マルチメディア通信と分散処理 91－22
（1999. 1. 22）

WWWを拡張した利用者による自由な情報加工の実現

岡 誠, 知念 賢一, 砂原 秀樹
奈良先端科学技術大学院大学

概要

WWWにおいては，多種多様な情報の形式変換や翻訳などの情報加工を行うサーバが登場している．しかし，情報利用者がそのようなサーバを用いて情報の加工を行う際には，種々の制約が存在する．これは現在のWWWの枠組が情報加工を行うサーバの同時利用を想定していないためである．本稿では，現在のWWWの枠組を拡張した利用者による自由な情報加工が可能なWWWの枠組を提案する．この枠組では，複数の加工サーバの利用およびその加工結果の識別，そして加工結果の利用者への提示を実現している．

A User-side Information Representation Framework over the WWW

Makoto Oka, Ken-ichi Chinen, Hideki Sunahara
Nara Institute of Science and Technology

Abstract

On the WWW, many intermediate servers which convert information types or translate documents have been developed in recent years. However, when users use such servers, there are many restriction. Because it is not supposed using multiple intermediate servers at the same time in current WWW framework. In this paper, we propose a user-side information representation framework over the WWW by extending current framework. The framework allows that users use multiple intermediate servers simultaneously.

1 はじめに

現在World Wide Web(WWW)は，インターネットにおいて最も多く利用されている情報提供および情報取得手段である．

WWWにおいては，テキストの他に画像や音声あるいは動画といったマルチメディアデータを扱うことが可能である．こういった特徴は，多種多様な情報のやりとりが自由に行なえるという大きなメリットを情報提供者および利用者双方にもたらした．情報を提供しようとする者は，WWWを利用することによって，さまざまな表現形式を用いながら自分の意図に基づいて情報を自由に構成し，それを広くインターネット上で公開することが可能となった．また，構成した情報は自身の管理下におくことができるので，公開後も自由に変更できる．

しかし，このように情報を自身の意図に基づいて管理できる情報提供者に比べると，情報利用者は自

由度が低いと考えられる．それは利用者は基本的に提供された情報を選択することしかできないからである．すなわち，情報が自分の求める形式と異なる場合には，最も近いものを選ぶことしかできない．もちろん一旦情報を取得してから，それを加工するということは可能であるが，すべての形式に対応することは，利用者にとって大きな負担となる．また，情報を加工するサーバも存在するが，そのようなサーバを同時に複数利用することはできない．

こういった WWW の現状を踏まえ，本稿では WWW において利用者が自身の意図に基づいて自由に情報を加工するための枠組を提案する．この枠組においては，URI(Uniform Resource Identifier) における新たなスキームを利用することにより，複数の加工サーバを利用できる．そして既存のオリジンサーバ[1] は変更せずに利用することができる．

2 WWW における情報の加工

2.1 情報の加工の必要性と現状

WWW の普及により情報利用者は多種多様の情報を取得することが可能となった．しかし提供される情報の多様性が，情報を取得する際には逆に問題となる場合がある．すなわち，提供されている情報が利用者の環境で利用できない，もしくは利用が困難であるということである．特に携帯端末など資源が限られている場合には，画像等の利用できるフォーマットやサイズに制約があることが多い．

もちろん，提供者側で同一の内容の情報を異なる形式で用意している場合もあるが，すべての情報において施されているわけではない．したがって情報が利用者の必要とする形式で提供されていない場合には，利用者自身で形式の変換を行う必要がある．

最近では利用者のニーズに応じて，形式変換を行うサービスが登場している．たとえば，フォーマット変換 (画像，音声など)，プロトコル変換，コンテンツ変換などのサービスが提供されている．

また，形式の変換のみならず，アノテーション (注釈) あるいはバージョン管理など，元の情報に対して情報を付加したり，言語の翻訳を行うサービスも現れている．

[1] RFC1945, RFC2068 などで用いられている用語で，WWW サーバのうち情報を提供したり生成したりするサーバを指す．

本稿では，以上で述べたような形式変換やアノテーションなど，提供された情報になんらかの形で改変を加えることを情報の加工と呼ぶ．

こういった情報の加工は，基本的にクライアント側およびオリジンサーバ側のどちらで行うことも可能である．実際，画像ファイルなどは，データを一度取得してから変換のためのソフトウェアを用いて求める形式にすることが行なわれている．しかし個々の利用者が WWW で提供され得るすべての形式に対応することは，かなりの負担となると考えられる．一方，オリジンサーバ側で行う場合についても，世界中に存在するオリジンサーバすべてに加工機能を持たせるということは，現実的ではない．

