Webページ間自律的関係構築機構の実装とその評価

大矢 健太[†] 小瀬木 浩昭[†] 河木 孝治[††]
鎌田 浩嗣[†] 村田 大樹[††] 武田 正之[††]

近年，Web2.0という言葉が登場している．Web上の記述内容の断片化が進み，個々の利用者がそれらの情報を自由に再編集，カスタマイズ可能であり，Webがあたかもプラットフォームのように振舞うようになることを示す．本稿では，動的にHTMLが生成されるWebサーバを対象として，ページ単位で自律的な関係構築が行える環境を提案する．提案環境の具体化として，複数のWikiサイト間で名前空間を共有する分散協調型のWikiの実装を示す．Wikiの名前空間は通常1つのWikiサイト内で閉じているが，実現したWikiは，複数のWikiサイト間で必要な情報を通知しあうことで名前空間の共有を実現する．論文中，類似の要求を実現する別のシステムの利用傾向の統計情報から，提案環境が大規模化した際の傾向の予測を行い，提案手法がスケーラビリティを備えることを示す．

Autonomous Relation Construction between Web Pages

Kenta Ohya,[†] Hiroaki Ozeki,[†] Kouji Kawaki,[††]
Hiroshi Kamata,[†] Hiroki Murata[††] and Masayuki Takeda[††]

The term of Web2.0 which has been appearing recently has shown following conditions. By the fragmentation of the description on Web, it is possible that users freely reedit and customize that information. Therefore, Web is served as a platform. We propose the environment which can carry out the autonomous interrelation for Web servers such as CGI in which HTML is dynamically created. As an example of our environment, we show distributed Wiki which shares the name space between multiple Wiki sites.

1. はじめに

近年，Web2.0[1]という言葉が登場している．従来の静的なHTMLで構成されたWeb（Web1.0），CGIやCMS（Contents Management System）を積極的に取り入れHTMLの大半が動的に生成され，サイト内連携が密なWeb（Web1.5）に対して，Web2.0とは，コンテンツやサービスがサイトと言う枠を超え，インターネットというインフラの上でシームレスに連動し，新しいコンテンツやサービスを複合的に作り出す，そのような状態を指す言葉である．

従来，未知のWebサイトの発見には主に検索エンジンが用いられてきた．検索エンジンは大量のWebページをクロールし（検索ボットにより収集し）集中型データベースを構築することにより，クロール済のページにおいては網羅性の高い検索が可能である．しかし，検索ボットによるクロール頻度とWebサイトの更新頻度が異なる場合，必ずしも最新の情報を得られないという問題点を抱えている．特に近年，WikiやBlogなどの更新が非常に活発なサイトが増え，クロール頻度と更新頻度の差による問題が顕在化してきている．また，異なるサイト間での連携の仕組みとしてBlogにおけるトラックバック機能があるが，利用者が手動で行わなければならない点が利便性を損ねている．現在あるBlogやCMSは無関係なサイト間でのリンク形成を手動で行う必要があり，サイトを超えてのコンテンツ同士の自律的な関係構築の仕組みがなく，コンテンツのシームレスな連動を行うには十分であるとは言えなかった．また従来の関係構築方法は，1つの単位としてWebサイトが存在し，そのサイト同士に対してリンク形成が行われていたため，的確な情報へコンテンツ参照が行えないという問題があった．

そこで本稿では，動的にHTMLが生成されるWebサーバを対象として，ページ単位で自律的な関係構築が行える環境を提案する．提案手法の位置付けを**図1**に示す．提案手法により，サーバ間のコンテンツ同士の関

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

係を自律的に構築することや，Permalink(Permanent link) の利用により，サイトという枠を超えての，コンテンツやサービス単位での直接的な関係構築が可能となる．さらに伝播する共有情報に求められる速報性と網羅性を考慮することで，全てのサイトが対等な関係にありながら，各サイトへの負荷を一定の範囲に抑制できる．以降，サイト同士が対等な関係にあることを強調する場合に，サイトを**ノード**と表記する．本稿では，サイトを越えてのコンテンツ同士の自律的な関係構築を実現する仕組みを提案する．提案手法は，"キーワード"を介した柔軟性の高い異サイト同士の連携を実現する．

2. 本提案モデル

2.1 提案モデルの概要と構成要素

提案モデルは，構成ノード同士が全て対等な関係であるネットワーク構成をとる．提案モデルでは最大のグループとして，全ノードが参加するグループが最低1つはある．提案モデルの構成要素を**図 2** に，キーワード作成から参照リンク形成までの流れを**図 3** に示す．ここでキーワードとは，公開された Web コンテンツのリンク情報とそれに付随する情報を表す．

