DEWS2005 6-A-04

Webアーカイブにおける時系列参照アルゴリズムの提案

小城　正士†　　廣瀬　信己††　　河野　浩之†††

† 京都大学大学院情報学研究科システム科学専攻　〒 606–8501 京都市左京区吉田本町
†† 国立国会図書館総務部　〒 100–8924 東京都千代田区永田町 1–10–1
††† 南山大学数理情報学部情報通信学科　〒 489–0863 愛知県瀬戸市せいれい町 27
E-mail: †kojo@sys.i.kyoto-u.ac.jp, ††nhirose@ndl.go.jp, †††kawano@it.nanzan-u.ac.jp

あらまし　近年，Web 情報の文化的・社会的価値に着目し，それらを保存する持続的な試みが世界各国で進められている．我が国では，国立国会図書館インターネット資源選択的蓄積実験事業 (WARP: Web ARchiving Project) において，Web アーカイブの構築が進められているが，データの収集，保存，運用において多くの技術的問題が存在する．本論文では，その中の収集データの時系列管理に焦点を当て，Web アーカイブコレクションを閲覧する際の時系列一貫性のあるリンク参照を，コレクションの収集期間と閲覧対象期間との関係から決定するアルゴリズムを提案する．提案した手法を用いることで既存の Web アーカイブの問題点を解決できることを示し，実際に Web から収集したデータを用いたシミュレーションから，閲覧可能性を向上できることを示す．
キーワード　ウェブアーカイブ，時系列管理，一貫性アルゴリズム

Browsing Algorithm with Time Consistency in Web Archive

Masashi KOJO†, Nobuki HIROSE††, and Hiroyuki KAWANO†††

† Graduate School of Informatics, Kyoto University　Yoshida Honmachi, Sakyou-ku, Kyoto, 606–8501 Japan
†† Accounts Division, National Diet Library　Nagatatyo 1–10–1, Chiyoda-ku, Tokyo, 100–8924 Japan
††† Department of Information and Telecommunication Engineering, Faculty of Mathematical Sciences and Information Engineering, Nanzan University　Seireityo 27, Setoshi, Aichi, 489–0863 Japan
E-mail: †kojo@sys.i.kyoto-u.ac.jp, ††nhirose@ndl.go.jp, †††kawano@it.nanzan-u.ac.jp

Abstract　In many countries, web archive projects have been promoted continually for preserving cultural and social properties on web systems. In Japan, a project called WARP (Web ARchiving Project) in the National Diet Library was promoted, it has many technical issues of archiving systems such as collection, preservation and management of web archive data. In this paper, we focus on the technical issues of the time series management of web archive collections, and propose an algorithm that identifies correct reference of link pages preserving the consistency while browsing web archive, depending on the relationship between the collection period and the browsing period. We show that the algorithm resolves the issues of existing web archives, and carry out simulation experiments by using real data collected from web.
Key words　web archive, time series management, consistency algorithm

1. はじめに

現在，インターネットの Web システム上に流通する情報量は，表層部分に 167TB 以上，深層部分に 91,850TB 以上存在すると推定されている [14]．しかし，知識流通基盤となるインターネット上の多様な情報は，従来の出版物に比べ，空間的にも時間的にも情報内容と存在が安定しないという問題点をもつ．

すなわち，情報内容の更新・改変が容易であるため，安定した原本保存が難しく（**内容の不安定性**），また仮に同じ内容であっても，URL (Uniform Resource Locator) が変更になることも多い（**存在の空間的不安定性**）．加えて，著者やサーバ管理者の都合により公開中止となる場合も頻繁に生じる．よって，数十年，数百年といった長期の視点で考えた場合，インターネット上の情報は，必ず消失すると言って過言でない（**存在の時間的不安定性**）．実際，Web ページの平均寿命は 44 日であると言われる [13]．そこで現在，Web 情報を国の文化資産として体系的に蓄積し，将来に渡って長期保存するウェブアーカイブ (Web Archive) プロジェクトが，世界各国の国立図書館等を中心に推進されている．プロジェクト遂行に関わる主要技術は Web データの「収集」「検索」「保存」に大別される．「収集」は，目的の

Web ページの内容の更新時期等に合わせて, Web ロボット等を用いて効率的に取得する技術である [16].「検索」は，収集ページの情報をデータベース等に格納して一貫性の高い管理を行う技術であり，例えば，全文検索機能等の検索支援機能が必要となる [6]．また「保存」は，収集データを利用性の高い形式で適切な記憶媒体に格納する技術である [15]．このうち「収集」と「検索」に関しては，分散協調型 Web ロボットや，Web マイニングをはじめ，サーチエンジン開発において多数の研究や技術開発が進められてきた [6], [16]．しかし,「保存」に関しては，単調増加する全ての Web ページを時系列順に長期間保存するため，ページ更新時に上書きを行えば良いサーチエンジンと異なるいくつかの機能が必要とされる．このうち「長期保存」「大容量保存」といった問題に関しては，階層型ストレージシステムを用いる手法を提案し，その評価を行った [8], [9].

そこで本稿では,「時系列管理」の問題に焦点を当て，収集したアーカイブコレクションを運用する際に，時系列一貫性を持った閲覧を行うための手法を提案する．通常のデータベースにおけるトランザクションの一貫性は，個々のイベントの発生時刻から決定される．提案するアルゴリズムにおいては，まず収集時に巡回できたデータの集合である個々のコレクションを一貫性のあるものとして捉え，その上で，各データ間での閲覧可能性を，個々のデータの更新時刻から決定する手法をとる．

まず，2 章では Web アーカイブの特徴と，サーチエンジンとの相違点を示し，諸外国及び我が国におけるプロジェクトとその問題点を紹介する．3 章では，収集したコレクションを基準に，時系列一貫性のある閲覧するためのアルゴリズムを定義する．4 章では，提案した手法を用いることで，既存のアーカイブの問題点が解決できることを示し，実際に収集したデータを用いたシミュレーションを行う．そして，結びと今後の課題を 5 章に述べる．

2. Web アーカイブの現状と問題点

Web アーカイブの現状と課題について説明する．

2.1 Web アーカイブの現状

Web アーカイブは,「バルク収集」と「選択的収集」の二つのアプローチに大別される．バルク収集とは一国全体，あるいは世界全体の Web 情報を一括して収集する方法であり，選択的収集とはサイト単位，あるいは資料単位でセレクションや著作権処理を行いながら収集していく方法である．諸外国において主にその国の国立図書館が主体となって多数のプロジェクトが推進されている [1]〜[3]．以下では，代表的な Web アーカイブにおける収集データの管理手法を紹介し，その問題点を示す.

