Webページ移動先発見のための公開実験システム

A System for Open Experiments on Finding Moved Web Pages

飯田 敏成 ♡ 澤 菜津美 ♢ 森嶋 厚行 ♠
杉本 重雄 ♣ 北川 博之 □

Toshinari IIDA Natsumi SAWA
Atsuyuki MORISHIMA Shigeo SUGIMOTO
Hiroyuki KITAGAWA

我々は Web ページの移動先を発見し，ページ移動によるリンク切れの自動修正を試みるシステムを開発して，実験を行ってきた．現在，次のフェーズとして Web ページ移動先追跡の公開実験を行うことを考えており，そのために必要な公開実験システムの開発を行っている．本論文では，この公開実験システムの開発について説明する．

We have developed a system for trying to find moved Web pages and correct broken links, and performed experiments with the system. Currently, we are developing another system to be used in open experiments for finding moving Web pages. This paper explains the development of the system.

1. はじめに

近年，World Wide Web (以下 Web) は社会における重要なメディアの一つとして大きな役割を果たしている．Web の特徴の一つとして分散管理が行われていることが挙げられる．この特徴は，Web を便利なツールとする一方で，Web コンテンツの一貫性の維持を困難としている要因でもある．コンテンツの一貫性が損なわれる一例として Web のリンク切れがあり，我々はこれに着目している．

我々はこれまでに次のような 2 つのシステムの開発，およびそれらを用いて実験を行ってきた．一つは，Web のリンク切れを発見すると変更先と考えられるリンクの候補を自動的に発見してリンクの修正を試みる LIM(Link Integrity Management) サーバである [4] ．もう一つは，移動した Web ページを発見するための強力な手がかりとなるリンクオーソリティを提供する LA(Link Authority) サーバである [3] .

これまで行ってきた実験では，人工的に集められたページ集合が対象として用いられてきた．我々は，次のフェーズとして公開実験を行うことを考えており，そのために必要な公開実験システムの開発を行っている．本稿では，公開実験システムの開発について説明する．

関連研究． リンク切れの自動修正を試みるシステムは，我々の知る限り IBM の Peridot[1][2] だけである．このシステムは，リンク切れなどによって一貫性が損なわれたリンクを自動的に別のページへのリンクに修正することを試みる．リンク切れの修正システムではないが，関連する研究としては lexical signunature[5] がある．この 2 つに共通している点は，Web コンテンツから抽出した特徴的なキーワードを利用してページの同定を行うアプローチを採用していることである．つまり，既にインデクシングされた Web ページ集合の中からリンクの修正候補として最適な Web ページを発見することを想定している．それに対して，我々はインデクシングされているという前提が存在しない中で，Web 中からリンクの修正候補として最適な Web ページを発見することを試みる．具体的には，我々のシステムは「移動した Web ページがどこに存在していそうであるか？」に関するヒューリスティクスを利用してリンクの修正候補の探索を行う．このように，本システムは他のリンク切れ処理関連技術にない特徴を備えている．また，我々の知る限りリンク切れの修正支援を目的とした公開システムは存在しない．

♡ 学生会員 筑波大学大学院 図書館情報メディア研究科
toshi@slis.tsukuba.ac.jp
♢ 学生会員 筑波大学 図書館情報専門学群
n163@slis.tsukuba.ac.jp
♠ 正会員 筑波大学大学院 図書館情報メディア研究科
mori@slis.tsukuba.ac.jp
♣ 正会員 筑波大学大学院 図書館情報メディア研究科
sugimoto@slis.tsukuba.ac.jp
□ 正会員 筑波大学大学院 システム情報工学研究科
kitagawa@cs.tsukuba.ac.jp

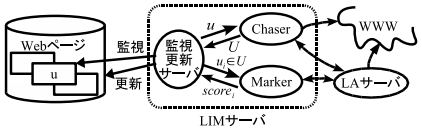

図 1 LIM サーバ アーキテクチャ
Fig. 1 Architecture of LIM Server

図 2 LA サーバの概要
Fig. 2 Outline of LA Server

2. リンク一貫性維持支援システム

本章では，我々が今までに設計・開発した LIM サーバおよび LA サーバについて述べる．ここでは本論文について必要な概要のみを述べる．LIM サーバの詳細は [4]，LA サーバの詳細は [3] にある．

2.1 LIM サーバ

LIM(Link Integrity Management) サーバとは，Web のリンク切れを発見すると自動的にその修正を試みるシステムである (図 1).