そこで，オリジンサーバとクライアントの間で通信を中継するプロキシサーバ上で加工を行う手法が，現在では多く用いられている．この手法は，加工機能を複数のクライアントで共用することによって，加工結果を共有可能であるという利点を持つ．

加工機能を持った代表的なプロキシサーバとしては，DeleGate[2] が挙げられる．この DeleGate は様々なプロトコルの中継やキャッシングの機能の他に，フィルタリングや文字コードの変換といった情報加工機能を持つ．また情報の表現形式を変換する「トランスレータ」をプロキシサーバと独立して構成したシステムも試作されている [3]．

2.2 現在の WWW の枠組における情報加工の限界

これまでに述べたように，現在の WWW の枠組の中でも利用者による情報加工は行なわれている．

しかし現状では制約が非常に多く，また利用者にとっての負担が大きいと言える．特に複数の加工機能の選択や，再加工のための手段はほとんど提供されておらず，利用者が状況に応じて行っているのが現状である．

たとえば，プロキシサーバで加工を行う加工サーバを利用する場合には，クライアントの設定を変更して，すべての要求および応答が加工サーバを経由するようにしなければならない．さらに，通常クライアントには複数のプロキシサーバを同時に設定できないので，この方法では加工機能の選択や複数の加工サーバの同時利用は困難である．

現在の枠組は，オリジンサーバとクライアントの一対一での通信が前提となっており，プロキシサーバもその間に直列に配置される．情報の識別を行うURIのhttpスキームは，これに準じた体系になっている．HTTP 1.1 [4] において，httpスキームのURIは，次のような形式で規定されている．

```
http://host:port/path?query#fragment
```

この形式は，基本的にhostとportで示された単一のサーバに存在する情報を指定するためのものである．複数のオリジンサーバや加工サーバ，そして加工機能について記述することは，元来想定されていない．

query部分に加工サーバやそれに対する命令を示すことは不可能ではないが，いくつもの加工サーバを利用した場合に非常に複雑な記述となることが予想され，現実的ではないと考えられる．

さらに，原文と翻訳文の両方を同時に表示する場合などのように，複数の情報を利用者に提示することも困難である．

3 情報加工のための新たな枠組

2.2節で述べたように，WWWにおいて利用者が情報の加工を行う場合，現在の枠組では様々な制約が存在する．すなわち複数の加工サーバを同時に取り扱えないことや，それらを指定するための手段が欠如していること，そして複数の加工結果を同時に利用者に提示できないことである．

これらの制約は，現在の枠組が，オリジンサーバとクライアントの一対一での通信を前提としていることに起因していると考えられる．

本研究では，それらの制約を取り払って利用者による自由な情報加工を実現することを目的とする．

新たな枠組の設計にあたっては，複数の加工サーバの利用およびそれによる加工結果の識別，そして加工結果の利用者への提示の実現を目標とした．そして，広く普及したWWWの現状を考慮し，まったく新しい独自の枠組を構築するのではなく，従来の枠組を拡張することとした．特に世界中に多数存在するオリジンサーバの変更を必要としないということを目標とした．

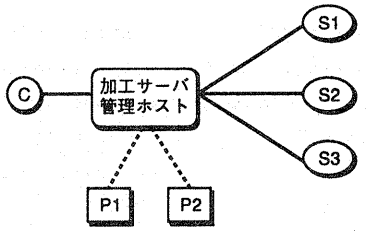

図 1: 集中管理方式

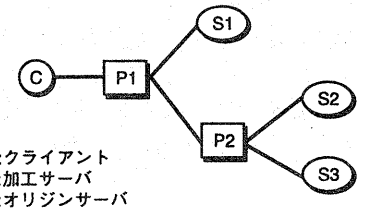

図 2: 分散処理方式

本節では，まず提案する枠組のモデルについて述べ，続いてそれに基づいて設計したURIにおける新たなスキームの設計について述べる．

3.1 モデル

今回構築した枠組は，基本的にHTTPを用いたクライアント–サーバモデルを踏襲している．このモデルにおいて，複数の加工サーバを同時に利用するためには，以下の2つの方式が考えられる．

- 集中管理方式 (図 1): 単一のホストが各加工サーバと通信を行ない，その結果をクライアントに返す．
- 分散処理方式 (図 2): 加工サーバがクライアントとオリジンサーバの間で処理を行う．