2.1.1 WARP

WARP とは，我が国の国立国会図書館が推進する「国立国会図書館インターネット資源選択的蓄積実験事業 (WARP: Web ARchiving Project)」のことであり，2002 年 11 月よりその成果をインターネット上で公開している(注1)．電子雑誌，政府ウェブ，協力機関ウェブの三つのコレクションからなり，一件ずつ著作権等を処理しながら，選択的収集を行っている．2004 年 6 月 30 日現在，電子雑誌 1,108 タイトル，政府機関 10 タイトル，協力機関 598 タイトルを所蔵する．現在は小規模な実験事業ではあるが，法制度面について国立国会図書館長の諮問機関である納本制度審議会において立法措置が行われ，2006 年より対象を「jp」ドメインにまで広げた収集を開始する [5]．WARP では，著作権の譲渡を受けたサイトのみを，一定期間ごとに収集する．収集したデータは，収集日毎に一つのコレクションとして分類し，ソースファイルのリンク先アドレスをアーカイブ内のアドレスに書き換えることで，収集時点のサイトの状態を保っている．よって，収集日以外のサイトの状態は不明である．また，収集対象以外のドメインは収集しないので，他のサイトへのリンクは開けない.

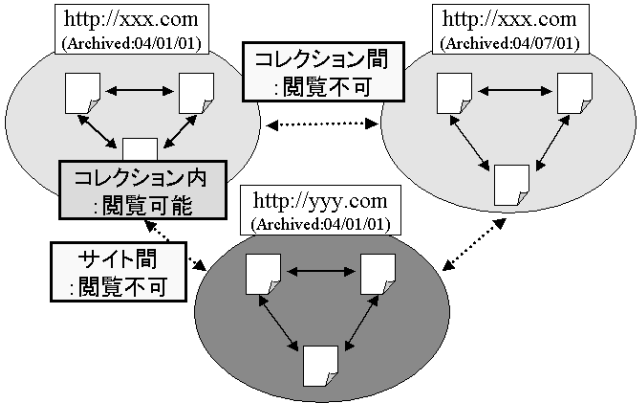

図 1　WARP の手法

この手法の場合，図 1 のように一つのコレクション内のリンク構造は再現されるが，そこから他のコレクションを閲覧出来ない．また，書き換えの困難なファイル (PDF，Flash 等) に対応出来ないことや，原本性保証の観点などからも問題が多い.

2.1.2 Internet Archive

現在のところ公開されている最大の Web アーカイブである，Internet Archive(注2) では，ページ内のリンクをクリックしたときに，Java Script によってリンク先のアドレスにアーカイブ内のアドレスを付加する手法をとっている．また，指定したアーカイブのアドレスにデータが存在しなかった場合，図 2(a) のように，その時点以前で最も近い時間のデータを表示する.

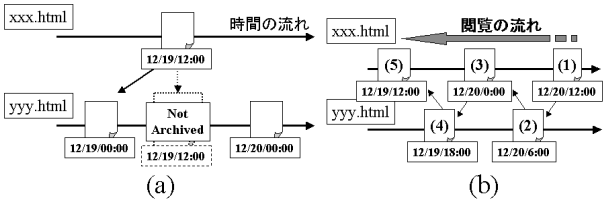

図 2　Internet Archive の手法

この手法では，指定したデータが存在しなかった場合に，意図したデータと全く違うデータが表示されることがある．また，

(注1)：http://warp.ndl.go.jp/

(注2)：http://www.archive.org/

ページ間のリンクの参照を繰り返すと，時間を逆行する場合がある．例えば，図 2(b) において，12/20/12:00 に収集された xxx.html(1) 内の yyy.html へのリンクを参照した時に，同時間に収集されたデータが存在しない場合，最も近い 12/20/6:00 の yyy.html(2) が開かれる．ここで，(1) に戻るつもりで xxx.html へのリンクを参照しても，(1) と (2) は同時間内にないために，(2) 以前で最も近い 12/20/0:00 の (3) が開かれてしまう．以後同様に，ページ間のリンク参照を繰り返すと (3) → (4) → (5) を時間の逆行が発生する．

2.1.3 NWA

NWA(Nordic Web Archive) は北欧各国の国立図書館が主体となって行われたプロジェクトであり(注3)，その成果として，Web アーカイブ管理運用ツール群である NWA Toolset が一般に公開されている [4]．NWA Toolset の Browser は，同一アドレスのデータのバージョンが複数ある場合に，図 3(a) のように，それぞれを時間軸上のポインタとして表示する機能を持っている．

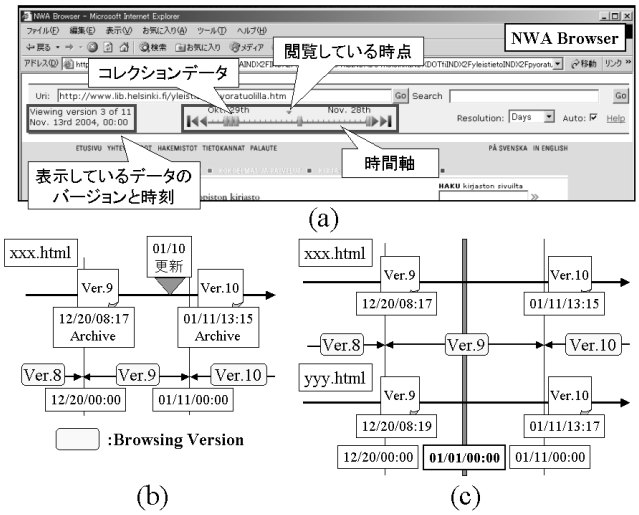

図 3　NWA の手法

ユーザーが時間軸上の一点を指定すると，指定した日時と，閲覧しているバージョンが示される．ユーザーが軸上のポインタ以外の時点を指定した場合には，その点より前で一番近いバージョンが表示される．バージョンの切り替えは日単位で行われ，バージョンが収集された日の前日の 23:59 までは前のバージョンを，0:00 以降は収集したバージョンを表示する．例えば図 3(b) において，ページ xxx.html は 12/22/08:17 に Ver. 9 が，01/11/13:15 に Ver. 10 が収集されているので，時間軸上で 12/22/00:00 から 01/11/23:59 までを指定した場合は Ver. 9 が，01/11/00:00 以降を指定した場合は Ver. 10 が表示される．また，ページ間のリンクを開く場合には，時間軸上の位置によって開くバージョンが異なる．閲覧している時間が，リンク先のページのバージョン間の中間点より前の場合，古い方のバージョンを，後の場合，新しい方のバージョンを開く．例えば図 3(c) において，xxx.html から yyy.html へのリンクを開く際に，閲覧している時間が，yyy.html の Ver. 9(12/22) と Ver. 10(01/11) の中間点である 01/01/00:00 より前の時は Ver. 9 を，それ以降の時は Ver. 10 を開く．図 3(c) について，時間軸上で指定した場合と，他のページからのリンクで開いた場合に表示されるページをまとめると表 1 のようになる．