提案モデルは，次の3層で構成される．**Web 層:**Web サーバ上の CGI として動作している Wiki，Blog，CMS(Contents Management System) などからのキーワード作成と利用を行う層．**フレームワーク層:** キーワード情報など情報の伝播を行う層．**ベース層:** 一意な URI を保障しフレームワークの動作とデータベースを保管する層．

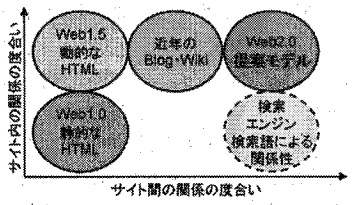

図 1 サイト内とサイト間での関係構築可能性の度合い
Fig. 1 Degree of possibility of relation construction between in site and site

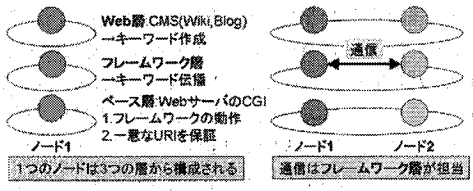

図 2 提案モデルの構成要素
Fig. 2 Component of proposal model

新規キーワード作成の流れ (1)Web 層である Wiki や Blog 内でキーワードを作成する，(2) キーワードの情報はフレームワーク層によって他ノードに伝播される，(3) 伝播された情報はベース層のデータベースに保存される．

参照リンク形成までの流れ (1)Web 層である Wiki や Blog に文章を執筆する，(2) その文章とベース層に保存されているキーワードの情報を比較し，一致するキーワードを抽出する，(3) 抽出されたキーワードを作成したノードにフレームワーク層はキーワードへの言及が行われたことを通知する，(4) 通知を受け取ったノードは言及した文章へのリンクを掲載する．

2.2 本提案モデルの特徴

(1) ページ単位で関係を構築

Wiki や Blog はページごとに情報が断片化されて記録される特徴を持つ．ページ単位で関係を構築することで，断片化された情報の連携が円滑になり，情報の再構成と再利用性を高めることが可能となる．

(2) グループ内の全ノードの URI を保持

新規参加ノードが自身の URI を全ノードに向けて1回だけ送信することで，グループ内の各ノードが，常に全ての他ノードの URI を保持していることを高い確率で保障する．これを用いてプッシュ型で情報伝播をすることで，速報性，網羅性を確保した柔軟な情報伝播が可能となる．

(3) ノード間でキーワードという情報を共有

各ノードは複数の「キーワード」を持っている．キーワードは，公開された Web コンテンツのリンク情報とそれに付随する情報である．

2.3 伝達プロトコル

速報性と網羅性の概念を**図 4** に，網羅・遅延型伝達プロトコルの処理例を**図 5** に，限定・速報型伝達プロトコルの処理例を**図 6** に示す．伝達プロトコルにより，負荷を考慮した柔軟な情報伝播が可能となる．

(a) 網羅性が求められる情報伝播

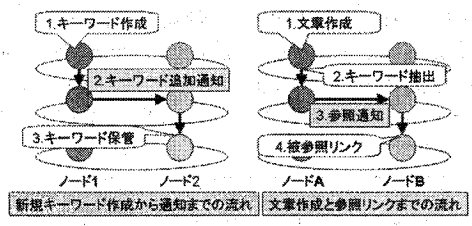

図 3 キーワード作成から参照リンク形成までの流れ
Fig. 3 Flow from keyword making to reference link formation

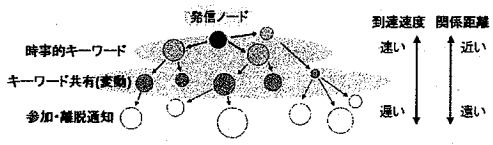

図 4 伝達プロトコルの速報性と網羅性

Fig. 4 News flash and covering of transmission protocol

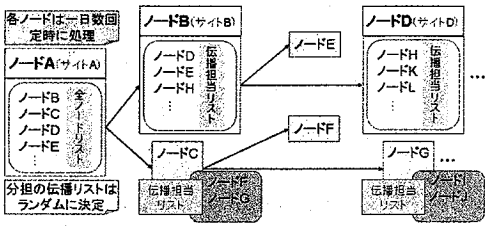

図 5 網羅・遅延型伝達プロトコルの処理例

Fig. 5 Example of processing covering and delayed transmission protocol

グループへのノードの参加・離脱通知などは，速報性よりも，他の全てのノードへ通知するという高い網羅性が求められる．そのため，全ノードへ通知されるよう伝播を依頼していき，依頼された各ノードでは定期的（1日に数回から数日に1回程度の頻度）に処理がされる．これにより，広くゆったりとほぼ全てのノードへ情報が伝播される．

