

Web サイト間キーワード共有による自律的關係創出機構の提案

村田 大樹†
Hiroki Murata

小瀬木 浩昭‡
Hiroaki Ozeki

武田 正之†
Masayuki Takeda

1. はじめに

現在、インターネット上には、有益な情報を掲載しているサイトが無数に存在している。しかし、それらのサイトは各々に独立・分散し、必要な情報を抽出することは困難である。サイト管理者としても、自分のサイトと関連性が高いサイトを見つけるには、検索サイトを通じて一つ一つ調べるしかない。また、トラックバックなどによってページ間の関係創出する場合も、手動で行う必要性がある。

そこで本稿では、異なる Web サイト同士においても、自律的に関係性を発見させ、サイト間に繋がりを持たせる、Web サイト間の自律的關係創出機構を提案する。本提案では、利用者が作成したキーワードページによって、サイトをグループ化し、キーワードページを基点としてサイト間の繋がりを見出しやすい環境を提供する。これにより、利用者に興味のある情報の発見を助け、サイト間のコミュニケーションを促す効果も与える。

2. 提案モデル

2.1. 前提

提案モデルにおいて、1つの Web サイトを1つのノードとみなす。ノード同士は全て対等な関係であるネットワーク構成をとる。ノードは公開された Web サーバ上で動作し、動的に HTML を生成するプログラム (CGI) である。また、ノードの保存領域 (ディスク容量) は十分にあるものと仮定する。

2.2. 概要

本論文では、ノード間の関係創出を行うためにキーワードの共有を行う。ノード間でキーワードを伝播し、受信したノードでは、キーワードが日記などの Web ページに存在するか検索する。存在した場合、対象の Web ページはそのキーワードと関係があるとする。ここで関係を表すために、Web ページからキーワードの解説ページへのリンクを生成する。またキーワード解説ページでも、リンク元の Web ページへのリンクを生成する。同様のことを各ノードが自律的に行い、最終的に、同じキーワードが出現した Web ページを、キーワード解説ページを中心としてリンクだけで繋げることができる。本提案モデルを図 1 に示す。

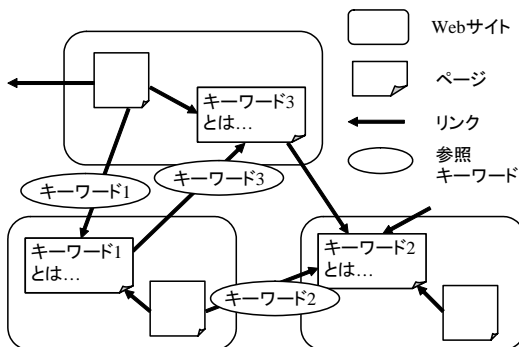


図 1 キーワードによるノード間の関係構築

2.3. 特徴

提案モデルには、以下の特徴がある。

(1) ノードは 3 層で構成

提案モデルは、Web 層、フレームワーク層、ベース層の 3 層に分類される (図 2)。Web 層は Wiki や Blog などの Web 編集システムであり、この層において、日記やキーワードの作成を行う。フレームワーク層は、ノード間でキーワードや他のノードの URI などの情報を伝播させる。ベース層は、フレームワーク層で伝播された情報のデータベースとして動作する。

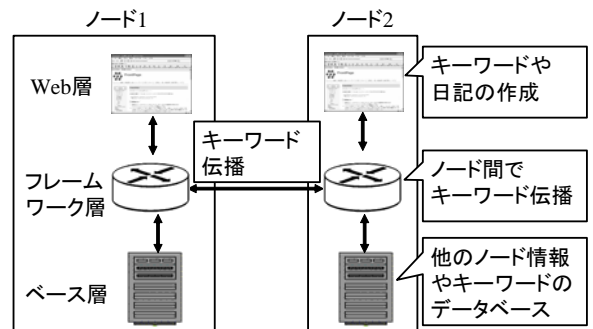


図 2 ノードの概要

(2) ページ単位で関係構築

ノードにおいて、ページごとの情報は断片化されている。Wiki や Blog においても、関係性はノード内で閉じている。はてなダイアリー [1] では、ノード間での関係構築を可能としているが、はてなダイアリーの管理サーバ内に限定されている。一方、提案モデルでは、特定サーバに依存せず、他ノードのページ間で関係を構築でき、断片化された情報の連携を可能とし、情報の発見と再利用を容易にする。