表 1　表示される xxx.html のバージョン

時刻	–12/21/23:59	12/22/00:00 –12/31/23:59	01/01/00:00 –01/10/23:59	01/11/00:00–
時間軸で指定	Ver. 8	Ver. 9		Ver. 10
yyy.html からの参照	Ver. 8 (yyy.html) → Ver．8	Ver. 9 → Ver. 9	Ver. 9 → Ver. 10	Ver. 10 → Ver. 10

NWA Browser では，分岐点となる日時をバージョン間の中間点としているが，これはあまり一貫性のある閲覧ではない．また，時間軸上で指定する場合でも，データの更新が行われている (図 3(b) の 01/10) にも関わらず，それ以降も前のバージョンを表示するなど，実際の Web の状態とは異なる挙動をしており，再現性に問題がある．

2.2 Web アーカイブの抱える課題

現在のところ，Web アーカイブは多くの技術的課題を抱えており，その中でも「収集対象の多様性」「品質管理」「原本性保証」「時系列管理」「メタデータと識別子」「長期保存」といった課題は，Web アーカイブ固有のものであり，サーチエンジン等の既存技術では解決出来ない [9]．これらのうち，本稿では特に時系列管理に焦点を当てる．これは，同一 URL のデータが更新された時に，異なるバージョンとして蓄積，管理，運用することであり，アーカイブのコレクションが飛躍的に増大することを考えると，今後重要となってくる課題であるが，上で示した通り，既存の Web アーカイブにおける時系列管理には様々な問題点がある．既存の Web アーカイブの多くは，収集したコレクションを閲覧の基本単位として，その中での収集時の状態を再現するための手法をとっているが，実際にユーザーが閲覧を行う際には，収集日以外のサイトの状態や，サイト間のリンク関係に対する閲覧要求が出てくるものと思われる．その際に必要となるのは，コレクション内に限定しない，データ単位での時系列的に一貫性のある閲覧アルゴリズムである．通常のデータベース等におけるプロセスの一貫性の保証は，個々のイベントの発生時刻の順序から決定される [11], [12]．Web アーカイブにおいても，収集したコレクション内においては，個々のデータの更新時刻から，その時に閲覧しようとしていたと思われるデータを決定することが可能である．一方で，コレクション間においては，その時の実際の Web の状態が不明であるために，簡単には決定できない．そこでまず，既存の Web アーカイブと同様に，収集時に巡回できたデータの集合である個々のコレクションを，一貫性のあるものとして捉え，そこから個々のデータ間での一貫性を決定していく．また，一貫性を決定するパラメータの一つとして，閲覧対象期間を用いる．これは，ユーザーが閲覧を行う期間を明示的に指定したもので，これにより，その期間内でデータが存在したかどうか (データの閲覧可能性) を決定することができ，より再現性の高い閲覧を行う

(注3)：http://nwa.nb.no/

表 2　使用するパラメータ

Crawling		$T_A(P_i(n))$	: $P_i(n)$ が Archive された時刻
$[t_s(n), t_e(n)]$	: n 回目の収集実行期間	$T_U(P_i(n))$	: 収集時に取得された $P_i(n)$ の最終更新時刻
$t_s(n)$	: 開始時刻	**Browsing**	
$t_e(n)$	: 終了時刻	$[T_s, T_e]$	: 閲覧対象期間
$\boldsymbol{S_n}$	: 収集されたコレクション	T_s	: 起点
Archive Data		T_e	: 終点
i	: アドレス	$\langle P_i(n), P_j(n)\rangle$	: $P_i(n)$ から $P_j(n)$ への閲覧
$P_i(n)$	: n 回目の収集において保存された i のデータ. (存在しない場合は ϕ)	$\boldsymbol{S_B}$	: Archive の中の閲覧可能なデータの集合
		$\boldsymbol{L(S)}$	: 集合 $\boldsymbol{S}$ における一貫性のある閲覧の集合

ことが出来る.

3. 時系列一貫性アルゴリズム

本稿で用いるパラメータを表 2 に示す. コレクション $\boldsymbol{S_n}$ は, n 回目の収集で集められたデータがアーカイブされた時刻順に並んでいるものとする.

$$\boldsymbol{S_n} = \{P_0(n) \cdots P_i(n) P_j(n) \cdots\}$$
$$T_A(P_0(n)) \leqq \cdots \leqq T_A(P_i(n)) \leqq T_A(P_j(n)) \leqq \cdots \quad (1)$$

$\boldsymbol{S_B}$ はアーカイブされたデータの内, 閲覧対象期間 $[T_s, T_e]$ 内において閲覧可能なデータの集合を表している. これに含まれるデータは, 期間内において, 単独では全て閲覧可能であるが, データ間のリンクによる閲覧に関しては, $[T_s, T_e]$ の状態によって異なる. データ $P_i(n)$ からデータ $P_j(m)$ への閲覧が可能であるとき, その状態を $\langle P_i(n), P_j(m)\rangle$ で表し, 相互に閲覧が可能である状態を $\langle\langle P_i(n), P_j(m)\rangle\rangle$($\langle P_i(n), P_j(m)\rangle$ かつ $\langle P_j(m), P_i(n)\rangle$) で表す. また, 集合 $\boldsymbol{S_B}$ における可能な閲覧状態の集合を $\boldsymbol{L(S_B)}$ で表す.

以下ではまず, 収集したコレクション $\boldsymbol{S_n}$ についての一貫性のある閲覧を定義する. その後に, 閲覧対象期間内で閲覧可能なデータの集合 $\boldsymbol{S_B}$ を定義し, そこから一貫性のある閲覧状態の集合 $\boldsymbol{L(S_B)}$ を定義する.

