2004－DBS－134　(II)　(54)
2004／7／14

WWWにおける信頼度の高いリンクの発見

中溝昌佳[†]　森嶋厚行[††]
杉本重雄[††]　北川博之[†††]

WWW は社会的に重要なメディアの一つとなったが，その特徴である動的な更新，分散管理などの理由により，WWW コンテンツの一貫性維持は一般に困難である．我々は，WWW のリンク切れやリンク先の内容の変化への対応という，リンクの一貫性維持を支援するシステムの研究開発を行ってきた．我々のシステムのポイントは，信頼度が高いリンクである「リンクオーソリティ」の概念を導入したことである．しかし，これまでに提案したシステムにおいては，リンクオーソリティの利用は，(1) 利用者がシステムに明示的にリンクオーソリティを与えた場合，もしくは (2) システムがたまたまリンクオーソリティを発見した場合，に限定されていた．リンクオーソリティはリンク一貫性維持のための有力な手がかりであるため，これを積極的に利用するためにリンクオーソリティの発見を支援する仕組みが必要と考えられる．本稿では，リンクオーソリティを利用者に提供するリンクオーソリティサーバと，リンクオーソリティを発見するためのアルゴリズムを提案する．

Discovery of Reliable Links in WWW

Akiyoshi Nakamizo,[†] Atsuyuki Morishima,[††] Shigeo Sugimoto[††]
and Hiroyuki Kitagawa [†††]

WWW has become one of the important media in our society, but it is difficult to maintain the integrity of its contents in general. We have been developing a system to maintain the integrity of links, which can cope with dead links and changes of the page contents of link destinations. The point of our system is that we introduced the concept of link authority, which means a well-maintained link we can depend on. In the previous works on our system, however, link authorities are used only when (1) they are explicitly specified by users, or (2) the system happens to find them. Since link authorities are effective in maintaining the integrity management of links, a mechanism to automatically find link authorities would be beneficial. This paper proposes a link authority server, which provides users with information on link authorities, and explains an algorithm to find link authorities.

1. はじめに

WWW は既に社会的に重要なメディアの一つとなったが，その特徴である動的な更新，分散管理などの理由により，WWW コンテンツの一貫性の維持は一般に困難である．

我々は，WWW のリンク切れやリンク先の内容の変化への対応という，リンクの一貫性維持の問題に焦点を当てたプロジェクトを推進している[1)2)3)]．これまで，リンク切れやリンク先の内容の変化に対して，自動的にリンクを更新し一貫性維持を試みるソフトウェア (Link Integrity Maintenance サーバ, LIM サーバ) の開発を行い，実験を通じてその実用可能性が高いことを示してきた[1)]．我々の手法では，信頼度が高いリンクである「リンクオーソリティ」の概念を導入したことがポイントである．

リンクオーソリティとは，リンク先の内容が変化した時にリンクを確実に変更するページのことを指す☆．たとえば，ある Web ページ p が，別の Web ページ q へのリンクを持っていたとする．このとき，q が q' に移動したとき，p の中の q へのリンクを q' に確実に変更するようなページ p をリンクオーソリティと我々は定義している．

† 芝浦工業大学大学院 工学研究科
Grad. Sch. of Eng., Shibaura Inst. of Tech.
†† 筑波大学 知的コミュニティ基盤研究センター
RCKC, Univ. of Tsukuba.
††† 筑波大学大学院 システム情報工学研究科
Grad. Sch. of Sys. and Info. Eng., Univ. of Tsukuba.

☆ Google などにおける Authority ページとは異なる概念である．

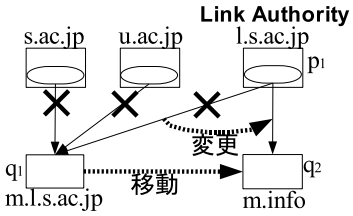

図 1 Link Authority

ここで，例を用いて説明する (図 1)．まず，ある大学 A の研究室の Web ページ q_1 が存在し，このアドレスが m.l.s.ac.jp であるとする．q_1 は，s.ac.jp，l.s.ac.jp，u.ac.jp という複数のページからリンクされている．これらのうち，p_1（l.s.ac.jp）はその研究室が所属する学科の研究室リストページである．このとき，一般には p_1 は q_1 に関する Link Authority である．したがって，例えば次のような状況が生じる．q_1 が q_2（アドレスは m.info）に移動したとする．すると，q_1 を参照しているページではリンク切れが発生するが，通常 p_1 だけは q_1 へのリンクを q_2 に張り替えるはずである．