以下では LIM サーバの働きについて説明する．話を簡単にするために，ここでは，システムが監視対象とするリンクを URL u で表されるただ一つのリンクに限定する．本システムは監視下の u がリンク切れであることを発見すると，u の移動先 u_{new} を発見し，u へのリンクを u_{new} に自動修正することを試みる．本システムの主要な構成要素は，対象となるリンクの監視及び更新をする監視・更新サーバ，移動先のページの URL である u_{new} の候補集合 U を収集する Chaser，U の各候補 u_i に対して “移動先らしさ” を表すスコア $score_i$ を計算する Marker である．

LIM サーバの Chaser および Marker は，候補収集およびスコアリングの際に 6 つのヒューリスティクスを利用している．例えば，「u_{new} と u のページの内容は似ているはずである」というヒューリスティクスを利用している．

2.2 LA サーバ

LA(Link Authority) サーバとは，ある Web ページの URL u を入力とし，u のリンクオーソリティであると考えられる候補 URL リスト $\bar{V} = [v_1, v_2, \ldots]$ を出力するシステムである (図 2).

ここでいうリンクオーソリティとは，リンク先のページが移動したときにリンクを確実に変更するページのことである[1]．例えば，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．「q が q' に移動したとき，p の中の q へのリンクを q' に

[1] Google などにおける Authority ページとは全く異なる概念である．

確実に変更する」時，p を q のリンクオーソリティであると我々は定義している．

3. 公開実験システムの概要

公開実験システムを開発する目的は，一般の利用者からのフィードバックなどの有用な情報を収集することである．以下では，公開実験システムの概要について説明する．

3.1 公開実験システムの特徴

まず，我々が今までに行ってきた実験 [4][3] で利用したシステムと，公開実験システムとの相違について説明する．(1) 今までは我々が一定の規則に従って収集したリンクの集合をシステムの監視対象として実験を行ってきた．公開実験システムでは監視対象とするリンクを指定するのは利用者であり，利用者の要求に応じて監視対象とするリンクがダイナミックに増減する．(2) 今までは高々数万のリンクの集合をシステムの監視対象として実験を行っていた．公開実験システムではより多くのリンクを監視対象とする．(3) 今までは実験結果を手作業で解析していた．公開実験システムでは結果を自動的に解析する．(4) 今までは，リンク切れが起きた際にシステムが発見したページの移動先候補は解析のためだけに利用されていた．公開実験システムでは，リンク切れとなったリンクの修正候補として利用者に提供する．(5) 今までは公開を前提としていなかったのでユーザビリティを考慮していなかった．公開実験システムでは多くの利用者に利用してほしいので，簡単にシステムの操作および結果の閲覧ができるようにする．(6) 今までは小規模なリンクを対象として実験を行ってきたため，システムには効率化の余地が残されていた．公開実験システムでは対象が大規模になるため，リンク切れ発生から短時間で探索を行い，利用者に情報提供を行うためにシステムの効率化を図る．

3.2 ログとして記録する情報

本公開実験システムでは，LIM サーバおよび LA サーバが監視対象のリンクについて監視および探索を行う過程で得ることができる様々な情報をログに記録する．内容は，探索を行った URL，探索日時などである．

3.3 フィードバックとして受け取る情報

本公開実験システムでは，システムが提供したリンクの修正候補に対する利用者からのフィードバックを受け取る．ここでは，システムが受け取るフィードバックについて説明する．

- システムによって提供されたリンクの修正候補の中に正しい移動先があった場合には，その移動先を受け取る．正しい移動先が提供されなかった場合にも正しい移動先が提供されなかったというフィードバックを受け取る．
- システムによって提供されたリンクの修正候補の中に正しい移動先がなかったが，利用者が自分で正しい移動先を発見することができた場合にはその URL を受け取る．
- 利用者の登録したページの中でリンク切れとなったリンクは利用者にとってどのような種類のリンクであるかを受け取る．(例えば企業のページ，大学のページ，友人のページ，他人 (面識がない) のページなど．)
- 移動先の探索の際にキーワード検索に利用されたクエリが適切にページの内容を表していたかどうかを受け取る．