(b) 速報性が求められる情報伝播

時事的なキーワードの追加など要求される速報性の高い通知は，即時に処理・伝播させる代わりに，少ないホップ数かつ，一定期間に1つのノードが作成できる数を制限し，グループ全体にかかる負荷を抑える．

(c) 伝播情報に応じた柔軟な情報伝播

通常のキーワードの追加通知などは，伝播される内容により，処理頻度や経由可能なホップ数を変化させる．速報性が低いキーワードは，伝播処理の頻度を低く抑える代わりに，経由可能なホップ数を多く設定し広範に伝播することを認める．速報性が高いキーワードは，経由可能なホップ数を少なく抑える代わりに，伝播処理の頻度を高く設定する．

3. 提案モデルの Wiki への適用

3.1 概　　要

Wiki[2] とは Web ブラウザから利用できる簡便な Web コンテンツ管理システムである．1つの Wiki サイトは複数の Wiki ページで構成され，1つの Wiki ページは通常一意な URI を持ち，WikiName 形式（2文字以上の大文字が含まれる英単語）のページ名とページ本文で構成される．

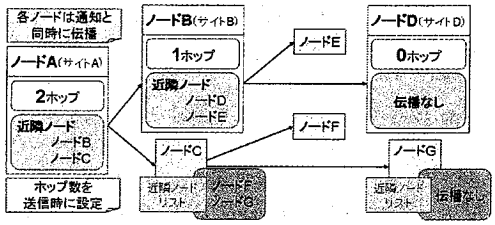

図 6 限定・速報型伝達プロトコルの処理例

Fig. 6 Example of part and news flash type transmission protocol processing

Web サーバ上の CGI で動作する Wiki を1つのノードとみなし，また本提案モデルにおけるキーワードを WikiName とそれに対応する URI とし，これを共有情報とする．これにより，通常1つの Wiki 内での閉じた WikiName の共有を複数の異なる Wiki 間で実現することが可能となる．

3.2 動作の概要

システムの動作概要を**図 7** に示す．なお，簡単のため，ページ名（キーワード）を「●」で表している．(1) ノード A が，キーワード「●」に対応するページを新規作成すると，(2) そのキーワード情報が他のノードへ伝播する．(3) ノード B が●について自身の Wiki サイト内のページ中で言及すると，●に対応するページに自動的にリンク付けされ，また言及通知がノード A に送信され，ノード A 上の●に対応するページに，そのページの言及ページとしてノード B の該当ページへのリンクが掲載される．この仕組みにより Web ページ同士の自律的な関係構築が可能となる．

3.3 実現システムの詳細

3.3.1 表 記 方 法

主体 X の公開鍵を $P(X)$ と表記する．主体 Y のネットワーク上での識別子を $URI(Y)$ と表記する．主体 Z によって作成されたキーワードに対応するページのネットワーク上での識別子を $URI(Z.$ キーワード名) と表記する．同様に，ページのネットワーク上

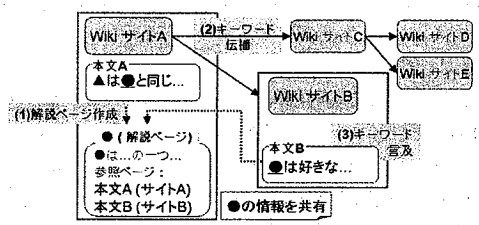

図 7 システムの動作概要

Fig. 7 Outline of operation of system

での識別子を $URI(Z.$ ページ名) と表記する.

3.3.2 各ノードの保持する情報

全てのノードの集合を U, U の部分集合を G とする. G は U のほとんど全ての要素を含むとする. 各ノードはそれぞれ以下の情報を保持する.

(1) $Node$ リスト: G に含まれるノードの URI, そのノードの公開鍵, の組みのリスト. $NList$ と略記する. 自身のノード情報はこのリストへ含めない.