LIM サーバはリンク切れ等を発見すると，更新後のリンク (ページの移動先へのリンク) を探すために，リンクオーソリティを利用する．しかし，これまでに我々が提案してきた手法[2)1)] では，リンクオーソリティの利用は次の場合にのみ限定されていた．(1) 利用者により明示的に与えられた場合．(2)LIM サーバが，移動先の探索過程でたまたまリンクオーソリティを発見した場合．これらの結果，リンクオーソリティはリンク一貫性維持のために有力な手がかりであるにも関わらず，実験などの際に実際に利用されたケースはそれほど多くはなかった．

本稿では，リンクオーソリティの情報を集め利用者に提供するリンクオーソリティサーバ (LA サーバ) の開発について説明する．具体的には，LA サーバは，あるリンク (URL) u を入力として受け取り，u が指すページに関するリンクオーソリティの URL v を求める．現実には，リンクオーソリティであるかどうかの決定は一般に困難である．したがって，LA サーバは，v の可能性が高いと考えられる URL 群をランキングしたリストを出力する．

関連システム，研究． まず，各種のリンク切れ発見ツール（リンクチェッカ）が存在する．これらは，指定されたページに記述されたリンクについて，リンク切れを起こしていないかをチェックをする．リンクチェッカを用いてリンク切れを発見すると管理者に対してメールで通知することで，リンクの更新管理を行っているサイト[5)] も存在する．また，Web のリンクにおける論理的不整合を検出する手法についての研究[4)] も行われている．しかし我々の知る限り，リンクの信頼度に着目した研究は存在しない．

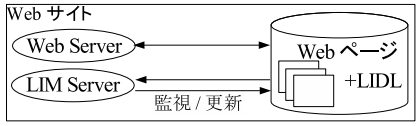

図 2 リンク一貫性維持支援システムの概要

2. リンク一貫性維持支援システムの概要

我々が開発中の，リンク一貫性維持支援システムの概要を図 2 に示す．本システムは，リンクに関する制約を記述する LIDL (Link Integrity Description Language) と，リンクの監視などを行う LIM サーバ (Link Integrity Maintenance Server) の二つによって構成される．これらは,「Web サーバとファイルシステムに格納された Web ページ群」という通常の Web アーキテクチャに追加する形式で利用する．本システムの利用は次のようになる．まず，Web ページ作成者は，ページ内のリンクに対して LIDL によって「リンク切れがあってはならない」という制約を記述する．すると，LIM サーバは制約の記述されたリンクを監視し，リンク切れ (つまり制約を満たさないリンク) を発見すると，代わりとなるリンク先ページの候補 (群) を求め，ユーザに提示する．具体的には，ユーザに対してメールで通知するといったような方法を用意している．

本システムの特徴は次の通りである．(1) 通常の Web アーキテクチャの自然な拡張であり，既存の Web コンテンツとも容易に組合せ可能である．(2) 単純かつ強力な制約記述言語を提供する[2)]．

2.1 LIDL による制約の記述

制約の記述は，Web ページのリンク (A タグ) に，制約を表すための属性を追加することによって行う．LIDL では 3 種類の制約 (isAlive, matches, follows) を定義している．本稿では，紙面の都合上，isAlive と follows についてのみ述べる．

[isAlive 属性] isAlive は，“リンク先のページがリンク切れでない” ことを表す制約である．次に例を示す．

```
<a li:isAlive
 href="http://www.mlab.info/news.html">ニュース</a>
```

ここで，li は LIDL の名前空間である．この例では，

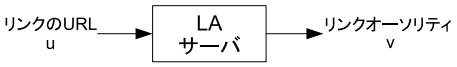

図 3 LA サーバの概要

リンク先のページ (http://www.mlab.info/news.html) がリンク切れであってはならないという制約を指定している．

[**follows 属性**] follows は，リンクオーソリティを明示的に指定する制約である．次に例を示す．

```
<A li:follows="l.s.ac.jp"
    href="http://m.l.s.ac.jp/">某研究室</>
```

