CMS 利用によるデジタルアーカイブにおける時間一貫性の保証

2003MT045 栗木亮 2003MT047 黒田豊

指導教員 河野浩之

1 はじめに

デジタルアーカイブとは，文化遺産を記録精度が高く，映像再現性に優れたデジタル情報で保存し，次世代に継承していくことが目的の技術である．文化遺産を保存する上で，資料の破損や劣化の防止，時間的，地理的な制約を越えた資料の提供など多くの利点がある．

また，デジタルアーカイブには様々な課題がある．表 1 はデジタルアーカイブの課題についてまとめたものである．

表 1 デジタルアーカイブの課題

法制度	著作権	組織化	メタデータ
			時系列管理
	納本制度		識別子
対象の認識	情報発見	利用･提供	ナビゲーション
	粒度	蔵書管理	格納形式
	セレクション		原本性保証
収集	収集性能		資料の位置付け
	品質管理	保存	長期保存
	再収集	システム	ストレージ
	深層Web		ネットワーク
	有償･登録制コンテンツ		収集戦略

この中で，我々が注目し，本研究で扱うのは時系列の問題である．アーカイブコレクションは時系列で統一性のない表示や，時間の逆行が起こることがある．我々はこれらを解決し，資料の年代に沿った閲覧ができればユーザビリティの向上につながると考えた．

2 Webアーカイブと時間一貫性

2.1 Webアーカイブ

近年ではインターネットの普及により，Web上のデータは爆発的に増加しつつあるが，サーチエンジンを使ってWebページの検索などを行うと，必ず閲覧できないページが存在する．その原因は，ページが移動してURLが変わっている，ページ自体が削除されてしまい存在しない，などが挙げられる．そこで，Webページを定期的に収集し，それを文化遺産として後世に残していこうとする取り組みがWebアーカイブである[1].

収集されたアーカイブを参照するとき，時系列が正しくない表示がされることがある．特にその特徴が顕著なのはアメリカの Internet Archive である．Internet Archive は 1996 年からバルク収集で行われており，約 150TB という規模は世界最大級である．ここではアーカイブにアクセスするとき，アーカイブを参照したいアドレスにデータが存在しない場合，指定したアドレスに最も近い時間のデータを表示するようなアルゴリズムが用いられている．そのため，リンクを辿るごとに時間の逆行が起こることがある．例えば，ページ A から B へとページ内のリンクを使って移動するとき，ページ A が収集された時間のページ B が存在しないとき，ページ B はそのひとつ前に収集したデータにアクセスしてしまうという問題が生じる．アーカイブを管理するためにはこのような時系列の問題を解決しなければならない．そのために，時系列一貫性アルゴリズムが提案されている．

2.2 時系列一貫性アルゴリズム

同一 URL のデータを管理するとき，新しくコレクションされるアーカイブを異なるものとして蓄積し，管理･運用していくことは重要である．2.1 節で述べたように，Internet Archive 社のアーカイブコレクションは時間の逆行のような問題が起こり，時系列閲覧ができない状態である．そこで，アーカイブを行ったときのコレクションの集合を一貫性があると定義し，データ間の一貫性を閲覧対象期間を定めることで決定するアルゴリズムが提案されている[2].

2.3 閲覧可能なアーカイブコレクション

n 回目に収集されたページ i を $P_i(n)$とし，その時点から次にわかっている更新時刻までの間のページ i のデータである．T_U, T_A はそれぞれページ i がアップデートされた時間，ページ i がアーカイブされた時間である．図 1 において，$[T_s, T_e]$をユーザが閲覧を要求した閲覧対象期間とする．このとき，その期間内に存在したページ i のデータを返さなければならない．この期間で，確実にページ i がそのデータであったといえる期間は$[T_UP_i(n), T_AP_i(n)]$及び，$[T_UP_i(n+1), T_AP_i(n+1)]$である．この確実にそのデータであったと判断できる期間を Determinable な期間という．また，期間$[T_AP_i(n-1), T_UP_i(n)]$, $[T_AP_i(n), T_UP_i(n+1)]$及び，$[T_AP_i(n+1), T_UP_i(n+2)]$はその期間中に収集が行われておらず，アップデートされた可能性があり，確実にそのデータであったといえない．その期間を Indeterminable という．

このアルゴリズムでは，まず閲覧対象期間を定め，その期間に Determinable な期間が含まれているページを閲覧可能，そうでないものは閲覧できないようしている．つまり，図 1 の $P_f(n-1)$は閲覧対象期間に含まれているが，実際にそのデータを見ることはできず，Not Found が出力される．これは，$P_f(n-1)$の Determinable な期間が閲覧対象期間に含まれていないからであり，確実にそのデータであったといえないからである．逆に，見ることができるページは閲覧対象期間に Determinable な期間が含まれている $P_f(n)$及び，$P_f(n+1)$である．