3.1 収集したコレクション内での一貫性のある参照

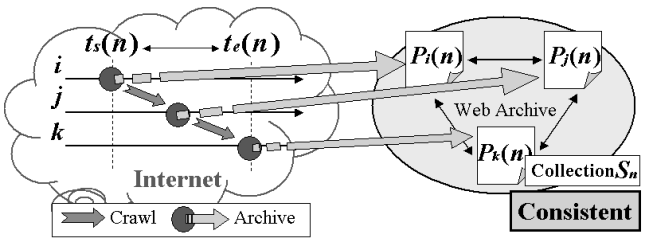

図 4　コレクション内での一貫性

Web 上のページは時々刻々と更新されており, ある時間内で意図していた参照と別の時間内で意図していた参照は, 内容が更新されている場合には異なるものと考えなければならない. しかし, その時点での Web の状態がどうであったかは不明で, アーカイブしたデータから再現することは出来ない. 一方, Crawler によって収集されたコレクションは, 少なくとも Crawler が収集した時には存在していた状態であるので, そのリンク構造は収集時間内においては一貫性があると言える. よって, 図 4 のように Crawler によって収集されたコレクション $\boldsymbol{S_n}$ を, その収集が行われた時間内 $[t_s(n), t_e(n)]$ において一貫性がある (Consistent) とし, $\boldsymbol{S_n}$ 内のデータ間の閲覧状態は全て正しい (閲覧可能) とする.

$$\boldsymbol{L(S_n)} = \{\langle\langle P_i(n), P_j(n)\rangle\rangle\}, (\forall i, j\, P_i(n), P_j(n) \in \boldsymbol{S}_n) \quad (2)$$

このコレクションにおける一貫性のある閲覧状態の集合を基準に, 個々のデータ間での閲覧の一貫性を, 閲覧対象期間 $[T_s, T_e]$ との関係から定義する.

3.2 閲覧可能なデータ

閲覧対象期間 $[\,T_s, T_e\,]$ を指定した場合に, Browser 上で閲覧されるデータは, その期間内で存在した可能性があるデータでなければならない. アーカイブされた $P_i(n)$ が, その時点での i のデータであったと確実に言える期間は, そのデータの最終更新時刻から収集時刻まで, つまり期間 $[\,T_U(P_i(n)), T_A(P_i(n))\,]$ である. この期間を *"Determinable"* な期間と呼ぶ. $T_A(P_i(n))$ から次の $T_U(P_i(n+1))$ までの間は, 収集が行われておらず, 更新の有無が不明であるために, その間の i のデータは $P_i(n)$ であるとは言い切れない. この期間を *"Indeterminable"* な期間と呼ぶ. この期間のうち, 少なくとも一部 (もしくは全部) の期間においては, i のデータは $P_i(n)$ であったと思われる. よって前述の議論で, 収集時に収集したコレクションを一貫性があるとしたのと同様に, Crawler が収集を行った時点を閲覧対象期間に含んでいる場合には, その時点から次に分かっている更新時刻までの間の i のデータを, $P_i(n)$ とする. つまり, データ $P_i(n)$ の *Determinable* な期間 $[\,T_U(P_i(n)), T_A(P_i(n))\,]$ が $[\,T_s, T_e\,]$ に含まれるとき, その後にくる *Indeterminable* な期間 $[\,T_U(P_i(n)), T_U(P_i(n+1)))$ においても, $P_i(n)$ を閲覧可能とする. また, 閲覧可能なデータが存在しない期間は, その期間内において i のデータはアーカイブされていないものとして, 閲覧不可とし, i への閲覧は `Not Found` を返すとする.

例えば, 図 5 では, $P_i(n), P_i(n+1)$ の *Determinable* な期間が $[T_s, T_e]$ に含まれているので, それぞれ期間 $[T_U(P_i(n)), T_U(P_i(n+1)))$,$[T_U(P_i(n+1)), T_e]$ において閲覧可能としているが, $P_i(n-1)$ は *Determinable* な期間が含まれていないので閲覧不可であり, 期間 $[T_s, T_U(P_i(n)))$ において i は `Not Found` である.

以上から, 閲覧対象期間内での個々のデータの閲覧可能性を定義した. しかし, 図 5 のように, $[T_s, T_e]$ を広く取った場合,

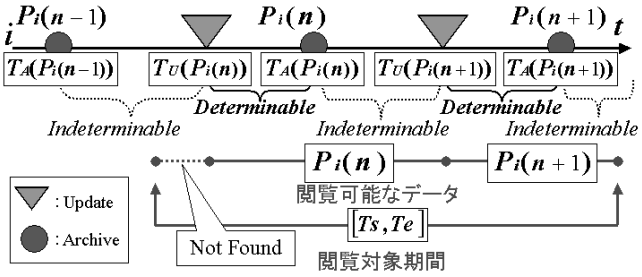

図 5　閲覧可能なデータ

一つのアドレスについて閲覧可能なデータが複数存在することになる．そこで次では，複数ある閲覧可能データ中から参照するデータを自動的に決定するために，データ間での一貫性のある閲覧を定義する．

3.3　データ間での一貫性のある閲覧

$[T_s, T_e]$ 内でデータ $P_i(n)$ から j へのリンクをクリックした時に (簡単化のため，全てのデータ間で相互リンクが存在するとする)，データ $P_j(m)$ を閲覧するのが妥当であるならば，閲覧可能状態の集合 $\boldsymbol{L(S_B)}$ に状態 $\langle P_i(n), P_j(m)\rangle$ を含める．これ以降は様々な状態における $\boldsymbol{L(S_B)}$ について議論する．まず始めに単一のコレクションに収まる場合を，その後に複数のコレクションにまたがる場合を議論する．

3.4　単一のコレクションに収まる場合

単一のコレクションに収まる場合とは，$[T_s, T_e]$ が $n-1, n+1$ 回目の全てのコレクションデータの *Determinable* な期間を含まず，全ての $P_i(n-1), P_i(n+1)$ が閲覧不可な状態を表す．$P_i(n)$ の閲覧可能性は $[T_s, T_e]$ の状態によって異なる．

まず，相互リンクを持つ二つのデータ $P_i(n)$ と $P_j(n)$ 間のリンクについて考えるが，その際に注目すべきなのは，各データの更新時刻 $T_U(P_i(n)), T_U(P_j(n))$(常に $T_U(P_j(n)) < T_U(P_i(n))$ であるとする) と $[T_s, T_e]$ の関係である．それぞれの間で成り立つ関係は図 6 で表される．