この例のように制約が指定された場合，follows による制約が満たされなくなるのは，指定されたリンクオーソリティにおいて m.l.s.ac.jp へのリンクが変更されるか，もしくはリンクオーソリティ (l.s.ac.jp) そのものが存在しなくなった場合である．

2.2 LIM サーバによる制約の解釈

1 つのリンクに対し，LIDL による制約は複数記述可能である．例えば，isAlive と follows を共に指定できる．LIM サーバでは，指定されている制約が満たされないことを発見すると，ページが移動したと解釈し，移動先と考えられかつ制約を満たすような代わりのリンク先ページの探索を行う．実際には，ページの移動先が見つからない場合や，そもそも単にページが消滅した場合もある．

また，LIM サーバはデフォルトで isAlive の指定が存在すると判断する．このようなデフォルトの制約の存在により，既存の WWW コンテンツであっても LIM サーバの機能を利用可能である．

3. リンクオーソリティサーバの開発

本稿で提案する LA サーバは，リンクオーソリティの情報を利用者や他のソフトウェアに提供するシステムである (図 3)．具体的には，LA サーバは，リンク (URL) u を入力とし，u が現在指しているページに関するリンクオーソリティの URL v を出力する．実際には，ある一つのページをリンクオーソリティであると特定することは困難である．そのため，LA サーバでは，リンクオーソリティである可能性が高いと考えられる候補 (群) を，ランキングし出力する．

LA サーバの利用例としては，LIM サーバとの連携があげられる (図 4)．まず，LIM サーバの利用者は監視させたいリンクの URL u を LIM サーバに登録する．もし，u がリンク切れなどを起こし，LIM サーバが代替リンクの収集を行う際，LIM サーバは LA サーバに u のリンクオーソリティ情報を問い合わせる．LIM サーバは，そのリンクオーソリティ情報を，代替リンク先の探索に利用する．

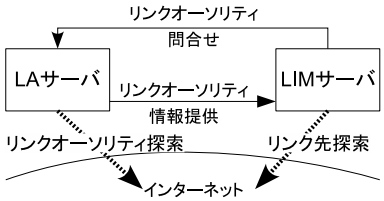

図 4 LIM サーバと LA サーバの連携

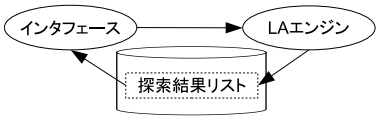

図 5 LA サーバのアーキテクチャ

LA サーバの構成要素は，システムの利用者や外部システムとの接続を担う**インタフェース部**と，リンクオーソリティの探索を行う **LA エンジン部**である (図 5)．

各構成要素は，具体的に以下の役割を持つ．

インタフェース部 外部システムとの連携を行うためのインタフェースを提供する．具体的には，先に説明した LIM サーバとの連携をするための API の提供などをする．

LA エンジン部 LA エンジン部は，リンクオーソリティ候補の探索を行う．また，見つけだされた各候補にスコアを付け，ランキングを行う．LA エンジン部が行う探索は，後述するリンクオーソリティ探索アルゴリズムを用いて行われる．

4. リンクオーソリティ探索アルゴリズム

本節では，ある Web ページ u のリンクオーソリティを探索するアルゴリズムについて説明する．

本アルゴリズムは，以下の 2 つの構成要素を定期的に行う．

[**1:リンクオーソリティ候補の収集**] u のリンクオーソリティの候補 (URL) の収集を行い，候補のリスト V を作成する．

[**2:収集された候補群のランキング**] 収集された候補群の各候補 $v_i \in V$ にスコアをつけ，V をスコア順に並べ替える．

1:リンクオーソリティ候補の収集によって収集され，**2:収集された候補群のランキング**によってランキングされた候補リスト V が本アルゴリズムによる探索結

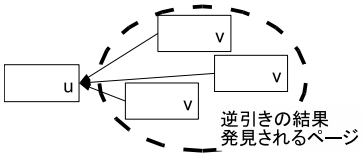

図 6 逆引き検索

果となり出力される．

本アルゴリズムでは，次に説明するリンクオーソリティの調査を定期的に行う．調査を定期的に行う理由は，(1) リンクオーソリティの情報を常に最新に保つため，および (2) 後述するヒューリスティクス H2-4 を用いるため，である．