(2) $Keyword$ リスト: 共有情報であるキーワード, そのキーワードに対応するページの URI, その対応するページを持つノードの URI, の組みのリスト. 1つのキーワードに対して複数の対応するページが存在する場合もある. $KList$ と略記する.

(3) $Keyword$ 参照リスト: 自サイトから参照を行っているキーワード, そのキーワードへ参照を行っている自サイトのページの URI, の組みのリスト. 自身が作成したキーワードへの参照もこのリストへ含める. $KRefList$ と略記する.

(4) $Keyword$ 被参照リスト: 参照が行われている自サイトのキーワード, そのキーワードへ参照を行っているページの URI , の組みのリスト. 自サイト内のページから自サイトのキーワードへの参照もこのリストへ含める. $KRefdList$ と略記する.

3.4 プロトコルの概要

プロトコルの概要を**図 8** に示す. なお各通知には, 宛先ノードの URI, 差出ノードの URI, 通知内容が含まれる. また, ノードの参加 (脱退) 通知には, $URI(A), P(A)$ が, キーワードの追加 (削除) 通知には, $Apple, URI(A.Apple), URI(A)$ が, キーワードの参照追加 (削除) 通知には, $Apple, URI(A.$ ページ $a)$ が, 含まれる.

4. 実装・動作例 (実行例)

4.1 処 理 系

情報伝播のフレームワークは Perl5[3] を利用して実装した. Web サーバには Apache1.3.29[4] を利用した. 情報伝播のフレームワークと連携する Wiki は PukiWiki1.4.5[5] を, 連携する Blog は tDiary 2.0.2[6] をベースにした.

4.2 動 作 例

動作例として実行画面のスクリーンショットを**図 9**, **図 10**, **図 11** に示す. 図 9, 図 10 では, サイト外への自動リンク機能が「[0]」と表されている. 1つのキーワードに対応するページが複数存在すれば, [0], [1]... とリンク付けされる. また, JavaScript と CGI を組み合わせて, 本フレームワークに対応していない, 一般の Blog でのキーワード共有のための自動リンクの仕組みを実現した (図 11). これは, キーワードリストを本フレームワーク対応のサーバから取得し, 取得したキーワードリストと, Blog 中に登場する単語の合致を判定し, リンクの自動推薦を実現する.

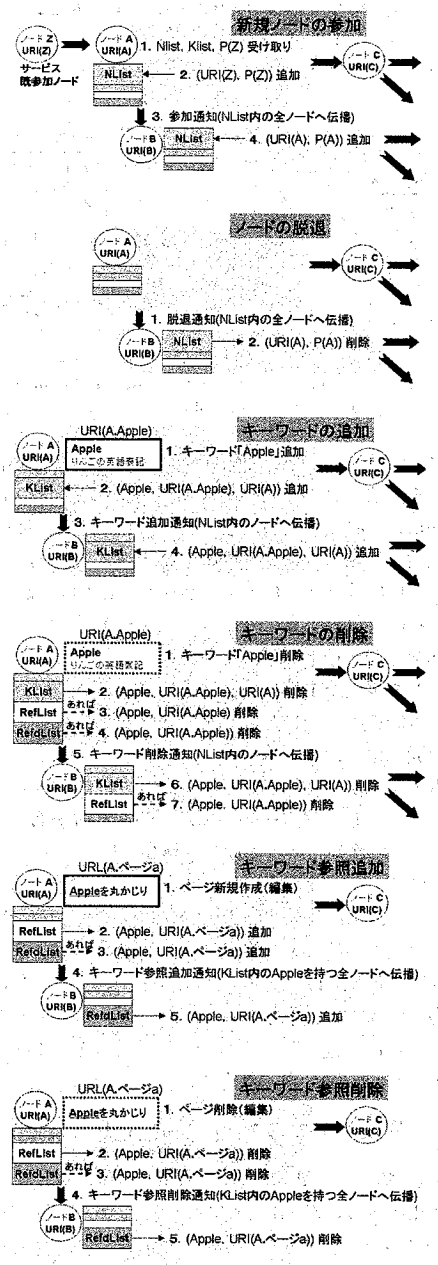

図 8 プロトコルの概要
Fig. 8 Outline of protocol

5. 提案システムにおける負荷の傾向予測

5.1 通信における負荷

各ノード同士で通信が発生するのは，任意のノードの参加・脱退，任意のノードのキーワード作成・削除，自ノードのキーワード参照追加・参照削除，自キーワードの参照追加・削除，の各処理の際である．各処理の発生頻度と処理1回あたりの通信回数の積の和が，グループ全体の通信頻度である．なお，この傾向予測においては伝達プロトコルの使用を考慮していない．