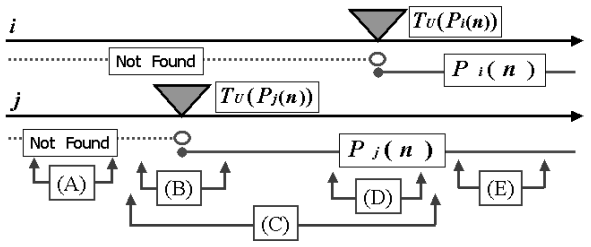

図 6　閲覧対象期間と最終更新時刻の状態

Case A. $\boldsymbol{T_s < T_e < T_U(P_j(n)) < T_U(P_i(n))}$
閲覧可能なデータがないので，i, j 共に `Not Found`.

$$\boldsymbol{S_B} = \{\phi\} \tag{3}$$

Case B. $\boldsymbol{T_s \leqq T_U(P_j(n)) \leqq T_e < T_U(P_i(n))}$
$P_j(n)$ は閲覧可能だが，i の閲覧可能なデータがないため，$P_j(n)$ から i への閲覧は `Not Found`.

$$\boldsymbol{S_B} = \{P_j(n)\}, \boldsymbol{L(S_B)} = \{\phi\} \tag{4}$$

Case C. $\boldsymbol{T_s < T_U(P_j(n)) < T_U(P_i(n)) \leqq T_e}$
i と j 共に $P_i(n), P_j(n)$ が閲覧可能である．また，i の更新が行われているので，$P_j(n)$ の意図していた参照と異なっている可能性があるが，i のそれ以前のデータが閲覧不可であるので，$P_i(n)$ を閲覧するのが最も妥当である．

$$\boldsymbol{S_B} = \{P_i(n), P_j(n)\}, \boldsymbol{L(S_B)} = \{\langle\langle P_i(n), P_j(n)\rangle\rangle\} \tag{5}$$

Case D. $\boldsymbol{T_U(P_j(n)) \leqq T_s < T_U(P_i(n)) \leqq T_e}$
Case C. と同様．

$$\boldsymbol{S_B} = \{P_i(n), P_j(n)\}, \boldsymbol{L(S_B)} = \{\langle\langle P_i(n), P_j(n)\rangle\rangle\} \tag{6}$$

Case E. $\boldsymbol{T_U(P_j(n)) < T_U(P_i(n)) \leqq T_s < T_e}$
$P_i(n), P_j(n)$ 共に閲覧可能で，更新も行われていないので，相互に閲覧可能である．

$$\boldsymbol{S_B} = \{P_i(n), P_j(n)\}, \boldsymbol{L(S_B)} = \{\langle\langle P_i(n), P_j(n)\rangle\rangle\} \tag{7}$$

3.5　複数のコレクションにまたがる場合

$[T_s, T_e]$ が複数のコレクションをまたぐ場合，ある i について閲覧可能なデータが複数あることになる．それぞれのデータの閲覧可能状態ごとに場合分けして議論する．

3.5.1　$P_i(n-1)$ は閲覧可能，$P_j(n-1)$ は閲覧不可の場合

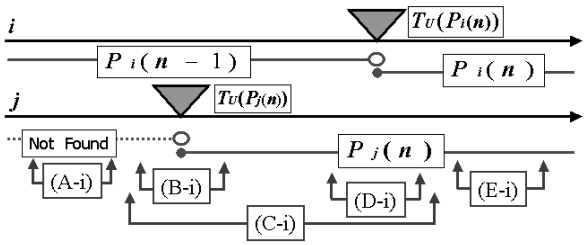

図 7　$P_i(n-1)$ は閲覧可能，$P_j(n-1)$ は閲覧不可

Case A-i. 閲覧可能なデータは $P_i(n-1)$ のみなので，j への閲覧は `Not Found`.

$$\boldsymbol{S_B} = \{P_i(n-1)\}, \boldsymbol{L(S_B)} = \{\phi\} \tag{8}$$

Case B-i. 閲覧可能なデータは $P_i(n-1), P_j(n)$．j が更新されているが，$P_j(n-1)$ は閲覧不可なため，$P_i(n-1)$ からは $P_j(n)$ を閲覧する．

$$\begin{aligned}\boldsymbol{S_B} &= \{P_i(n-1), P_j(n)\},\\ \boldsymbol{L(S_B)} &= \{\langle\langle P_i(n-1), P_j(n)\rangle\rangle\}\end{aligned} \tag{9}$$

Case C-i. 閲覧可能なデータは $P_i(n-1), P_i(n), P_j(n)$．j が更新されているが $P_j(n-1)$ は閲覧不可なので，$P_i(n-1)$ からは $P_j(n)$ を閲覧する．

$$\begin{aligned}\boldsymbol{S_B} &= \{P_i(n-1), P_i(n), P_j(n)\},\\ \boldsymbol{L(S_B)} &= \{\langle\langle P_i(n-1), P_j(n)\rangle\rangle, \langle P_i(n), P_j(n)\rangle\}\end{aligned} \tag{10}$$

Case D-i. 閲覧可能なデータは $P_i(n-1), P_i(n), P_j(n)$．i が更新されているので，$P_j(n)$ からは $P_i(n-1)$ を閲覧する．

$$\begin{aligned}\boldsymbol{S_B} &= \{P_i(n-1), P_i(n), P_j(n)\},\\ \boldsymbol{L(S_B)} &= \{\langle\langle P_i(n-1), P_j(n)\rangle\rangle, \langle P_i(n), P_j(n)\rangle\}\end{aligned} \tag{11}$$

Case E-i. 閲覧可能なデータは $P_i(n), P_j(n)$ で，相互に閲覧可能である．

$$\boldsymbol{S_B} = \{P_i(n), P_j(n)\}, \boldsymbol{L(S_B)} = \{\langle\langle P_i(n), P_j(n)\rangle\rangle\} \quad (12)$$

3.5.2 $P_i(n-1)$ は閲覧不可，$P_j(n-1)$ は閲覧可能

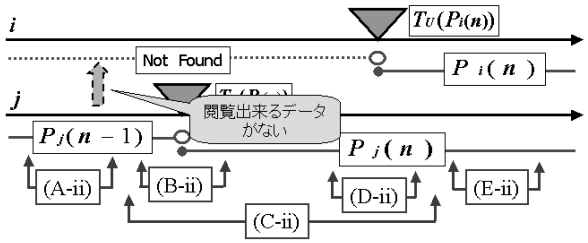

図 8　$P_i(n-1)$ は閲覧不可，$P_j(n-1)$ は閲覧可能

Case A-ii. 閲覧可能なデータは $P_j(n-1)$ のみなので，i への閲覧は `Not Found`.