以下に，アルゴリズムの詳細を説明する．

4.1 リンクオーソリティ候補の収集

u に対応するリンクオーソリティ候補の収集を行い，候補 URL のリスト V を作成する．

我々は，候補の収集をヒューリスティクスを用いることで実現する．用いるヒューリスティクスを以下に示す．

H1-1 リンクオーソリティとなるページは，リンクの逆引き検索結果の中に存在する可能性がある．ここで「逆引き検索」とは，リンク u を指しているページの集合を検索するものであり，google などで実現されている (図 6)．これにより，u に対してリンクを張っているページを収集する．

H1-2 同一サイト内にリンクオーソリティとなるページが存在する可能性がある．

H1-3 u より上位のドメインのサイトにリンクオーソリティとなるページが存在する可能性がある．

H1-4 u と相互リンクをしているページ (群) の中に，u のリンクオーソリティとなるページが存在する可能性がある．

これらのヒューリスティクスを利用して，具体的には以下の方法でリンクオーソリティ候補の収集を行う．

H1-1 による収集 サーチエンジンの逆引き検索機能を利用して収集する．

H1-2 による収集 u と同一のサイトの中から，u へリンクを持つページを収集する．しかし，サイトのルートを起点にページを収集するのは，特に大規模サイトの場合などに効率が悪い．そのため，後述する H2-1 を考慮し，本システムでは u のページの属するディレクトリを起点にし上位のディレクトリへの順番で収集を行う．次に例を示して説明する．図 7(a) はサイトのルートから u までのディレクトリ構造を示している．この場合，u の上位ディレクトリ (/, /dir1/, /dir1/dir2/) それぞれのデフォルトのページ (index.html) からリンクをたどって，リンクオーソリティとなりうるページを収集する．

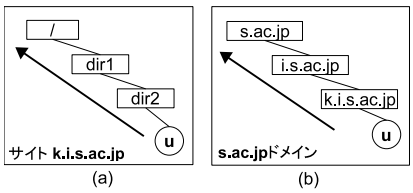

図 7 同一サイト内の探索順序 (a) と同一ドメイン内の探索順序 (b)

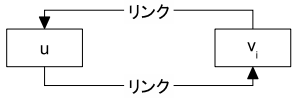

図 8 相互リンク

H1-3 による収集 u のサイトより上位のドメインのサイトから u に対してリンクを張っているページを収集する．以下に例を用いて説明する (図 7(b))．ここで，u のページが属するサイトのドメインが k.i.s.ac.jp だとする．このとき，k.i.s.ac.jp より上位のサイトは i.s.ac.jp と s.ac.jp である．これらのサイトのルートにあるデフォルトのページ (index.html) を起点に候補を収集する．

H1-4 による収集 u のページが持つリンクを頼りに，相互リンクしているページ (群) を収集する．ここで，「相互リンク」とは，u のページからリンクをされていて，かつ u へのリンクを持っていることを意味する (図 8)．つまり，u のページからリンクしているページ (群) から，u へのリンクを持つページを収集する．

これらの方法により収集された候補を並べた候補リスト V を作成する．

4.2 リンクオーソリティ候補のランキング

探索によって見つけだされたリンクオーソリティの各候補 $v_i \in V$ に，次で説明する基準によってスコアを割り当て，スコアの高い順にランキングする．

以下では説明を簡単にするため，URL におけるドメインの構造とディレクトリの構造は区別せず，単一のディレクトリして扱う．たとえば，s.ac.jp/dir1/u.html は/jp/ac/s/dir1/u.html とする．また，すべての最上位ドメインは仮想のルートノードを共有しているとして扱う．これにより，インターネット上に存在するすべての Web ページがディレクトリで表現できるようになる．図 9 に，ディレクトリの木構造の例を示す．