5.2 はてなダイアリーに対する調査

5.2.1 類似のシステムの傾向データ

実現システムの通信頻度の計算におけるパラメータを決定するため，類似の要求をするシステムであるはてなダイアリー[7]について調査した．2005年9月現

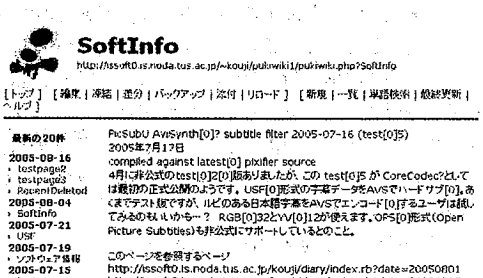

図 9 Wiki 上で動作している様子

Fig. 9 Appearance of operation on Wiki.

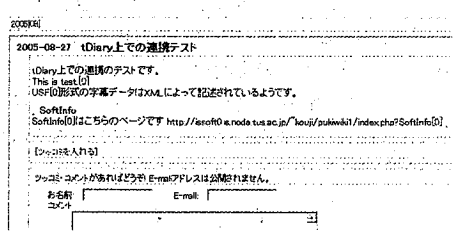

図 10 tDiary 上での連携の様子

Fig. 10 Appearance of cooperation on tDiary

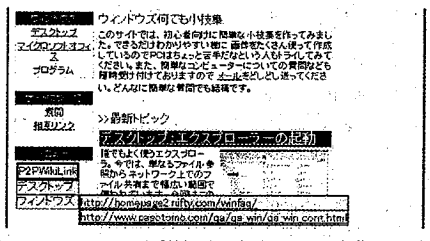

図 11 JavaScript を利用した一般サイトへの自動リンク付け

Fig. 11 Automatic link putting on general site using JavaScript

在公開されているデータによると，日記数は19万件，キーワード数は12万6千個であり，最近16ヶ月で共にほぼ線形に，日記は13万8千件，キーワードは8万4千個増加している．最近更新された日記を50件おきに調べたところ，1記事の平均キーワード言及数は4.05個，1日記の1日の平均記事数は1.31個であった．なお，日記の減少数，記事やキーワードの削除件数は不明だが，増加数に比べれば無視できる程度の数と考えられ，考慮から外した．

5.2.2 負荷傾向の推定

ノード数と通信回数の関係は図12のようになった．1日の平均新規参加ノード数は283.95個（＝最近16ヶ月間の日記増加数／最近16ヶ月間の日数），1日の平均キーワード増加数は172.84個（＝最近16ヶ月間のキーワード増加数／最近16ヶ月間の日数），1日の1ノードの平均キーワード参照数は5.32件（＝1記事の平均キーワード言及数×1日記の1日の平均記事数），1日の1ノードの平均自キーワード被参照数は5.32件（＝1記事の平均キーワード言及数×1日記の1日の平均記事数），である．よって，通信頻度は1日に平均して640.27回（＝1日の平均新規参加ノード数＋1日の平均自キーワード増加数×自分以外のノード数＋1日の平均他キーワード増加数×1＋1日の1ノードの平均キーワード参照数＋1日の1ノードの平均自キーワード被参照数＝$283.95+172.82\times 1/190000\times 189999+172.82\times 189999/190000\times 1+5.32+5.32$）となる．つまり，1ノードあたり2分に1回程度の通信頻度になると推定される．また日記数とキーワード数は共に比例関係にあり，キーワード数と1記事あたりのキーワード言及数も，キーワード数が多いほど記事中に含まれるキーワードも多くなると考えられることから比例関係にあるものとして，ノード数が2の場合と現在の倍の38万の場合も推定した．

多くのキーワードはコンスタントに言及されており，

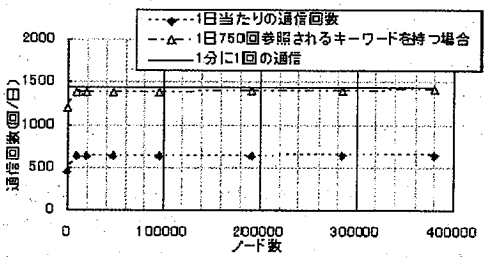

図 12 日記数（ノード数）と通信回数の関係

Fig. 12 Relation between number of diaries(number of node) and communication frequency

