

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

Automatic Recovery of Broken Links with a WWW Link-Maintenance System

中溝 昌佳[♡] 森嶋 厚行[◇] 杉本 重雄[▲]
北川 博之[★]

Akiyoshi NAKAMIZO Atsuyuki MORISHIMA
Shigeo SUGIMOTO Hiroyuki KITAGAWA

WWW の特徴の一つに、ページ間に張られたリンクがある。我々は、WWW コンテンツにおけるリンクの一貫性維持を支援するソフトウェアである LIM(Link Integrity Maintenance) サーバを開発している。LIM サーバは、Web ページのリンクを監視し、リンク切れを発見すると自動的に新しいリンク先の候補を探索する。本稿では、LIM サーバが新しいリンク先を探索するアルゴリズム、および実際の Web サイトを用いた実験結果について述べる。

One of the features of the WWW is links among Web pages. We have been developing the LIM (Link Integrity Maintenance) server, a system to support the maintenance of link integrity in the WWW. The LIM server monitors links among Web pages and searches for new links when it finds broken links. The paper explains an algorithm used by the LIM server to search for new links, and gives some experimental results.

1. はじめに

近年、WWW は社会における重要なメディアのひとつとして大きな役割を果たしている。WWW は社会に不可欠なメディアとして発展してきたが、その特徴である分散管理や動的な更新、リンクなどの理由により、コンテンツの一貫性を維持することが非常に難しい。

データベースの分野では一般に、データの一貫性を維持するために、データベースが満たすべき制約（一貫性制約 [1]）を記述するというアプローチがとられる。我々は、この考え方を WWW の文脈に導入する。すなわち、WWW コンテンツで成立すべき制約を記述し、それを用いて一貫性維持を行うための機構を提案する。

我々は、特にリンク切れ等への対応という、リンクの一貫性維持の問題に焦点を当てた研究を行っている [2]。ジョージア工科大の GUV センターにおける調査 [4] では、WWW の利用における重大な問題として、「速度」「欲しい情報が見つからない」と並び、約 6 割のユーザが「リンク切れ」と答えており、リンクの一貫性を維持することは重要な問題と考えられる。しかし、前者の 2 つに関しては様々な対応が行われているが、リンク切れへの対応は現在までほとんど進んでいなかった。リンクの一貫性維持という問題は、分散管理という WWW の性質上、完全に保証することは

[♡] 学生会員 芝浦工業大学大学院 工学研究科
m103198@sic.shibaura-it.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

[▲] 非会員 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@is.tsukuba.ac.jp

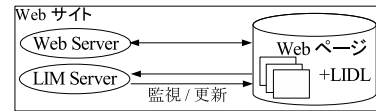


図 1 アーキテクチャ
Fig. 1 Architecture

困難と考えられるが、可能な限り一貫性を満たすよう WWW が自律的にリンクの更新を行う技術の実現を目指す。

我々は [2] でプロジェクトの概要を説明したが、現時点でシステムの一部を設計・実装し、実験を開始している。本稿では、(1) リンク一貫性維持支援システムの概要、(2) 手がかりとなる情報が少ない状態（これについては 3. 章で説明する）でリンク切れを発見したときに代替りのページを探索するための具体的なアルゴリズム、および (3) 実験結果、を説明する。

[関連システム・研究] まず、指定されたページのリンクについて、リンク切れを起こしていないかチェックするリンク切れ発見ツール（リンクチェッカ）がある。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 [3] も行われている。しかし我々の知る限り、WWW コンテンツのリンク一貫性維持の自動化の問題に取り組んだ研究は存在しない。

2. リンク一貫性維持支援システム

リンク一貫性維持支援システムにおける重要な構成要素は、リンクに関する制約を書くための LIDL (Link Integrity Description Language) および、リンクの監視等を行う LIM サーバ (Link Integrity Maintenance Server) の二つである (図 1)。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。本システムの利用例は次のようになる。(1) Web ページのリンクに対し、LIDL を用いて「リンク切れがあってはならない」という制約を記述する。(2) LIM サーバはそのリンクを監視し、制約を満たさないリンク（ここではリンク切れをしているリンク）を検出すると、代替りとなるリンク先ページの候補群を求める。