また，閲覧対象期間中のどの時点でどのページが出力されるかであるが，期間$[T_s, T_UP_f(n)]$では，前述したように$P_f(n-1)$の Indeterminable な期間であり表示するページがない．よって Not Found である．期間$[T_UP_f(n), T_UP_f(n+1)]$では，$P_f(n)$の Determinable な期間が含まれているため，$P_f(n)$のデータを出力する．期間$[T_UP_f(n+1), T_e]$では，$P_f(n+1)$の Determinable な期間が含まれているため，$P_f(n+1)$のデータを出力する．

以上，ユーザが要求する閲覧対象期間内における時系列に沿ったデータ閲覧について述べた．

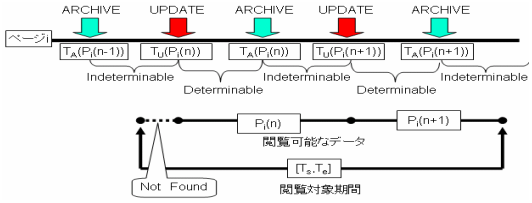

図 1 時系列一貫性アルゴリズム(文献[2]から引用)

3 CMS

Web コンテンツを構成するテキストや画像，レイアウト情報などを一元的に保存，管理し，サイトを構築したり編集したりするソフトウェアを CMS(Contents Management System)という．広義にはデジタルコンテンツの管理を行うシステムの総称である．Web サイトを構築するには，テキストや画像を作成するだけでなく，HTML や CSS などの言語でレイアウトや装飾を行ない，ページ間にハイパーリンクを設定するなどの作業も行う必要がある．これらの要素を分離してデータベースに保存し，サイト構築をソフトウェアで自動的に行うようにしたものが CMS である．

3.1 Nordic Web Archive

アーカイブを管理する CMS として NWA(Nordic Web Archive)がある[3]．NWA はWebアーカイブコレクションの管理ツールとして，北欧の国立図書館が共同で開発したものである．NWA はアーカイブを管理するために開発されたツールなので，その構造や実行環境を理解すれば，XOOPS などほかの CMS を用いた場合のアーカイブの時系列閲覧においても参考になると考え，我々は NWA を実際に動かした．

4 時間一貫性を保つ手法

先行研究で紹介したアルゴリズムを参考に，アーカイブのバージョンを読み込み，時間一貫性を保証する手法を提案する．

4.1 NWA の問題点

NWA は複数のアーカイブにバージョンがある場合は，時間軸上にポインタを表示し，閲覧可能なバージョンが確認できる．しかし，ページ間をリンクで移動した後などに時間軸上のポインタ以外の場所でページが表示される場合がある．これは，もし移動先などにページが存在しない場合はそれ以前のバージョンで一番近いものを表示するというアルゴリズムが NWA で用いられているためである．

この問題が起こる原因は NWA では基点となるページを作っていないからである．基点となるページとは，最初にアクセスしたページである．最初にアクセスしたページを基点とし，常にその時間を中心に移動できるような仕組みがあれば NWA でも時間一貫性を持った移動を行えるはずである．

4.2 時間一貫性を保証するアルゴリズム

時間一貫性を保証するアルゴリズムのフローチャートを図 2 に示し，その説明を示す．

① 閲覧対象期間を入力し時系列閲覧を開始する
② アーカイブのホスト名及び閲覧対象期間内のバージョンへのリンクを動的に生成する
③ 最初にアクセスしたページを基点とする
④ 基点より過去で閲覧対象期間内か判定する
⑤ 基点から過去のバージョンの目的のページを探す
⑥ 移動先が基点のページかどうか判定する
⑦ 基点のページを表示する
⑧ 探してきたページを表示する
⑨ 基点より未来で閲覧対象期間か判定する
⑩ 基点より未来のバージョンの目的のページを探す
⑪ 目的のページを表示する
⑫ 閲覧対象期間内に目的のページが無いことを表示する
⑬ 目的のページが存在する全てのバージョンのリンクを生成する
⑭ 目的のページが存在するバージョンを示す
⑮ 閲覧対象機関外の目的のページを表示する
⑯ アーカイブコレクションに目的のページが存在しないため終了する