(3) ノードのネットワーク離脱の可能性の低さ

本提案において、構成ノードは Web サイトである。故にネットワークへの参加・離脱はそれほど頻繁ではないと判断できる。よって、構成ノードは他のノード URI を所持しても、多くは有効な URI である。他のノードの URI を知ることで、ノード間で PUSH 型のデータ送信が可能となる。これにより、即時性・網羅性を考慮した、柔軟なデータ配信が可能となる。

(4) 伝播情報の限定

提案モデルでは、ノード間で伝播させるものは、キーワードとその解説ページへの URI だけであり、ページデータの複製の伝播は行わない。なぜなら、解説文の修正や削除が簡単にできるからである。また、転送データを以上のものに限定しているため、転送トラフィックは極めて少量に抑えることができる。

(5) 同一のキーワードを複数のノードで解説可能

有名なキーワードや時事的なキーワードは、複数の利用者によって解説される可能性がある。それに対し提案モデルでは、キーワード解説ページは独立に存在することにし、同一キーワードに複数の解説ページが存在した場合、Web ページからはリンクを複数張ることで解決する。

2.4. 基本プロトコル

本提案のノード間通信は以下の 6 種類存在する。

(1) 新規ノードの参加

新規に参加したいノード A は、サービス既参加のノード Z に対し、キーワードリスト $KList$ とノード URI リスト $NList$ 、ノード Z の公開鍵 $P(Z)$ を取得する。ノード A はベース層にその情報を保存する。そして、 $NList$ 中の任意のノード B, C に対し参加通知を送

†東京理科大学 理工学部 情報科学科
Dept. of Information Sciences, Tokyo University of Science

‡東京理科大学大学院 理工学研究科 情報科学専攻
Graduate School of Science and Technology,
Tokyo University of Science

信する。参加通知は、 $URI(A)$ と $P(A)$ を含んでいる。ノード B, C は受信するとベース層にノード A の情報を保存し、次のノードに送信する。

(2) ノードの脱退

ノード A は $NList$ 中の任意のノード B, C に対し、脱退通知を送信する。脱退通知は $URI(A)$ を含む。ノード B, C は受信すると、ベース層に保存してある $P(A)$ により正当性を検証し、問題なければノード A の情報を削除する。そして次のノードに送信する。

(3) キーワードの追加

ノード A はキーワードと本文で構成される Web ページを生成する。ベース層の $KList$ に新規キーワード $NewKey$ と $URI(NewKey)$, $URI(A)$, の組を保存する。そして $NList$ 中の任意のノード B, C に対し、キーワード追加通知を送信する。キーワード追加通知は $NewKey$ と $URI(NewKey)$ の組を、ノード A の秘密鍵で暗号化したものを含んでいる。ノード B, C はノード A の公開鍵で正当性を確認したら、自身の $KList$ に保存した後に、次のノードに送信する。

(4) キーワードの削除

キーワードの削除は、それに対応するページを持つノードだけが行うことができる。

ノード A は削除したい削除キーワード $RmKey$ と $URI(RmKey)$, $URI(A)$ の組の情報を、自身の $KList$ から削除する。キーワード参照リスト $KRefList$ やキーワード被参照リスト $KRefdList$ にも存在すれば、そこから $RmKey$ と $URI(RmKey)$ の組の情報を削除する。そしてキーワード削除通知を任意のノードに送信する。キーワード削除通知は $RmKey$ と $URI(RmKey)$ の組で構成されている。受信したノードは、ノード A の公開鍵で正当性を確認したら、自身の $KList$ から削除する。また $KRefList$ に存在する場合は、同様に削除する。そして次のノードの送信する。

(5) キーワード参照追加

ノード A の編集した Web ページ $page$ 上に共有キーワード $ShareKey$ が存在した場合、 $page$ 上に $ShareKey$ へのリンクを生成する。自身の $KRefList$ に $ShareKey$ と $URI(page)$ の情報を保存した後、キーワード参照追加通知を、 $ShareKey$ を管理するノードに送信する。キーワード参照追加通知は、 $ShareKey$ と $page$ の組を、ノード A の秘密鍵で暗号化したものを含んでいる。受信したノードは、ノード A の公開鍵で正当性を確認し、自身の $KRefdList$ に、 $ShareKey$ と $URI(page)$ の組の情報を保存する。