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 [2] を提供する。

本稿では省略するが、本プロジェクトでは、このアーキテクチャ以外にも様々なアーキテクチャを提案している [2]。

2.1 LIDL

LIDL は、HTML ページもしくは XML ページ (以下 Web ページ) 間のリンクに関する制約を記述するための言語である。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 種類の属性 (isAlive, matches, follows) を定義している。本稿では、紙面の都合上、isAlive 属性と follows 属性について説明する。

[isAlive 属性] isAlive 属性は「リンク切れであってはならない」という制約を表す。

[follows 属性] follows 属性が表す制約の説明を行うために、まず **Link Authority** という概念を導入する。これは、「リンク先の内容が変化したときに、リンクを確実に更新するページ」のことである。

定義. ある Web ページ p は次の条件を満たすとき、別の Web ページ q に関する Link Authority であると言う。(1) p が q へのリンクを持っており、かつ (2) q が q' に移動すると、 p 中の q へのリンクが q' へのリンクに確実に変更される。

例を図 2 に示す。ここで、ある大学 A の研究室の Web ページ q_1 (m.l.s.ac.jp) が存在し、このページは複数のページからリンクされている。このうち p_1 (l.s.ac.jp) はその研究室が所属する学科の研究室一覧ページである。このとき、一般には p_1 は q_1 に関する Link Authority である。従って、 q_1 が q_2 (m.org) に移動した

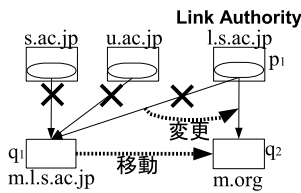


図2 Link Authority
Fig. 2 Link Authority

記号	意味
u	監視対象リンクアドレス (URL)
$P(u)$	u が指すページの内容
c	LIM サーバが $P(u)$ を保持するためのキャッシュ
$link(u, u')$	$P(u)$ が u' を含むとき真となる述語
$LinkingPages(u)$	$\{u_i \mid link(u_i, u)\}$
$domain(u, u')$	u と u' が同じドメインの時, 真となる述語
ru	u のリダイレクト先の URL

図3 記号一覧
Fig. 3 Notation

とき, q_1 へリンクしているページはリンク切れを起こすが, p_1 は q_1 へのリンクを q_2 に張り替えるはずである.

follows では, リンクに対して Link Authority を指定する. 図2の例を用いて説明すると, u.ac.jp が l.s.ac.jp を Link Authority と指定した場合, この制約が満たされなくなるのは Link Authority(l.s.ac.jp) 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 Link Authority(l.s.ac.jp) のページ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, 制約を満たすような新しいリンク先ページの探索を行う.

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 Web 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. Link Authority が指定されていない状況でのリンク探索アルゴリズム

本稿ではリンクに isAlive の制約のみが指定されており, そのリンクが制約を満たさなくなった(すなわち, リンク切れを起こした)場合の, リンク探索アルゴリズムについて説明する. これは, 2.2 節で説明したように, 既存の Web コンテンツに LIM サーバを適用したときに必ず生じる重要なケースである.

説明を簡単にするため, 本稿では, リンク切れの監視対象となるリンクがただ1つである場合について説明する. 図3は本稿で利用する記号をまとめたものである.

詳細は後述するが, 本アルゴリズムは大きく3つの構成要素からなる.

- [1. 定期的なチェック] 定期的に u をチェックし, ページが存在すれば $P(u)$ を LIM サーバのキャッシュ c にコピーする. u がリンク切れであれば, 次の2. と3. を行う.
- [2. 代替ページ候補の収集] u が指していたページの移動先の候補となる Web ページ (URL) のリスト U を作成する.
- [3. 代替ページ候補のランキング] 各候補ページ (の URL) $u_i \in U$ に, ある基準に基づき「移動先のページと考えられる確信度を表すスコア」を割り当ててランキングする.

