	<input type="radio"/>	7h
嶋村 勇志	男性	<input type="checkbox"/>	9h

図 5: アンケートページの動作サンプル

システムが動的に出力する XML を指定することで、クライアントからみて大まかな範囲において動的なデータの探索が可能となり、次にサイト作成者が作成した XSLT によってそれらのデータが整形されて、クライアントは要求するデータを得ることが出来る。

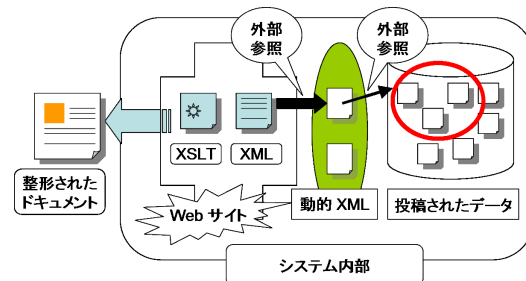


図 6: システム内部処理

クライアントの投稿は、timestamp を付与されてマークアップする。そのデータはシステム内部で時系列順に分割格納される。格納された投稿がどのファイルに格納されているのかといった情報をシステム内で保持しており、目的のデータをクライアントが探索する手順は、HTTP リクエスト内の URI の一部である QUERY_STRING 中に探索対象とする timestamp を付与し、その近傍のデータを本システムから探索し、その後クライアントサイドで、XSLT によって詳細探索およびドキュメントの整形

を行ってゆく。

6 今後の課題

本論文の執筆時点では未だ十分に検討されていないシステムの箇所がある。サーバ内部で保持されたクライアントからの投稿データである XML ファイルの管理と、トランザクションの処理である。これらについては今後とも十分に検討していかねばならない。また、本研究では従来の動的 Web サイトの全てを再現することは不可能である。というのは、Web アプリケーションによるスクリプトを利用しないため、本システムで実装された機能とクライアントサイドでの XML 処理機能しか利用することが出来ない。今後の課題としては、本研究で提案されたシステムでしか実現できない独自の機能を持つことと、先の不十分なシステムの箇所を詰めていくことである。また、このシステムについてクライアントから見た応答時間等を定量測定し、その性能を従来のシステムと比べて評価すると共に、その他の評価方法に関しても考察していきたい。

7 まとめ

動的 Web サイトの環境として、XML を用いることにより Web サーバとデータベースを集約した統合環境を提案した。本研究では、クライアントから投稿されたデータを XML として Web サーバ内にデータの処理機能を設けることで、従来のシステムでは必要だった Web アプリケーションおよびスクリプトを使用せずに、動的 Web サイトを実現できる環境を作成した。本研究は、例えばアンケートサイトのような、データを格納するのにスキーマが必要で、かつ低寿命である Web サイトに非常に有効だと思われる。今後は格納された XML データの処理やトランザクション処理について、さらに検討をすすめると共に、本システムの処理に関して測定を行い、従来のシステム等と性能を比較検討していきたい。

参考文献

- [1] Anindya Datta, Kaushik Dutta, Helen Thomas, Debra Vandermeer, Krithi Ramamritham: Proxy-Based Acceleration of Dynamically Generated Content on the World Wide Web: An Approach and implementation. In *ACM Transactions on Database Systems*, Vol.29, No.2, June, 2004, pp.403-443.
- [2] Filippo Ricca, Paolo Tonella: Using Clustering to Support the Migration from Static to Dynamic Web Pages. In *Proceedings of IWPC*, 2003, pp.207-216.
- [3] Lance Titchkosky, Martin Arlitt, Carey Williamson: A Performance Comparison of Dynamic Web Technologies. In *ACM SIGMETRICS*, Vol.31, Issue 3, December, 2003, pp.2-11.
- [4] Laurent Mignet, Denilson Barbosa, Pierangelo Veltri: Open hypermedia and the web: The XML web: a first study In *ACM Proceedings of the 12th international conference on World Wide Web*, May, 2003, pp.500-510.
- [5] Alexandros Labrinidis, Nick Roussopoulos: Generating dynamic content at database-backed web servers: cgi-bin vs. mod perl In *ACM SIGMOD* Vol.29, Issue 1, March, 2000, pp.26-31.