(6) キーワード参照削除

ノード A が編集した Web ページ $page$ 上に共有キーワード $ShareKey$ のリンクが存在し、その記述を削除する場合、まずノード A の $KRefList$ から $ShareKey$ と $URI(ShareKey)$ の組の情報を削除する。そして $ShareKey$ を管理するノードに対し、キーワード参照削除通知を送信する。これを受け取ったノードは、自身

の $KRefdList$ から $page$ の情報を削除する。

2.5. 実装

提案モデルの実装を図 3 に示す。上は集中型であるはてなダイアリー、下は分散型である提案手法である。提案手法は、Web 層におけるキーワード解説ページを PukiWiki、日記を tDiary で実装した。また、フレームワーク層とベース層は Perl で実装した。図に示すように、集中型のはてなダイアリーと同様のことが、分散型の提案手法においても可能になっている。

3. 考察と課題

3.1. なりすまし、改ざんの防止

各ノードは専用の公開鍵と秘密鍵の組を生成、保持する。各通知には送信ノードの電子署名を付けることでなりすまし、改ざんを防ぐ。各ノードはノードリストの中に公開鍵を保持していることで独自に検証することが可能である。

3.2. ソーシャルブックマークサービス(SBS)への応用

提案手法の SBS[2]への応用が可能である。具体的には、キーワード追加・削除通知を拡張し、各ノードでブックマークしたページの URI とその内容を表すタグを、通知に含める。伝達方法は同様に行い、他の全てのノードにブックマーク情報が伝達される。その情報を元に、各ノードは SBS として稼動することが可能となる。

3.3. 特定キーワードへのリンクが通常より多い場合

本提案は、各ノードでキーワード解説ページを自由に作ることを許可している。故に、同一キーワードへの解説が数百ページ存在してしまう可能性はある。そのような場合、Web 層でリンクを最新のものに限定することや、JavaScript を用いてポップアップさせるなどして、表示をコンパクトにする手段が考えられる。

3.4. 伝達プロトコル

ノード間でのキーワード転送を行う場合、配信ノードを頂点としたツリー構造で行う方法が適していると考えられる。その場合、既知のノードを元にツリーを作成するが、1 つのノードが何個のノードに対して伝播するのが妥当であるかということは現在検討中である。また、キーワードの追加の際、何ホップまで伝播を許すかということも検討課題である。

3.5. 関連研究

キーワードに基づく分散サーバ型情報探索システムとして Ingrid[3]がある。これは、各ページの特徴を表すデータ (RP) を用いて、RP 間の類似度を元に作られたリンクを利用して、サーバ間の検索を行う。これは検索クエリを Pull することで結果を得るが、本提案は、検索対象のキーワードは生成時に各サーバに Push され、検索もサーバ内に閉じられる点で異なっている。

また、類似サービスとして Blog360[4]がある。これは各 Blog の RSS に対し、形態素解析などでキーワード抽出し、それを用いて Blog 間を繋げるものである。ただし、これは Blog360 のサーバで情報をまとめているだけであり、Blog 間の移動はサポートしていない。また、集中型なので、耐障害性やスケーラビリティなどで分散型の方が優位であると考えられる。

4. まとめ

本稿では、動的に HTML を生成される CGI を設置した Web サーバを対象として、サイト間でページ単位の自律的な関係構築が行える環境を提案し、実装を示した。今後は伝達プロトコルの詳細などを検討してみたい。

参考文献

- [1] はてなダイアリー : <http://d.hatena.ne.jp/>
- [2] SBS : <http://del.icio.us/>
- [3] Ingrid : <http://www.ingrid.org/>
- [4] Blog360 : <http://blog360.jp/>

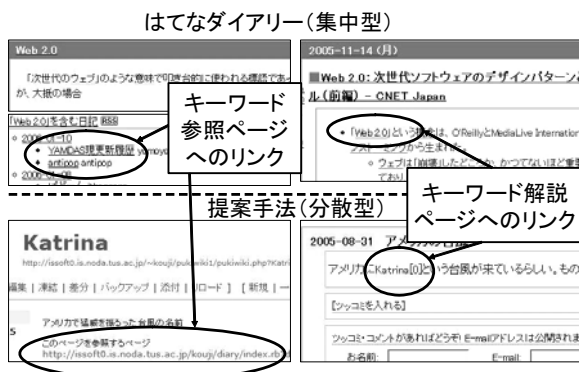


図 3 従来手法との比較