本アルゴリズムでは, 代替ページ候補の収集およびランキングの双方のために, 以下のヒューリスティクスを利用する. これらのヒューリスティクスは, 著者が「人間はリンク切れに遭遇した場合, 移動先のページをどう探すのか?」について議論をし, さ

```

1. while (true) {
2.   if (isAlive(u)) { // link is alive
3.     if (redirect(u) and link(u, u')) {
4.       ru=u';
5.       c=P(u');
6.     } else {
7.       c=P(u);
8.     }
9.   } else { // broken link
10.    U=LinkSearch(u, ru, c);
11.    sort(U) by score(u_i, u, ru, c);
12.  }
13.  makeInterval();
14. }
    
```

図4 アルゴリズムの最上位レベル
Fig. 4 The Top Level Algorithm

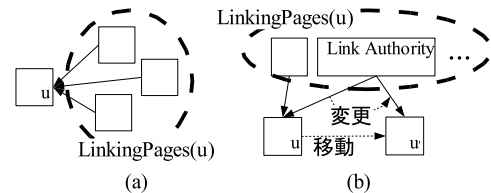


図5 逆引き (a) と, 逆引きを利用した移動先の発見 (b)
Fig. 5 (a) LinkingPage function and (b) its application to Link Discovery

らに予備実験結果の検証などを通じて抽出したものである.

- H1:** 移動先の Web ページは移動前の Web ページと内容が似ている可能性が高い.
- H2:** Web ページが移動するとき, 同じサイト内で移動する可能性が高い.
- H3:** Web ページの移動先がリダイレクトされている場合, 移動先の Web ページの URL がわかる.
- H4:** Web 検索エンジンなどで「逆引き」をすると, Link Authority を発見できる可能性がある.

ここで「逆引き」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. 逆引きの結果は $LinkingPages(u)$ と記述する (図3, 図5(a)). ポイントは, Web の更新と Web 検索エンジンの更新との間に時間差があることである. すなわち, u のページが u' に移動し, Link Authority から u へのリンクも u' に変更されたとする (図5(b)). この時, まだ Web 検索エンジンの更新が行われていないとすると, $LinkingPages(u)$ にはまだ Link Authority が含まれている可能性があり, その場合は Link Authority を通じて移動先の URL u' がわかる.

図4はLIMサーバで動くアルゴリズムの最上位レベルである. 次に, アルゴリズムの各構成要素について説明する.

3.1 定期的なチェック

リンク u を定期的にチェック (Line 2) し, u の状態に応じて次の処理を行う.

u がリンク切れでない場合: $P(u)$ を LIM サーバのキャッシュ領域に保存 (Line 7) する. もし, u がリダイレクトされており, そのリンク先が u' の場合 (Line 3), リダイレクト先の URL として u' を保存 (Line 4) し, $P(u)$ の代わりに $P(u')$ をキャッシュ (Line 5) する.

u がリンク切れの場合: [2. 代替ページ候補の収集] に移動する (Line 10).

3.2 代替ページ候補の収集

ヒューリスティクス H1 ~ H4 を利用して, 次のように代替ページ候補の収集を行い, URL の集合 U を作成する.

(A)H1 を利用: 検索エンジンのキーワード検索による収集
移動先の Web ページは URL が変更されても内容が似ている可能性が高い. そこで, Web 検索エンジンのキーワード検索によって, 類似したページを収集する.