$$\boldsymbol{S_B} = \{P_j(n-1)\}, \boldsymbol{L(S_B)} = \{\phi\} \quad (13)$$

Case B-ii. 閲覧可能なデータは $P_j(n-1), P_j(n)$．閲覧可能な i のデータがないので `Not Found`.

$$\boldsymbol{S_B} = \{P_j(n-1), P_j(n)\}, \boldsymbol{L(S_B)} = \{\phi\} \quad (14)$$

Case C-ii. 閲覧可能なデータは $P_i(n), P_j(n-1), P_j(n)$．図 8 のように $P_j(n-1)$ と $P_i(n)$ の存在した期間で重なるところがなく，$P_i(n-1)$ が閲覧不可であることから，$P_j(n-1)$ から i へは閲覧できるデータがないことになり，`Not Found` になる．

$$\begin{aligned}\boldsymbol{S_B} &= \{P_i(n), P_j(n-1), P_j(n)\},\\ \boldsymbol{L(S_B)} &= \{\langle\langle P_i(n), P_j(n)\rangle\rangle\}\end{aligned} \quad (15)$$

Case D-ii. 閲覧可能なデータは $P_i(n), P_j(n)$．i が更新されているが $P_i(n-1)$ は閲覧不可なので，$P_j(n)$ からは $P_i(n)$ を閲覧する．

$$\boldsymbol{S_B} = \{P_i(n), P_j(n)\}, \boldsymbol{L(S_B)} = \{\langle\langle P_i(n), P_j(n)\rangle\rangle\} \quad (16)$$

Case E-ii. 閲覧可能なデータは $P_i(n), P_j(n)$ で，相互に閲覧可能である．

$$\boldsymbol{S_B} = \{P_i(n), P_j(n)\}, \boldsymbol{L(S_B)} = \{\langle\langle P_i(n), P_j(n)\rangle\rangle\} \quad (17)$$

3.5.3 $P_i(n-1)$，$P_j(n-1)$ 共に閲覧可能

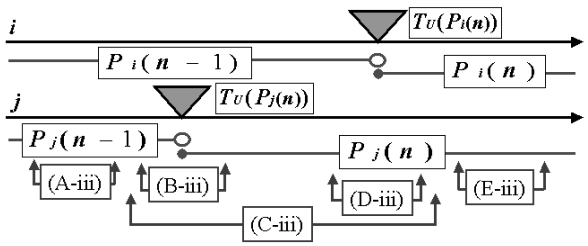

図 9　$P_i(n-1)$，$P_j(n-1)$ 共に閲覧可能

Case A-iii. 閲覧可能なデータは $P_i(n-1), P_j(n-1)$．更新がないので，相互に閲覧可能．

$$\begin{aligned}\boldsymbol{S_B} &= \{P_i(n-1), P_j(n-1)\},\\ \boldsymbol{L(S_B)} &= \{\langle\langle P_i(n-1), P_j(n-1)\rangle\rangle\}\end{aligned} \quad (18)$$

Case B-iii. 閲覧可能なデータは $P_i(n-1), P_j(n-1), P_j(n)$．$j$ が更新されているために，$P_i(n-1)$ が意図していた閲覧と異なっている可能性がある．よって $P_i(n-1)$ からは $P_j(n-1)$ を閲覧する．

$$\begin{aligned}\boldsymbol{S_B} &= \{P_i(n-1), P_i(n-1), P_j(n)\},\\ \boldsymbol{L(S_B)} &= \left\{\begin{array}{c}\langle\langle P_i(n-1), P_j(n-1)\rangle\rangle,\\ \langle P_j(n), P_i(n-1)\rangle\end{array}\right\}\end{aligned} \quad (19)$$

Case C-iii. 閲覧可能なデータは $P_i(n-1), P_i(n), P_j(n-1), P_j(n)$．$i, j$ 共に更新されているので，それぞれの閲覧するデータが異なる．

$$\begin{aligned}\boldsymbol{S_B} &= \{P_i(n-1), P_i(n), P_j(n-1), P_j(n)\},\\ \boldsymbol{L(S_B)} &= \left\{\begin{array}{c}\langle\langle P_i(n-1), P_j(n-1)\rangle\rangle,\\ \langle P_j(n), P_i(n-1)\rangle, \langle P_i(n), P_j(n)\rangle\end{array}\right\}\end{aligned} \quad (20)$$

Case D-iii. 閲覧可能なデータは $P_i(n-1), P_i(n), P_j(n)$．i が更新されているので，$P_j(n)$ からは $P_i(n-1)$ を閲覧する．

$$\begin{aligned}\boldsymbol{S_B} &= \{P_i(n-1), P_i(n), P_j(n)\},\\ \boldsymbol{L(S_B)} &= \left\{\begin{array}{c}\langle\langle P_i(n-1), P_j(n)\rangle\rangle,\\ \langle P_i(n), P_j(n)\rangle\end{array}\right\}\end{aligned} \quad (21)$$

Case E-iii. 閲覧可能なデータは $P_i(n), P_j(n)$ で，相互に閲覧可能．

$$\boldsymbol{S_B} = \{P_i(n), P_j(n)\}, \boldsymbol{L(S_B)} = \{\langle\langle P_i(n), P_j(n)\rangle\rangle\} \quad (22)$$

複数のコレクションをまたぐ場合の定義は以上である．またぐコレクションの数が三つ以上になっても同様で，各データの更新時間と存在する期間との関係から，妥当と思われる参照を決定する．

4. 提案したアルゴリズムの性能評価

提案した手法を実際に適用した場合の考察を行う．

4.1 既存の Web アーカイブへの適用

4.1.1 WARP への適用

提案した手法は，閲覧対象期間内であれば，サイト内，サイト間に関わらず一貫性のある参照ができ，閲覧対象期間を収集日以外にとっても，各ページの最終更新日から，一貫性のある参照を決定することが出来る．また，この手法は元データに手を加える必要がないので，変更の難しいフォーマットにも適用出来る．

4.1.2 Internet Archive への適用

Internet Archive においては，参照するデータが同時間内にない場合に時間軸を逆行する問題があった．提案した手法では，T_s 以前に収集されたデータは参照しないので，逆行は発生しない．閲覧対象期間を長くとった場合に，その期間内においては逆行が発生することがあるが，必ず T_s 付近で停止する．また，