3.4 フィードバックの解析

本公開実験システムの利用者によるフィードバックは，以下のような解析に利用される．

- ユーザから得たフィードバックの総数 fb_all，発見に成功した場合のフィードバックの総数 $fb\_success$ から，システムが Web ページの正しい移動先の発見に成功した割合を計算する．

$$\text{移動先ページ発見成功率 (\%)} = \frac{fb\_success}{fb\_all} \times 100$$

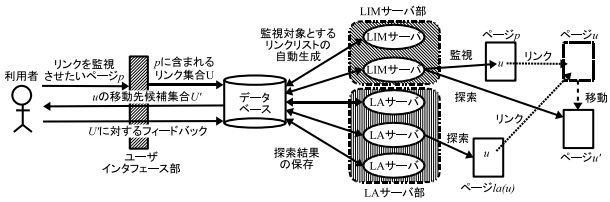

図 3 システムアーキテクチャ
Fig. 3 System Architecture

- フィードバックから得たリンクの種類を基に，リンクの種類に応じた発見成功数，発見失敗数の内訳がどのようになっているかを分析する．
- フィードバックから得た正しい移動先 URL と移動元 URL の比較などから，Web ページの正しい移動先が提供できなかった場合の原因が何かを考察する．
- キーワード検索で利用するためにシステムが選択したキーワードは適切であったかを考察する．

また，フィードバックと公開実験システムによって取得するログから次の解析を行うことができる．

- Web ページの正しい移動先の発見に成功した場合の，各ヒューリスティクスの貢献度を分析する．
- Web ページの移動先の探索およびリンクオーソリティの探索に要する平均時間を分析する．

4. システムの開発における問題点と工夫

3.1 節で述べたように，これまでの実験で利用していたシステムと公開実験システムでは利用方法が異なる．よって，これまでのシステムをそのまま利用するには次のような問題がある．以下では，それらの問題を解決するための工夫と共に説明する．

問題 1: 今までの実験では，我々が一定の規則に従って収集したリンクの集合からリンクリストを生成し，そのリンクリストを対象として実験を行っていた．リストの作成は一度だけで，以後リストは変化させなかった．また，数多くのリンクに対して短時間で探索するためにサーバを複数台 (3 台) 用意して実験を行っていた．その際リストが固定だったので，リストを単純に 3 分割して各サーバで探索を行っていた．しかし公開実験システムでは対象とするリンクが利用者の要求に応じてダイナミックに増減するため，今までのリストの分割手法では各サーバに対して対象となるリンクを適切に分配することができない．

問題 2: 今までの実験では対象とするリンクが固定だったが，公開実験システムでは対象とするリンクがダイナミックに変動するため，頻繁に探索が行われているリンク，まだ一度も探索が行われていないリンクなどが混在する．そのなかで，全てのリンクに対して網羅的に探索を行い，かつ最終探索日時の平均が大きくなるようにしたい．そのために，公開実験システムでは探索を行う際の探索順序を考慮する必要がある．

問題 1 および問題 2 に対する工夫: 今までの実験では，各ヒューリスティクスに基づく探索の処理を順番に行い，最後にスコアリングを行っていた．公開実験システムではそれぞれのヒューリスティクスごとに探索の処理を平行して行う．これによって時間あたりの探索効率を向上させる．今までのシステムと公開実験システムの違いを図 4 に示す．

また，今までの実験では，システムの管理者があらかじめ固定されたリンクリストを LIM サーバおよび LA サーバに対して与えていたが，これでは監視対象ページのリンクの増減に対応できない．この問題に対処するため，公開実験システムアーキテクチャでは，図 3 に示すように各 LIM サーバおよび LA サーバが自律的にリンクリストを作成する仕組みを用意する．このアーキテクチャでは，データベースが監視対象のリンクリストを保持しており，各サーバはそのデータベースにアクセスして，自分が処理すべきリンクリストを作成する．以下では，LIM サーバがデータベー