(B)H2 を利用: 同一サイト内ページの探索による収集
Web ロボットを用いて, u と同じサイト内にある Web ページの

$$score(u_i, u, ru, c) = A \times B \times C \times D$$

$$A = sim(P(u_i), c)$$

$$B = \begin{cases} \alpha : domain(u_i, u) \\ 1 : otherwise \end{cases}$$

$$C = \begin{cases} \infty : u_i = ru \\ 1 : otherwise \end{cases}$$

$$D = \begin{cases} \beta : \exists n \in LinkingPages(u)(link(n, u_i)) \\ 1 : otherwise \end{cases}$$

図 6 ヒューリスティクスを利用したスコアの計算
Fig. 6 How to compute Scores based on Heuristics

中からキャッシュ c と類似度の高いページを収集する。

(C)H3 を利用: リダイレクト先ページ ru を利用
リダイレクト先は [1. 定期的なチェック] で既に入手済みなので特にここでは何もしない。

(D)H4 を利用: サーチエンジンの逆引きによる収集
Web サーチエンジンのリンクの逆引き機能を用いて u へのリンクを持つページ $LinkingPages(u)$ を求め、そこからリンクされているページを収集する。

3.3 代替ページ候補のランキング

元の URL を u , LIM サーバが保持する $P(u)$ のキャッシュとリダイレクト情報をそれぞれ c, ru とする (図 3)。このとき、収集した代替ページ候補の各ページ $u_i \in U$ に対し、スコア $score(u_i, u, ru, c)$ を割り当て、スコアの高い順にページを並べる (Line 11)。ヒューリスティクス H1 ~ H4 を考慮し、次の場合にスコアが高くなるようスコアの計算を行う。

(A)H1 の利用: 候補の Web ページ u_i とキャッシュした Web ページ c の類似度が高い。

(B)H2 の利用: u_i と u が同一ドメインにある。

(C)H3 の利用: $u_i = ru$ (リダイレクト先) である。

(D)H4 の利用: $LinkingPages(u)$ に含まれているページから u_i へのリンクが存在する。

具体的な計算式を図 6 に示す。ただし、 $sim(p, q)$ は p と q の類似度である。 α と β はその条件が成立している場合にスコアを上げるための重み係数である。例えば $\alpha = 1.2, \beta = 2$ といったようにそれぞれ 1 より大きな数を割り当てる。

4. 実験

3 章で説明したアルゴリズムを用いて実験を行った。重み係数としては $\alpha = 1.2, \beta = 2$ を用いた。

実験において監視の対象としたのは、筑波大学、芝浦工業大学、北海道大学、東北大学、東京大学、名古屋大学、京都大学、大阪大学、九州大学の計 9 大学のドメインに属するサイトから収集したページに含まれるリンクである。本システムでは、これらのページが持つリンクのうち、外部リンクであるもののみを監視対象とした。ここで、外部リンクとは、リンク元ページとリンク先ページの属するサイトが異なるリンクを指す。外部リンクのみを監視対象とした理由は、リンク切れの可能性が比較的高いと我々が考えたためである。また、本システムが対応している言語は日本語及び英語のみであり、それ以外の言語のページの移動先は探索することができないため、監視の対象から除外した。監視対象としたリンクの数は合計 29971 である。

リンク探索のためには、3.2 節で説明したようにキーワード検索のための検索語の選択を行う必要がある。また、H2 によるサイト内ページの探索方法を決める必要がある。さらに、3.3 節で説明したように、ページ間の類似度を計算する必要がある。以下では、本実験でこれらのために用いた手法を説明する。

キーワード検索の検索語を選ぶ手法: 本実験では、キャッシュ c から、次の手法を用いて 10 個の単語を抽出した。10 は google の検索語の上限である。

(1) キャッシュ c から HTML のタグを取り除く。

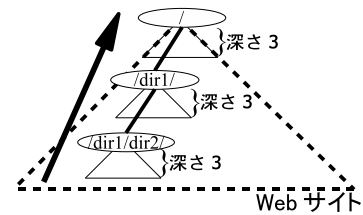


図 7 サイト内探索
Fig. 7 Search inside a Web Site

(2) ChaSen を用いて形態素解析を行う。

(3) 各単語の TF を求める。

(4) TF の上位 20 語の IDF を求める。

(5) これら上位 20 語に対してのみ $TF \cdot IDF$ を求め、上位 10 語を検索語として選ぶ。

ここで問題となるのは、IDF の求め方である。IDF を求めるためには文書集中に単語が出現する文書の数が必要である。しかし、Web ページの場合、文書集合を特定することができないので、IDF を求めることができない。そこで、本実験では IDF を求める単語を Google で検索しその検索にヒットした件数の逆数で IDF を近似する。20 語に対してのみ IDF を求める理由は、Google API の利用に 1 日 1000 回という制限があることを考慮してのことである。