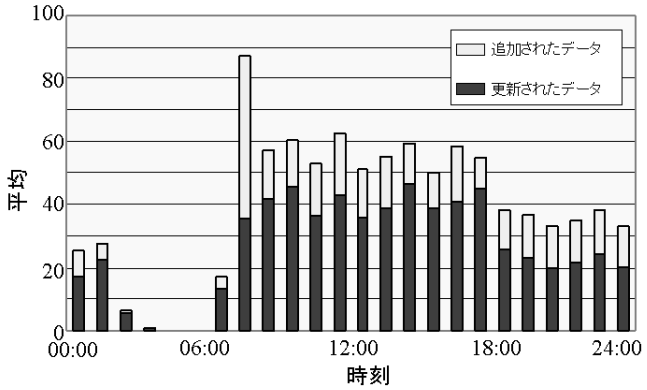

図 10　時間毎の追加されたデータ数と更新されたデータ数

収集を行っている間に更新されるなどして，意図したものとは異なるデータを開いてしまう問題に関しては，この手法では解決出来ない．この問題を解決するためには，収集経路を最適化したり，更新頻度の高いページは頻繁に収集したりするなどの収集戦略を考える必要がある．

4.1.3　NWA への適用

NWA における問題点は，あるアドレスにおけるデータのバージョンの切り替えが，データの更新を考慮していない点と，他のアドレスへのリンクを閲覧する際に，表示するバージョンを決定する手法の一貫性が低い点であった．例えば，図 3 のような状態においては，表 1 のように，閲覧方法によって表示されるバージョンが異なっている．これに対して，提案した手法では，リンク先のデータの更新時刻が，現在閲覧しているデータの更新時刻より前か後かで表示するバージョンが異なる．更新時刻が前の場合と，後の場合に，閲覧するバージョンを示したのが表 3 である．これを見ると，個々のアドレスについて閲覧するバージョンは，データの更新に合わせて切り替わっていることが分かる．また，他のアドレスへのリンクに対しては，それぞれのデータの更新時刻から，その時にリンク先として意図していたであろうバージョンを閲覧していることが分かる．

表 3　表示する xxx.html のバージョン

xxx.html の更新時刻が yyy.html の更新時刻より前の場合			
時刻	–01/10/07:59	01/10/08:00 –01/10/11:59	01/10/12:00–
時間軸で指定	Ver. 9		Ver. 10
yyy.html からの参照	Ver. 9 → Ver. 9	Ver. 10 → Ver. 9	
xxx.html の更新時刻が yyy.html の更新時刻より後の場合			
時刻	–01/10/11:59	01/10/12:00 –01/10/17:59	01/10/18:00–
時間軸で指定	Ver. 9	Ver. 10	
yyy.html からの参照	Ver. 9 → Ver. 9		Ver. 10 → Ver. 10

4.2　実データを用いたシミュレーション

実際のアーカイブデータに提案した手法を用いた場合のシミュレーションを行う．まず，Crawler を用いて，日本経済新聞社 (http://www.nikkei.co.jp/) のサイトの全てのデータを，一時間ごとに収集し，アドレス，収集日時，更新の有無，更新日時をログとして記録した．このログを用いて，提案したアルゴリズムを適用した場合の一貫性の向上を調べる．

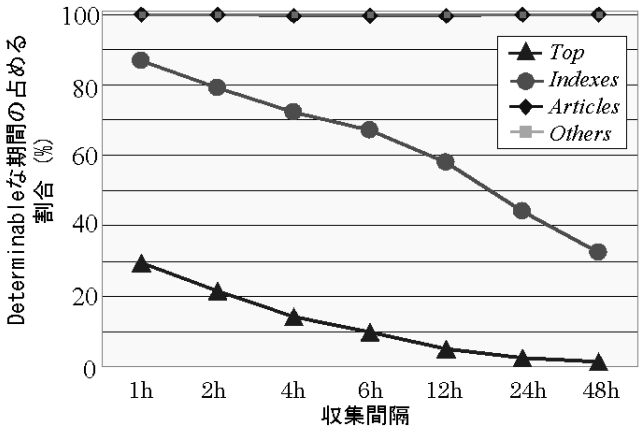

図 11　*Determinable* な期間の占める割合の変化

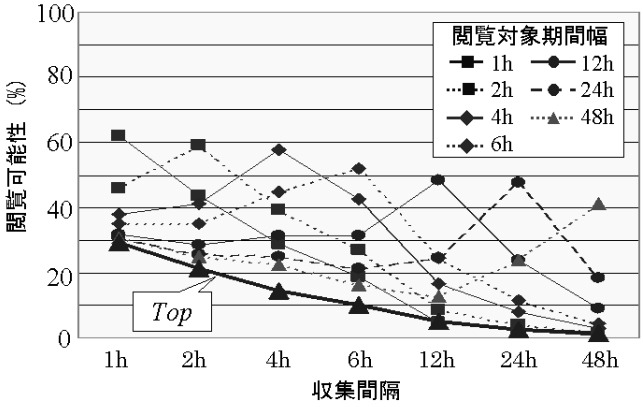

図 12　閲覧対象期間を変化させた時の，閲覧可能性の変化

一日における各サイトの新規追加ページ数，更新ページ数の時間ごとの平均を求めたものが図 10 である．サイトの総ファイル数の平均は，6352.5 である．これを見ると，ページの追加，更新に特徴があるのが分かる．こういったサイトの更新の特徴は，収集戦略を決めるのに役立つと思われる．

アーカイブ内において，あるアドレスのデータが存在した時間は，そのアドレスを最初に巡回した時に取得したデータの最終更新時間から，最後にそのアドレスのデータを収集した時間までの間である．この時間のうち，*Determinable* である期間の占める割合はどれくらいになるだろうか．ここでいう *Determinable* である期間とは，収集時に取得した最終更新時間から収集完了までの時間の総和を指す．この割合は，そのアドレスのアーカイブ内での閲覧可能性を示しており，更新が頻繁に行われるページでは低く，一度も更新が行われていないページでは 100%になる．また，データの種類，性質によっても異なる．そこで，収集したデータを，その性質から，*Top*，*Indexes*，*Articles*，*Others* に分類する．*Top* とは，そのサイトの一番初めに開かれるページであり，*Indexes* とは，各ディレクトリのインデックスページのことである．これらは頻繁に更新が行われているデータである．*Articles* とは末端の記事ページなど，一度