我々はこのアルゴリズムを基に PHP によってプログラムを作成し，アーカイブコレクションを想定した環境で実験し，評価した．

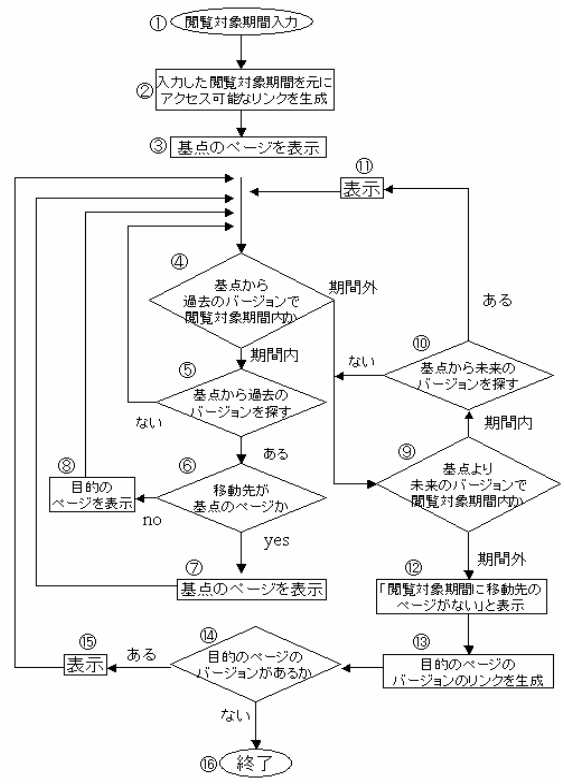

図 2 時間一貫性を保証するアルゴリズム

5 時系列閲覧プログラムの実装と評価

この章では，NWAで問題となっている時系列問題を解消するプログラム及び実行環境について説明する．

5.1 実験環境の構築

我々はヘッダを付けた簡単な html ファイルを用いて Web アーカイブ環境を構成し，そこでプログラムの実験を行った(図 3)．プログラムとアーカイブのディレクトリは同一のディレクトリで管理した．アーカイブのディレクトリには各ホストのディレクトリがあり，さらにそこでバージョン毎に管理されている．

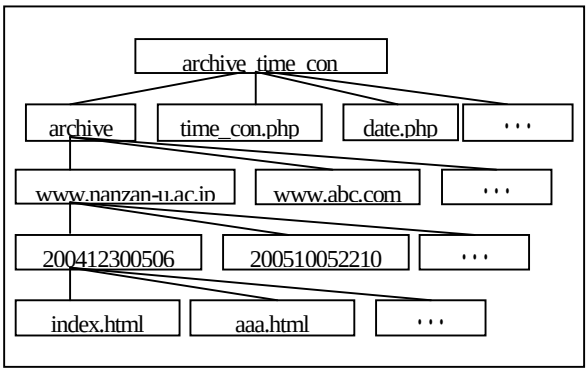

図 3 実験環境の構造

html ファイルは wget を使用して Web サイトを保存したときのヘッダと，各ファイルへのリンクで構成されている．index.html には同一ホストのファイルへのリンクとホスト間の移動のためのリンクを用意した．その他のファイルには同一ホストのファイルへのリンクを用意した．また，各リンクは time_con.php を経由して移動する．その際に URL 変数で目的のホスト名とファイル名を渡している(図 4)．

```
<a href="../../../time_con.php?host=www.abc.com&file=aaa.html>aaa</a>
<a href="../../../time_con.php?host=www.abc.com&file=bbb.html">bbb</a>
<a href="../../../time_con.php?host=www.abc.com&file=ccc.html">ccc</a>
<a href="../../../time_con.php?host=www.abc.com&file=ddd.html">ddd</a>
<a href="../../../time_con.php?host=www.abc.com&file=eee.html">eee</a>
```

図 4 URL 変数を用いたリンク例

5.2 時系列閲覧プログラムの解説

input.html， date.php， first_time.php， time_con.php，error.php という 5 つのプログラムを作成した．

- input.html

閲覧対象期間を入力し，date.php へ渡す役割をしている．ここで閲覧対象期間を入力することでアーカイブの参照を始めることができる．ここでは図 2 の①の処理を行っている．

- date.php

input.html から受け取った閲覧対象期間を元に，動的にアクセス可能なリンクを生成している．受け取った閲覧対象期間のデータを$_SESSION に格納することで他のファイルからの参照も可能にしている．ここでは図 2 の②の処理を行っている．

- first_time.php

date.php から受け取った URL 変数である最初にアクセスするアーカイブのバージョンの時間，URL 及びホスト名を $_SESSION に格納している．最初にアクセスしたアーカイブを基点とする移動を可能とするために行う処理である．格納後は URL 変数で受け取った URL へ移動する．ここでは図 2 の③の処理を行っている．