選択された 10 語による検索結果が 3 つ以下の場合、検索語を 1 つ減らして再度検索を行う。減らされる検索語は、 $TF \cdot IDF$ の値が最も小さな語である。この作業は、検索結果が累積 3 つ以上見つかるまで繰り返される。

サイト内ページの探索方法: H2 による探索は次のように行う (図 7)。リンク切れになった URL のアドレスを基に、下位ディレクトリから上位ディレクトリに向かってディレクトリ構造をたどり、それぞれのディレクトリから深さ 3 リンク探索する。

ページ類似度の計算方法: $sim(P(u_i), c)$ の計算には、TF のみを用いて単語の重み付けを行い、各ページを表すベクトル間のコサイン距離を求めることで計算した。単語の重み付けに $TF \cdot IDF$ を用いずに TF のみで行った理由は次の通りである。(1) 本実験では IDF の計算に Google API を用いた検索の結果を利用するが、その利用回数に制限がある。(2) 2 つのページの「そっくり度」を知ることが目的であり、頻度の低い単語が一つ無くなったとしても、他の大部分が同じであれば似ているとしたいため、IDF は考慮しない。

ただし、この類似度の計算手法が我々の目的にとって適切であるというのはあくまで仮説であり、今後検証が必要である。

4.1 実験結果

図 8 に、2004 年 10 月 23 日から 11 月 15 日まで行った実験の結果を示す。監視中、LIM サーバは 68 個のリンク切れに遭遇した。68 個のリンク切れのうち、27 個がページの削除によるものであり、41 個がページの移動によるものであった。ここで、削除されたページとは、探索結果のログ等をもとに、様々な方法で手作業で移動先の探索をおこない、それでも発見できなかったものとした。本システムは、41 個のページ移動に起因するリンク切れのうち、26 個の移動先を発見した。ここで「発見」とは LIM サーバが出力した候補ページリストの上位 3 位以内に正しい移動先が含まれている事である。

4.2 各ヒューリスティクスの貢献度の分析

図 8 の H1, H2, H3, H4 の列は、それぞれのヒューリスティクスが移動先の発見に貢献したリンク切れの個数である。発見には H1, H2, H4 がそれぞれに貢献していることがわかる。リダイレクトがなかったため、H3 は貢献しなかった。

次に、H1 必須, H2 必須, H3 必須, H4 必須の列は、そのヒューリスティクスが存在しなければ発見できなかったリンクの個数で

	総数	削除	移動	移動先発見成功率	H1	H2	H3	H4	H1 必須	H2 必須	H3 必須	H4 必須
リンク切れ	68	27	41	26(63.4%)	15	21	0	8	4	12	0	1

図 8 実験結果とヒューリスティクスの貢献度
Fig. 8 Experimental Results and Contributions of Heuristics

分類	A1	A2	B1	C1	C2
ページ数	3	1	11	2	5

図 9 失敗したページの分類とその個数(のべ)
Fig. 9 Causes of Failures

ある。H1 がなければ移動先を発見できなかったリンク切れは 4 個、同様に H2 については 12 個、H4 については 1 個存在した。H3 については存在しなかった。