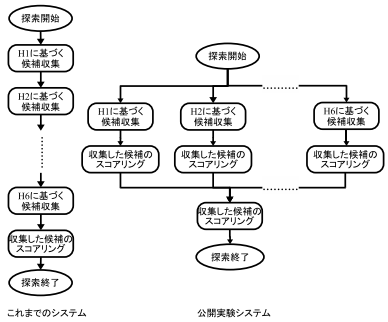

図 4 これまでのシステムとの違い
Fig. 4 Difference from the previous system

監視対象 URL	リンク切れ日時
u1.jp	2005/6/1 12:00
u2.jp	2005/6/5 10:00
u3.jp	2005/6/11 20:00
u4.jp	2005/6/9 20:00
u5.jp	2005/6/4 10:00
u6.jp	2005/6/10 1:00
u7.jp	2005/6/10 12:00
u8.jp	2005/6/2 16:00
u9.jp	2005/6/8 23:00
u10.jp	2005/6/3 12:00

図 5 リンク切れしたリンク一覧
Fig. 5 Broken link list

探索した URL	最終探索日時
u1.jp	2005/6/2 0:00
u5.jp	2005/6/5 0:00
u8.jp	2005/6/3 0:00
u10.jp	2005/6/4 0:00

図 6 LIM サーバが h に基づく探索を行ったリンクの最終探索日時一覧
Fig. 6 List of dates when the LIM server tried to find new links before

スから得たデータを基にリンクリストを作成し，そのリンクリストを利用して検索を行う手続きについて説明する．この手続きを擬似コードで表現したものを図 8 に示す．LIM サーバは 2.1 節で説明したように，システムに組み込まれたヒューリスティクスを用いて処理を行う．各ヒューリスティクスごとに処理は並列して行われるため，検索の手続きも各ヒューリスティクスごとに行われる．図 8 の手続き Search-LIM$_h$ の h はその手続きで利用されるヒューリスティクスを表している．具体的には，2.1 節で示したヒューリスティクス H1 などに対応する．

図 8 の Target は，リンク切れしたリンクの集合である．Target の例を図 5 示す．そして status は，過去に探索したリンクと探索日時である．status の例を図 6 示す．擬似コードの 3-4 行目が，自律的なリンクリストの作成部分である．具体的には，まずデータベースにアクセスして status を得た後，sort メソッドを呼び出し，Target(図 5) と status(図 6) から図 7 のように，未探索のリンク，探索日時の古いリンクの順にリンクをソートする．そして上位 N 個のリンクを取得し，このリンクをリンクリストと見なす．そしてリンクリストの各リンクについて探索を行う．図 8 では LIM サーバの探索について示したが，LA サーバの各ヒューリスティクスに基づく探索においても同様である．

問題 3: 今までは，我々以外がシステムを利用することを想定していなかったのでユーザビリティを考慮していなかった．しかし公開実験システムではできるだけ簡単に操作できるようにして多くの人に利用してもらいたい．

問題 3 に対する工夫: 公開実験システムでは，監視を行う Web ページの登録およびシステムの探索結果の閲覧のためのユーザインタフェースを提供する．これによってユーザビリティを向上させる．ユーザとシステムの間のインタラクションは以下の通りである．

システムへの登録時:

1. リンクの監視を希望する Web ページの URL p を送信する (図 9 画面例 1)
2. p のページに含まれるリンク一覧が表示されるので，監視を行うリンクを指定する (図 9 画面例 2)
3. システムへの登録が完了し，リンクの監視が行われる

監視していたリンクが切れたとき:

探索順序	探索のためのリンクの集合
1	u3.jp
2	u7.jp
3	u6.jp
4	u4.jp
5	u9.jp
6	u2.jp
7	u5.jp
8	u10.jp
9	u8.jp
10	u1.jp

図 7 ソートされたリンク一覧
Fig. 7 Sorted link list

```
1. void Search-LIM_h() {
2.     while() {
3.         list status = getStatusFromDB(h);
4.         list target = Top-N(sort(status));
5.         for(i = 0; i < N; i++) {
6.             setOfURLs result = target(i) について h に基づいた探索結果;
7.             setResultToDB(result, h);
8.         }
9.     }
10. }
11.
12. list sort(List status) {
13.     list target = Target と status を結合した結果;
14.     target = 過去に探索を行っていないリンク，探索日時が古いリンクの順に
15.              ソートした結果;
16.     return target;
17. }
```