- time_con.php

アーカイブからリンクを使って移動するときに閲覧対象期間内であるかどうかを判定している．基点となるアーカイブのバージョンの時間に目的のページがない場合は，閲覧対象期間内の過去のページからページを探し，存在しなければ未来のページを見に行く．閲覧対象期間内にページが存在しなければ，error.php へ移動する．図 5 はページ移動時に閲覧対象期間内であるか判断しリンクを動的に生成する処理である．ここでは図 2 の④から⑪までの処理を行っている．

```
while($count > -2){
    if($ver_time[$a] >= $before && $ver_time[$a] <=  $after){
      $host2 = "archive/".$host."/".$ver_time[$a];
      exec("ls $host2",$look2);
      $count2 = 0;
      while($look2[$count2] != ""){
        if($look2[$count2] == $file){
          $fn = "archive/".$host."/".$ver_time[$a]."/".$file;
          $count = -2;
            if($file == 'index.html' && $host == $f_host){
            $fn = $f_url;}
          break;}
        $count2++; }
      }
    elseif($count == -1){
      $fn = "error";
      break; }
    $count--;
    $a--; }
```

図 5 URL を動的に生成するプログラム

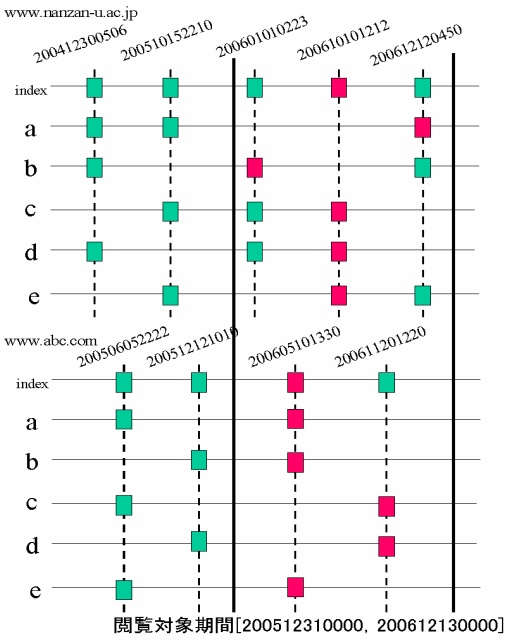

図 6 実行結果のイメージ

- error.php

time_con.php で閲覧対象期間内に目的のファイルがなかった場合，閲覧対象期間外の目的のファイルの一覧を示している．ここでは図 2 の⑫から⑯の処理を行っている．

5.3 時系列閲覧プログラムの評価

我々が実装したプログラムでは時間を 12 桁で読み込んでいる．例えば 2005 年 5 月 20 日 12 時 30 分なら 200505201230 としている．図 6 は閲覧対象期間を 200512310000 から 200612310000 と定めている．この間において基点を 200610101212 とした場合を説明する．この場合，ページ a で閲覧可能なページは基点のバージョンに a が存在しないため，200612120450 のページ a である．ページ b では 200601010223 と 200610101212 の二つのバージョン内で存在するが，基点となるページより過去のものが優先されるため，閲覧可能なページは 200601010223 のページである．ページ c，d，e では基点と同一のバージョンにページが存在するためそのページを閲覧することができる．また，index ページへ戻る場合は，どのバージョンにおいても基点のページへ戻ることが確認できた．

また，www.nanzan-u.ac.jp と www.abc.com のホスト間の移動について検証した(図 6)．この閲覧対象期間のとき www.nanzan-u.ac.jp の 200610101212 のバージョンにアクセスし，www.abc.com へ移動する場合，表示されるページは 200605101330 の index ページである．これは基点が www.nanzan-u.ac.jp の 200610101212 のバージョンだからであり，www.abc.com 内での移動も基点を中心に移動する．

また，閲覧対象期間を変更した場合も同様に，一貫性のあるアクセスを確認できた．

6 まとめ

本稿では Web アーカイブで起こる時間一貫性の問題を Internet Archive や NWA といった実働している Web アーカイブを例にして指摘し，その問題を解消できるプログラムを実装し検証した．その結果，単一ディレクトリ構造の Web アーカイブを想定した環境において閲覧対象期間内における時間一貫性を保証したアクセスを確認することができた．

今後の課題としては，単一ではなく複雑なディレクトリ構造を持つアーカイブへの URL 生成に対応する必要がある．また，外部からのアクセスに対するセキュリティが施されていないため，それも考慮しなくてはならない．そして実装したプログラムを実際に CMS へ組み込むことが課題である．

参考文献

[1] 廣瀬信己，“国立国会図書館における Web・アーカイビングの実践と課題，”情報処理学会研究報告 No.51，pp.95-111，2003.

[2] 小城正士，廣瀬信己，河野浩之，“Web アーカイブにおける時系列閲覧：単一コレクションへ適用，” DBSJ Letters，Vol.4，No.1，pp.153-156，2005.

[3] The NWA Toolset Manual, http://nwa.nb.no/docs/nwaToolsetManual.html (accessed 2006.8.11)