集中管理方式を採用しているシステムの例としては，林氏のシステム [3] が挙げられる．この方式は，加工サーバの制御については加工サーバを管理するホストが行なえばよいため，管理が比較的容易である．しかし，加工サーバ管理ホストが扱うことのできる加工サーバに限界があることや，管理ホスト

```
nhttp_URI      = "nhttp:" "//" 1*( "(" nhttp_internal ")" )
nhttp_internal = nhttp_elements | http_URL | HTTPで扱えるスキームのURL
nhttp_elements = command "," "(" sources ")" "," hostentry
http_URL       = "http:" "//" host [ ":" port ] [ abs_path ]
host           = <A legal Internet host domain name or IP address >
port           = *DIGIT
```

図 3: nhttp スキームのシンタックス

がダウンするとすべての加工機能が利用できなくなる，といった問題がある．

今回構築した枠組では分散処理方式を採用した．それは，インターネット上に散在している加工サーバを自由に選択可能であること，および負荷の分散が図れるという利点があるからである．

3.2 情報の識別

現在の URI における http スキームでは，単一のオリジンサーバのみを指定し，オリジンサーバとクライアントの間に位置するサーバを明示することができない．したがって前節で定義したモデルを記述するには，現在の http スキームでは不十分だと考えられる．そのため，このモデルに適合するような情報の識別方法が必要となる．

そこで今回，定義したモデルを実現するために必要な記述能力を備えた新たなスキーム"nhttp" を設計した．この nhttp スキームは，現在の WWW の枠組との共存を重視する基本方針に基づき，リソースの要求がクライアントから加工サーバを経由し，最終的にオリジンサーバに渡される際には，通常の http スキームの URI となるように設計を行なった．

nhttp スキームについては 4節で詳しく述べる．

3.3 プロトコル

新たな枠組においては，加工サーバの指定などは，すべて nhttp スキームに基づいた URI で表現できるため，HTTP に対する変更や加工サーバ間での独自のプロトコルは必要ない．

4 nhttp スキーム

4.1 概要

今回提案した枠組の中で設計した nhttp スキームにおける URI では，情報を提供するオリジンサーバの他に，加工サーバの名前や命令を明示的に示すことができる．そして加工サーバに対して，加工の対象となる情報とそれに適用する加工機能を指定可能である．その際，加工の対象となる情報として，他の加工サーバの処理結果やオリジンサーバの応答を複数入力することが可能である．

また複数の加工結果を並べて記述することも可能である．この場合にはクライアントでそれらを同時に表示することを意味する．

4.2 シンタックス

新たなスキーム"nhttp" のシンタックスの概要を図 3に示す．なお，ここでのシンタックスの記述には，RFC2068 で示されている拡張 BNF 記法を用いている．"(" と")" によって囲まれたエントリが，一つのオリジンサーバや加工サーバで提供される情報を示す．http スキームと同様に文字列の長さについては制限を設けない．

command, sources, hostentry の詳細を以下に示す．

4.2.1 command

```
command          = 1*nhttp_unreserved
nhttp_unreserved = ALPHA | DIGIT | safe |
                   national | "!" | "*" | "'"
```

これは加工サーバに対して，どの加工機能を利用するかを指示するためのものである．加工機能は加

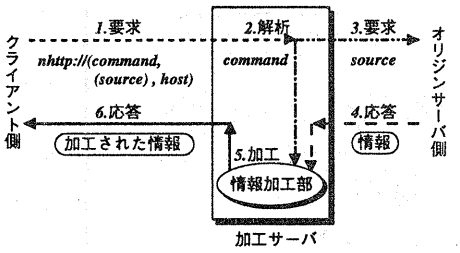

図 4: 加工サーバでの処理

工サーバに依存する. nhttp スキームでは"(", ")" および",","をデリミタとして用いているため, http スキームにおける unreserved 文字集合から除いている.

4.2.2 sources

```
sources = 1*( "(" nhttp_internal ")" )
```

ここでは加工の対象となる情報を指定する. 情報には nhttp および HTTP で扱えるスキームを指定可能である. "(" と")" とで囲むことにより, 複数の情報を指定できる.

4.2.3 hostentry

```
hostentry = host [ ":" port ]
```

ここには加工サーバを記述する. この記述は, http スキームで用いられている host および port の記述と同一である. port を省略した場合には HTTP のデフォルトポートである 80 番ポートとする.

5 新たな枠組における情報取得

まず, 加工サーバにおける処理の内容について述べ, 続いて複数の加工サーバを用いた情報取得の流れについて例に基づき説明する.

加工サーバにおける処理は URI の解析と, 情報の加工である. これを以下に示す (図 4).

1. クライアントは加工サーバに対して以下の要求を行う.