図 8 LIM サーバの H1 に基づく手続き
Fig. 8 Search procedure for LIM Server based on H1

1. p においてリンク切れが発生
2. LIM サーバが p の移動先候補を計算
3. システムから利用者に結果表示用の URL をメールで送信し，利用者は提示された結果をもとに p の移動先 p' を発見
4. 利用者は結果に対するフィードバックを送信 (図 10)

問題 4: 今までの実験では，LIM サーバおよび LA サーバはデータの受け渡しを行わず，人手によって行っていた．つまり，LIM サーバはヒューリスティクスによってリンクオーソリティを参照しているが，今までは LIM サーバが直接参照していたのではなく，間に人手が介入していた．何故なら，サーバを複数台利用していたがそれぞれが連携するような仕組みを用意していなかったからである．しかし公開実験システムでは利用者に対してできるだけ早く情報を提供する必要があり，いちいち人手を介入させるわけにはいかない．

問題 4 に対する工夫: 今までの実験では，各サーバの探索結果は各サーバごとにファイルに保存していた．このため，各サーバ同士が通信する仕組みがなかったのでデータの共有 (リンクオーソリティの参照など) ができなかった．公開実験システムでは各サーバの探索結果を一つのデータベースに格納することによって，各サーバ同士が通信を行うような特別な実装を行わなくてもデータの共有を行えるようにする．それによって LIM サーバの全ての探索を自動化することができる．

問題 5: 監視対象のリンクが大規模になると，リンク切れチェックに要する時間は監視対象に比例して増加する．しかし，公開実験システムでは発生したリンク切れを出来るだけ早く発見したい．そのためにはリンク切れを効率的に発見できるようなリンク切れチェックの方法が必要である．

問題 5 に対する工夫: 今までの実験システムでは，全てのリンクに対して等確率にリンク切れチェックを行っていた．しかし，我々はリンク切れには次のような特徴があると考え，公開実験システムではそれを利用する．

特徴 1: サイトの末端のページはリンク切れになりやすい (サイトの末端とは，図 11 における●のページである)

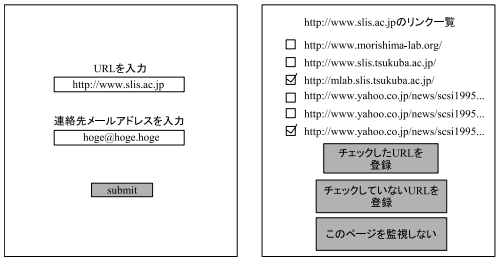

図 9 リンク登録時の画面例
Fig. 9 Example of link registration page

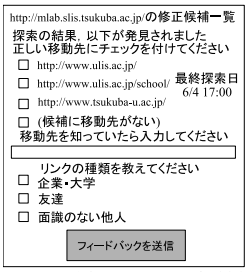

図 10 システムによる新しいリンク候補提供時の画面例
Fig. 10 Example of page to offer new link candidates

特徴 2: サイトの末端のページがリンク切れとなったパスでは，他のページもリンク切れである可能性がある (図 12 においてページ A がリンク切れであるとき，●のページもリンク切れである可能性がある)
これらの特徴から，サイトの末端に近いページほど高確率にリンク切れチェックを行い，リンク切れを発見した際には同一パス上の他の監視対象ページのリンク切れチェックも行う．これによって，効率的にリンク切れを発見することができる．

図 3 に示すように，公開実験システムの主要な構成要素はユーザインタフェース部および LIM サーバ部，LA サーバ部である．
ユーザインタフェース部: 利用者がシステムを利用するためのインタフェース．リンク切れを監視する Web ページの URL の登録，リンク切れ修正候補の提供，フィードバックの送信などを行う機能を提供する．ユーザインターフェース部は PHP によって実装され，Web ブラウザから利用することができる．
LIM サーバ部: 通常はリンク切れが発生していないかどうかをチェックしている．リンク切れを発見すると，リンク先の Web ページの移動先を探索する．その際，リンクオーソリティの情報も利用する．LIM サーバ部は Java によって実装を行う．また，探索は複数のサーバを利用して行う．探索の過程で利用する検索エンジンは，Google[6]，および Yahoo[7] である．
LA サーバ部: 常に各リンク先のページに対するリンクオーソリティを計算する．LA サーバ部は，Java によって実装を行う．また，探索は複数のサーバを利用して行う．探索の過程で利用する検索エンジンは，Google[6]，Alexa[8] である．