以下にスコア付けに用いるヒューリスティクスを示す．

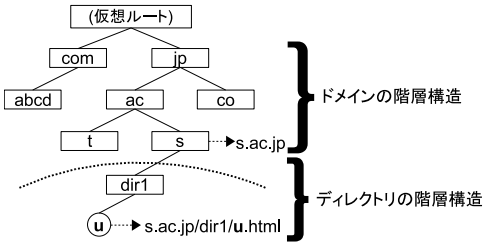

図 9 木構造の例

H2-1 ディレクトリの木構造において，リンクオーソリティ候補のページが属するノードが，u のページが属するノードより上位の場合，その候補はリンクオーソリティである可能性が高い．

H2-2 同一ディレクトリ内にリンクオーソリティ候補が複数存在する場合，その中でもディレクトリのデフォルトのページ (index.html 等) はそれ以外のページよりリンクオーソリティである可能性が高い．

H2-3 u のページとリンクオーソリティ候補のページとが相互リンクをしている場合，その候補はリンクオーソリティである可能性が高い．

H2-4 定期的な調査によって，常に正しくリンクを保ち続けているリンクオーソリティ候補は，リンクオーソリティである可能性が高い．

収集されたリンクオーソリティ候補 $v_i \in V$ には，スコア $score(v_i, u)$ を割り当てられる．$score(v_i, u)$ の計算式は以下の通りである．

$$score(v_i, u) = A \times B \times C \times D$$

以下に，A, B, C, D それぞれの値の割り当て方について述べる．

H2-1 によるスコア付け リンクオーソリティ候補 v_i のページが，ディレクトリの木構造における u より上位ノード (/jp/ac/s/, /jp/ac/, /jp/, /) に属する場合，$A = \alpha$ とし．それ以外の場合は $A = 1$ とする．ここで，ディレクトリ構造において v_i と u の距離が近いものほど高く評価する．つまり，v_i と u とが同一ディレクトリにある場合に最も高い評価となり，逆に v_i がルートに近づくほど低い評価となる．これを満たすような重み係数 α を以下に定義する．

$$\alpha = min + \frac{max - min}{N} \times n$$

ここで，max は 重み係数 α の最高値，min は最低値である．また，N, n はそれぞれ u, v_i の属するディレクトリの深さである．たとえば，$max = 1.5$, $min = 1.0$, $N = 10$, $n = 8$ としたとき，α の値は以下のようになる．

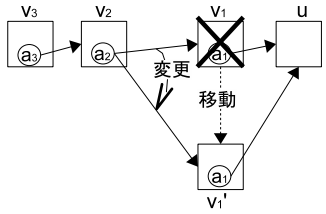

図 10 「リンクオーソリティのリンクオーソリティ」の利用

$$\alpha = 1.0 + \frac{1.5 - 1.0}{10} \times 8 = 1.4$$

H2-2 によるスコア付け リンクオーソリティ候補 v_i のファイル名がディレクトリのデフォルトページ (index.html 等) である場合，$B = \beta$ とし．それ以外の場合は $B = 1$ とする．

H2-3 によるスコア付け u のページとリンクオーソリティ候補 v_i のページとが相互リンクをしている場合，$C = \gamma$ とし．それ以外の場合は $C = 1$ とする．

H2-4 によるスコア付け リンクオーソリティ候補 v_i のページのリンクが常に正しい場合，$D = \delta$ とし．それ以外の場合は $D = 1$ とする．

各重み係数 (α, β, γ, δ) は，条件に当てはまった候補のスコアを上げるために用いる係数である．そのため，各重み係数とも必ず 1 以上の値を割り当てる．

ここで求められたスコアによってランキングされた候補リスト V が本アルゴリズムの探索結果として出力される．

5. 移動したリンクオーソリティの発見

一般に，発見されたリンクオーソリティは永遠に有効であるとは限らない．例えば，リンクオーソリティそのものが移動してしまう可能性がある．そのような場合には，移動先のリンクオーソリティを発見する必要がある．このような状況に対応するためには「リンクオーソリティのリンクオーソリティ」を保存しておくことが考えられる．そうすれば，一段上のリンクオーソリティを調べることによって，移動先のリンクオーソリティを発見できる可能性がある (図 10)．