4.3 探索に失敗したページの分類

本システムは発生したリンク切れに対する変更先リンクを約 6 割の確率で発見できたが、発見出来なかったリンクがまだ約 4 割存在する。我々は LIM サーバのログを分析し、変更先リンクの候補の発見に失敗したケースの分類を行った。分析は、リンク切れが起こった際、移動先発見に失敗した 15 個について行った。各分類項目に当てはまるページの数(のべ)を図 9 に示し、それぞれの分類項目について次に説明する。

[大分類 A] 現在の手法で本来発見できるはずのもの。

[A1] 監視開始時に既にページが移動していたため、本手法が効果的でなかった。

[A2] google の情報が更新されていなかった。

実験の時点で google はページ移動前の情報を保持しており、移動先と考えられるページの情報を保持していなかった。

[大分類 B] 現在の手法の工夫により発見可能性が高くなると考えられるもの。

[B1] サーチエンジンを利用する際の検索キーワードとして、適切な単語が選択されなかった。

この場合、人手によって検索語を選択し、探索を行った結果、移動先のページを発見することが出来た。

[大分類 C] 新たなヒューリスティクスの追加により発見可能性が高くなると考えられるもの。

[C1] キャッシュの単位を複数ページにする事により、対応可能と思われる。

元ページが「入口」や「Entrance」などという文字列しか存在しないページ、もしくはフレーム構造を表すページであった。

[C2] 「サイトのグループ」の概念を導入することにより、対応可能と思われる。

サイト A を構成するページのうち、一部だけが別のサイト B へ移動したものがあつた。このうち、いくつかはヒューリスティクス H1 などによって移動先を発見できたが、既存のヒューリスティクスだけでは発見できないものが存在した。

4.4 精度を上げるための対応策

以上の分析から、精度をあげるために次の対応策が考えられる。

(1) キーワード検索のための語の選択手法の改良。[B1] のようなページに対しては、検索語の選択アルゴリズムを改良が有効であると考えられる。アルゴリズムの改良の方法としては、HTML のタグによるキーワードの重み付けの導入などが考えられる。

(2) ページ内容キャッシュ方法の工夫。[C1] のような場合のページに対応するには、キャッシュの内容として、リンク先のページだけを保存するのではなく、そこからリンクを張っているいくつかのページもまとめてキャッシュする様な手法が考えられる。

(3) 探索範囲の拡大。[C2] のような場合、同一サイト内探索だけでは、一般にサイト B に移動したページを発見することは出来ない。しかし、元々サイト A に属していたページのうち、サイト B に移動したものが存在することが他のヒューリスティクスによ

てわかっている。そのようなときには、サイト B のコンテンツはサイト A に属していると考えられるので、サイト B とサイト A をある種のまとまりであるとみなし、サイト B を含めた探索を行うことが有効であると考えられる。

5. おわりに

本稿では、WWW のコンテンツにおけるリンク一貫性維持を支援するシステムの概要、変更先リンクの発見アルゴリズム、および実験結果について述べた。それにより、約 6 割のリンク移動に対して、正しく移動先を発見できることを示した。また、その実験結果の検証の結果、さらに精度向上の余地があることがわかった。今後の課題としては、実験結果を基にしたアルゴリズムの改良による探索精度のさらなる向上等が挙げられる。

【謝辞】

本研究において多大なご支援をいただきました有山智洋氏に感謝いたします。また、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、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。

【文献】

- [1] S. Abiteboul, R. Hull, V. Vianu: Foundations of Databases. Addison-Wesley 1995.
- [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] 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. 日本データベース学会 Letters, Vol. 3, No. 1, pp. 45-48, 2004 年 6 月
- [4] Georgia Institute of Technology GVU Center. GVU's 8th WWW User Survey. http://www.gvu.gatech.edu/user_surveys/survey-1997-10/.

中溝 昌佳 Akiyoshi NAKAMIZO

芝浦工業大学大学院工学研究科修士課程在学中。WWW を利用した情報システム, XML, データベースなどに興味を持つ。情報処理学会, 日本データベース学会各学生会員。

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。京都大学工学部情報工学科, 同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, 情報処理学会他会員。

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。