5. おわりに

本稿では，Web リンクページ移動先を発見し，リンク切れを自動修正するための公開実験システムの開発について述べた．特に，公開実験システムの実現における問題点と，対処のための工夫について説明した．今後は公開実験を実施し，利用者からのフィードバックやログデータを基にしてシステムの精度の向上をはかる予定である．

[謝辞]

システムに関するご助言をいただきました中溝昌佳氏に感謝致します．ゼミなどでご議論いただきました筑波大学大学院図書館情報メディア研究科の田畑孝一教授，阪口哲男助教授，永森光晴講師に感謝致します．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．

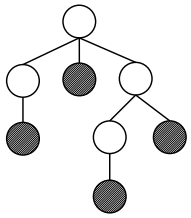

図 11 末端のページの例
Fig. 11 Examples of end pages

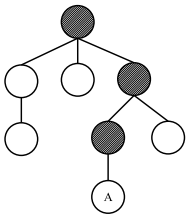

図 12 リンク切れ依存関係の例
Fig. 12 Example of broken-link dependency

[文献]

[1] M.Beynon, A.Flegg: Guaranteeing Hypertext Link Integrity. US Patent Application Publication, US 2005/0021997 A1, Jan, 2005.
[2] M.Beynon, A.Flegg: Hypertext Request Integrity and User Experience. US Patent Application Publication, US 2004/0267726 A1, Dec, 2004.
[3] Akiyoshi Nakamizo, Toshinari Iida, Atsuyuki Morishima, Shigeo Sugimoto, Hiroyuki Kitagawa: A Tool to Compute Reliable Web Links and Its Applications. International Special Workshop on Databases for Next Generation Researchers (SWOD2005), pp.146-149, April 2005.
[4] 中溝昌佳，森嶋厚行，杉本重雄，北川博之, WWW リンク一貫性維持支援システムにおけるリンク切れ自動修復. 日本データベース学会 Letters, Vol.3, No.3, 2004 年 12 月.
[5] Seung-Taek Park, David M.Pennock, C.Lee Giles, Robert Krovetz: Analysis of lexical signatures for improving information persistence on the World Wide Web. ACM Trans. Inf. Syst. 22(4): 504-572 (2004)
[6] Google Web APIs: http://www.google.com/apis/.
[7] Yahoo! Search Web Services: http://developer.yahoo.net/.
[8] Alexa Web Information Service: http://pages.alexa.com/prod_serv/WebInfoService.html.

飯田 敏成　Toshinari IIDA
筑波大学大学院図書館情報メディア研究科博士前期課程在学中．日本データベース学会学生会員．
澤 菜津美　Natsumi SAWA
筑波大学図書館情報専門学群在学中．日本データベース学会学生会員．
森嶋 厚行　Atsuyuki MORISHIMA
筑波大学大学院図書館情報メディア研究科/知的コミュニティ基盤研究センター助教授．1998 年 筑波大学大学院工学研究科修了．博士 (工学)．ACM, IEEE-CS, 情報処理学会，電子情報通信学会，日本データベース学会各正会員．
杉本 重雄　Shigeo SUGIMOTO
筑波大学大学院図書館情報メディア研究科/知的コミュニティ基盤研究センター教授．京都大学大学院工学研究科情報工学専攻博士後期課程修了．工学博士．ACM, IEEE-CS, 情報処理学会, 日本データベース学会各正会員．
北川 博之　Hiroyuki KITAGAWA
筑波大学大学院システム情報工学研究科/計算科学研究センター教授．1980 年東京大学大学院理学系研究科修了．理学博士．ACM, IEEE-CS，情報処理学会，電子情報通信学会，日本ソフトウェア科学会，日本データベース学会各正会員．