例えば，はてなダイアリーの 2005 年 9 月 24 日現在の最も言及されているキーワードである「orz」は 1 日約 750 回の言及数であった．時事的なキーワードとして一時的な言及数の多い「台風」は，過去 3 ヶ月間のうちでの最大が 4500 件の言及数であった．仮に最大 1 日に 5000 回の言及数のあるキーワードを持った場合，通常の負荷と合わせると 1 日に 5640 回，1 分間平均で 4 回程度の通信頻度となるが，この程度の負荷であれば通常の利用の範囲内と考えられるため，実用上は支障はないと思われる．

5.3 負荷要因からの応用システムの条件の推測

調査の結果から，実現システムの負荷のうち大きな要因となるのは，1 日のキーワード増加数である．このパラメータはノード数に比例せず，ほぼ定数となっている．このため実現システム全体の負荷もノード数の増加に対して増えにくく，現実的なノード数において実用上問題ない値となっている．つまり 1 日のキーワードの増加数が一定となることが実現システムを応用する際に実用性を保つための条件と考えられる．例えば，各日記サイトを 1 つのノードとし，日記中の個々の記事のタイトルを共有するシステムは，記事が各ノードである程度の頻度で書かれ続けるため，1 日の記事増加数がノード数に比例するため，問題となる．

6. 考　　察

6.1 なりすまし，改ざんの防止

各ノードは専用の PKI に基づく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ので独自に検証することが可能である．

6.2 ノードによる選別

伝播情報を受け取った各ノードがその内容を確認し，その内容が有益でないと判断した場合には，その伝播を取り止める，ノードによる選別を考える．これにより，各ノードの判断で洗練された有益な情報のみが情報伝播の過程で生き残っていく．また，ノードが伝播を阻止する確率を p，各ノードが通知を受け取るまでの経由ホップ数の平均を h とすると，1 つのノードが受け取る伝播情報は p^{h-1} に減少する．同様の手法で，通知内容の評価を行うこともできる．

6.3 ソーシャル・ブックマークサービスへの応用

ソーシャル・ブックマークサービス (SBS)[8] への提案モデルの適用を考える．各ノードで，ブックマークしたサイトの URI と内容を表すタグを共有情報とすることで分散 SBS が実現できる．

7. 関 連 研 究

本稿の提案モデルにおけるキーワード共有は，はてなダイアリーキーワード[7] と見かけ上似ているが，C/S で完全集中型のそれに対して，提案手法は，分散環境下で，特定のホスティングサーバに依存せず，7) と同様の機能を実現可能である．

ソーシャル・ネットワークサービス (SNS)[9] の技術的な本質は，利用者同士の親近度に応じたアクセス制御にある．分散型 SNS の試みとして Affelio[10] などがあげられるが，提案手法は，分散型 SNS と併用可能で，例えば提案手法を適用した Wiki や Blog にさらに SNS によるアクセス制御を付与することで容易に SNS 化することが可能である．

Ingrid[11] では，Web サイト毎にインデックスを保持し，複数のサイトにまたがった分散検索を実現する手法を提案している．このような従来の検索がプル型で過去の情報を検索対象とするのに対し，提案手法は未来の登場語に対するプッシュ型検索といえる．つまり，「キーワード」の設置により，未来に対する登場語の逐次観察が可能である．未来の登場語に対しては，その都度全文検索する従来手法より，登場したサイトから通知を受ける提案手法の方が効率的である．

8. 結　　論

本稿では，動的に HTML が生成される Web サーバを対象として，ページ単位で自律的な関係構築が行える環境を提案した．提案環境の具体化として，複数の Wiki サイト間で名前空間を共有する Wiki の実装を示し，その有効性を検討し，スケーラビリティを備えることを示した．今後，継続してプログラムを改良し，実用的な基盤となるよう改善していきたい．

参 考 文 献

1) http://en.wikipedia.org/wiki/Web_2.0
2) Wiki Wiki Web. http://c2.com/cgi/wiki?WikiWikiWeb
3) Perl5. http://dev.perl.org/perl5/
4) Apache. http://jakarta.apache.org/
5) PukiWiki. http://pukiwiki.org
6) tDiary. http://www.tdiary.org/
7) はてなダイアリー. http://d.hatena.ne.jp/
8) http://en.wikipedia.org/wiki/Social_bookmarking
9) http://en.wikipedia.org/wiki/Social_network
10) Affelio. http://affelio.jp/
11) Ingrid. http://www.ingrid.org/