   ```
   GET nhttp://(command,(source),host)
   HTTP/1.1
   ```

2. 加工サーバでは, 要求中の URI を解析して command, source を取り出す. command は応答を加工する際に必要なため, 保存しておく.
3. 加工サーバは source で示されたオリジンサーバあるいは他の加工サーバに対して要求を行う.
4. 加工サーバに応答が到着する.
5. 受け取った情報に command で示される加工を行う.
6. 加工された情報をクライアント側に返す.

また, 複数の加工サーバを利用する場合について, 以下の URI を用いて情報取得の流れを説明する (図 5).

```
nhttp://(merge,(http://www.aaa.org/)
         (e-to-j,(http://www.bbb.com/),
          trans_proxy.bar.ac.jp),
         proxy.foo.co.jp)
```

1. クライアント : 加工サーバ proxy に, この URI を含んだ要求を行なう.
2. proxy : sources 部分を解析し, オリジンサーバ www.aaa.org および加工サーバ trans_proxy に接続する. 同時に merge を保存する.
3. trans_proxy :
   (a) proxy 同様の処理を行ない, オリジンサーバ www.bbb.com に接続する.
   (b) オリジンサーバから受け取った応答に加工 e-to-j を施す.
4. proxy : www.aaa.org および trans_proxy から応答が返され次第, merge を実行する.
5. クライアント : proxy からの応答を受け取り, 表示する.

6 実装

現在提案した枠組を実現するためには, オリジンサーバを変更する必要はない. しかし加工サーバについては, nhttp スキームの URI の解析および複数のサーバ (オリジンサーバもしくは加工サーバ) との

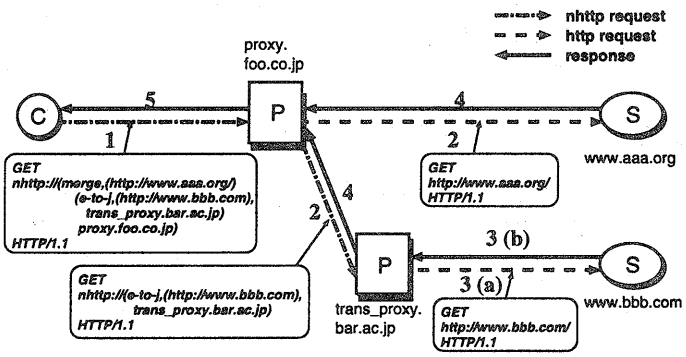

図 5: 複数の加工サーバを利用した情報の取得

同時接続を行なわなければならない．これらは，既存の加工サーバおよびクライアントが持つ機能の拡張であるので，実装にあたっては既存のソフトウェアを利用した．

加工サーバについては，われわれが独自に開発したプロキシサーバを改変した．また，一般的に利用されているプロキシサーバとして，現在 Apache HTTP Server[5] のプロキシモジュールについても，現在改変を行なっている．

クライアントについては，先に述べた機能の他に，利用者に対して複数の情報を提示する機能が必要である．Mozilla [6] を利用して開発に取り組んでいく予定である．なお，複数の加工サーバを利用する場合には URI の文字列が非常に長くなり，手で入力することが困難になることが予想される．したがって，URI の入力を支援するような機能がクライアントに求められると考えられる．

7 まとめ

本稿ではインターネットの利用者が，WWW において情報の加工を行うサーバを用いて自由に情報加工を行うために構築した枠組について述べた．この枠組では，加工した情報を指定するために今回設計した URI の新たなスキーム"nhttp" を利用している．このスキームは，情報加工を行うために必要なオリジンサーバおよび情報の加工を行うサーバのトポロジに即した記述が可能である．

また，オリジンサーバに対する要求は，既存の http スキームの URI を用いて行なわれるため，オリジンサーバを改変する必要はない．したがって，世界中に多数存在する従来のオリジンサーバを，引き続き利用できる．

今後，クライアントの改変を行ない，提案した枠組の新たな有効性について検証を行う予定である．

参考文献

[1] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC2396, August 1998

[2] 佐藤豊, "多目的プロトコル中継システム DeleGate", 電子技術総合研究所彙報第 59 巻第 6 号, <URL:ftp://etlport.etl.go.jp/pub/DeleGate/ETL-BULLTIN-95-06.ps.gz>, 1995

[3] 林 周志, "インターネット上での情報表現形式自動変換機構の実現", 慶応義塾大学大学院 政策・メディア研究科 修士論文, 1996

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1", RFC2068, January 1997

[5] Apache HTTP Server Project, <URL:http://www.apache.org/>

[6] Netscape Communications Corporation, "Mozilla", <URL:http://www.mozilla.org/>, 1998