アップロードされたら更新がほとんど行われないページのことである．*Others* とは，HTML 以外の，画像（jpg,gif など），文書（pdf,doc など）等のデータを指す．これらも更新はほとんど行われない．図 11 は，あるアドレスのデータが存在した時間のうち，*Determinable* な期間の占める割合の，収集間隔を変えた場合の変化を，データの種類ごとに求めたものである．これをみると，*Articles*, *Others* はほとんど更新されないために，ほぼ 100%となっている一方，更新が頻繁に行われる *Top*, *Indexes* は収集間隔が広がるにつれて割合が低くなっていることが分かる．特に，サイト中最も更新が頻繁に行われる *Top* は，その落ち込みが激しくなっている．

これらのデータに対して，提案した手法を適用した場合のシミュレーションを行う．まず，収集したデータのうち，最も更新が頻繁に行われる *Top* のデータを用いる．このデータに，提案したアルゴリズムを適用して，閲覧対象期間内における閲覧可能なデータを決定し，閲覧可能な期間が閲覧対象期間内に占める割合を計算する．閲覧対象期間の始点を少しずつずらし，データが存在した期間内での平均を求める．閲覧対象期間の幅を変化させた場合の平均値の変化を表したものが図 12 である．これを見ると，全体的にデータの閲覧可能性が向上していることが分かる．また，収集間隔と閲覧対象期間の幅が一致している時に閲覧可能性が急激に向上する傾向があるが，これは提案したアルゴリズムにおいては，収集したデータを基準に閲覧の一貫性を決定しているためであり，収集時点と閲覧対象期間の始点，終点が一致している場合には閲覧可能性は 100%となる．このことは，データごとの更新頻度に合わせた収集を行えば閲覧可能性を大きく向上できることを示しており，このことを用いた最適な収集戦略を見つけることが，今後の課題といえる．

次に，閲覧対象期間幅を短く (例えば 1 分以下) とった場合について議論する．この場合は，閲覧可能性の向上は見られなかった．これは，提案した手法は収集したコレクションを基準としているため，閲覧対象期間が短かくなるにつれ，Not Found なページの数が増加するためである．よって，データの収集間隔から，最適と思われる閲覧対象期間の幅を閲覧者に伝えることも必要となってくるであろう．また，収集時に最終更新時刻を取得できない，もしくはその時刻が信用できない (収集時点より後である場合など) ときは，収集を行った期間 $[\,C_s(n), C_e(n)\,]$ のみが *Determinable* な期間となり，この場合の閲覧可能性は大幅に低くなると思われる．こういった場合も含めたより一貫性のある閲覧方法を考えることも今後の課題といえる．

5. まとめと課題

本稿では，Web アーカイブコレクションを閲覧する際の時系列一貫性のあるリンク参照を，コレクションの収集期間と閲覧対象区間との関係から決定するアルゴリズムを提案した．まず，収集実行時に参照できたリンクに関しては一貫性があるものと定義して，そこから様々な状態における一貫性のある参照を定義した．また，提案した手法が既存の Web アーカイブが抱える問題のいくつかを解決できることを示し，シミュレーションから収集したデータに対して提案した手法を用いることで，閲覧可能性を向上出来ることを示した．今後の課題としては，実際のアーカイブへの実装や，提案した手法の特性を活かせる収集戦略の決定などが上げられる．

謝辞　本稿の一部は，文部省科学研究費 (16016248, 13680482) の研究成果によるもので，ここに記して謝意を表します．

文　献

[1] Arms, W., Adkins, R., Ammen, C., and Hayes, A., “Collecting and Preserving the Web: The Minerva Prototype,” RLG DigiNews, Vol.5, No.2, 2001.4.15.

[2] Abiteboul, S., Cobena, G., Masanes, J., and Sedrati, G., “A First Experience in Archiving the French Web,” Research and Advanced Technology for Digital Libraries, Springer, 2002.

[3] Day, M., “Collecting and preserving the World Wide Web,” (online), available from http://www.jisc.ac.uk/uploaded_documents/archiving_feasibility.pdf, (accessed 2005.2.3).

[4] Hallgrimsson, P. and Bang, S., “Nordic Web Archive,” 3rd ECDL Workshop on Web Archives, 2003.8, (online), available from http://nwatoolset.sourceforge.net/docs/nwa@ecdl2003.pdf, (accessed 2005.2.3).

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

[6] 河野浩之, 川原稔, “Web 検索におけるテキストマイニング,” 人工知能学会誌, Vol.16, No.2, pp.212-218, 2001.

[7] Kawano, H., “Web archiving strategies by using web mining techniques,” Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, CD-ROM, 2003.

[8] 小城正士，廣瀬信己，河野浩之，“階層型記憶装置を用いた Web アーカイビングシステムの提案,” ディジタル図書館，No.24，pp.62-69，2003．

[9] 小城正士，廣瀬信己，河野浩之，“Web アーカイブにおける長期ストレージシステムの提案,” DBWeb2004，Vol.2004, No.14，pp.33-40，2004．

[10] 総務省編『情報通信白書平成 16 年版』2004.7, (online), available from http://www.johotsusintokei.soumu.go.jp/whitepaper/ja/h16/, (accessed 2005.1.5).

[11] Lamport, L., “Time, clocks, and the ordering of events in a distributed system,” Communications of the ACM, Vol.21, Issue 7, 1978.

[12] Lamport, L., “Synchronizing Time Servers,” SRC Research Report 18, 1987, (online), available from http://research.microsoft.com/users/lamport/pubs/synchronizing-time-servers.pdf, (accessed 2005.2.3).

[13] Lyman, P., “Archiving the World Wide Web,” Building a National Strategy for Preservation: Issues in Digital Media Archiving,” 2002.4, (online), available from http://www.clir.org/pubs/reports/pub106/web.html, (accessed 2005.2.3).

[14] Lyman, P. and Varian, H. R., “How Much Information? 2003,” 2003.10, (online), available from http://www.sims.berkeley.edu/research/projects/how-much-info-2003/, (accessed 2005.2.3).

[15] 国立国会図書館, “電子情報保存に係る調査研究報告書,” pp.101, 2003.3, (online), available from http://www.ndl.go.jp/jp/aboutus/preservation_02_01.html, (accessed 2005.1.5).

[16] Yamana, H., Tamura, K., Kamei, S., Kawano, H. et al., “Experiments of Collecting WWW Information using Distributed WWW Robots,” Proc. of SIGIR’98, Melbourne, Australia, pp.379-380，1998.