しかし現実には，完全にそのような仕組みを実現することは難しい．その理由は次の通りである．(1) 一般に，このような「リンクオーソリティのリンクオーソリティ」の並びは無限に続く．(2) LA サーバが求めるものはあくまでリンクオーソリティの候補であり，本当にリンクオーソリティであるという保証はない．したがって必ずしも信頼できない．

以上のような問題はあるが，本システムでは次のような制限のもとで,「リンクオーソリティのリンクオー

```
Global List of String A=[];
Link newLinkAuthority(v:url, L: List of URLs) {
  L.drop(); // 先頭の (v,a) を除去する
  (v', a')=L.head(); // v のリンクオーソリティ
  A=A.add(a'); // アンカー文字列のリストに追加
  if (dead(v')) {// リンクオーソリティも移動
    if (L.length()>1) {
         v'= newLinkAuthority (v', A, L);
         if (A==[]) return u';
    } else null; // 発見できなかった.
  }
  if (v'!=null) {
     // リンクオーソリティの移動先を発見
     a= v' のページが持つ全てのリンクの中で,
        最も A にマッチするアンカーを持つもの;
     A=A-アンカーにマッチした文字列;
     return a;
  } else null;
}
```

図 11 移動したリンクオーソリティの発見処理

ソリティ」の並びを計算し，保持する．(1) リンクオーソリティの並びの数は，あらかじめ与えられた URL から固定数 (例えば 20) だけたどることとし，それ以降は保存しない．(2) LA サーバによって求められるリンクオーソリティ候補のうちもっともスコアが高いものを正しいリンクオーソリティであると仮定して処理する．

このように LA サーバに保存された「リンクオーソリティのリンクオーソリティ」の並びを用いて，リンクオーソリティが移動した場合には，その移動先の発見を行う．実際の処理では，保存するものはリンクオーソリティだけの並びではなく，組 (リンクオーソリティ，リンクオーソリティからのリンクが持つアンカー文字列) の並びである．図 10 の場合は，$L = [(v_1, a_1), (v_2, a_2), (v_3, a_3)]$ となる．ここで，v_{i+1} が v_i のリンクオーソリティである．このとき，v_1 が移動した場合には，関数 `newLinkAuthority`(v_1, L)(図 11) を実行する．この処理では，まず上位のリンクオーソリティにむかって順にたどり，アクセス可能なリンクオーソリティを探す．見つかると，次はアンカー文字列を頼りに下位のリンクオーソリティに j 順次移動し，最終的に移動先と考えられるリンクオーソリティを探す．

6. おわりに

本稿では，信頼できるリンクである「リンクオーソリティ」の情報を提供するリンクオーソリティサーバと，本サーバがリンクオーソリティを発見するためのアルゴリズムを提案した．今後は，システムの実装およびそれを用いた実験を行う．またその結果を基に，適切な係数の設定やアルゴリズムの改良などについても検証する予定である．

謝　辞

ゼミなどでご議論いただきました芝浦工業大学工学部情報工学科の古宮誠一教授に感謝いたします．本研究の一部は日本学術振興会科学研究費補助金若手研究 (B)(課題番号 15700108) による．

参 考 文 献

1) 中溝昌佳, 有山智洋，森嶋厚行, 杉本重雄，北川博之．WWW におけるリンク一貫性維持支援システムの開発．電子情報通信学会第 15 回データ工学ワークショップ (DEWS2004), 2004 年 3 月.
2) 中溝昌佳，森嶋厚行，有山智洋，杉本重雄，北川博之．WWW コンテンツ一貫性維持のためのリンク更新機構の提案．日本データベース学会 Letters, Vol. 2, No. 2, pp. 65-68, 2003 年 10 月
3) 中溝昌佳，有山智洋，森嶋厚行，杉本重雄，北川博之，リンク切れ対応機能を持つ HTTP プロキシの開発．情報処理学会第 66 回全国大会講演論文集 (3), pp. 57-58, 2004 年 3 月.
4) 河合英紀, 河野泉, 石黒義英, 福島俊一, サイト品質管理のためのリンク不整合検出．電子情報通信学会第 15 回データ工学ワークショップ (DEWS2004), 2004 年 3 月.
5) Planet SOSIG - A spring-clean for SOSIG: a systematic approach to collection management: http://www.ariadne.ac.uk/issue33/planet-sosig/