WebRelievo: ウェブにおけるリンク構造の発展過程解析システム

WebRelievo: A System for Analyzing the Evolution of Web Link Structure

豊田 正史　　喜連川 優*

Summary. WebRelievo is a system for browsing and analyzing the evolution of the web graph structure based on link analysis. This system enables us to answer historical questions, and to detect changes in topics on the Web. WebRelievo extracts web pages related to a focused page using link analysis, and visualizes the evolution of their relationships with a time series of graphs. This visualization enables us to understand when related pages appeared, and how their relationships have evolved over time. The user can interactively browse those related pages by changing the focused page and by changing layouts of graphs. WebRelievo is implemented on six Japanese web archives crawled from 1999 to 2003.

1　はじめに

ウェブの発展過程は実世界の動向と密接な関係を持つ傾向を強めつつある．例えば，戦争などの重大な事件から，新しい携帯端末の発売などの日常的な事件まで，さまざまなイベントに対応して，関連したウェブページが次々と作成され，重要な情報には多くのページからハイパーリンクが張られていき，多数のユーザが訪れるようになる．多くのウェブページが，生成，更新，および消滅の過程を経て日々変化しており，それに応じてウェブのネットワーク構造も動的に変化を続けている．こうした背景の下，ウェブの発展過程を把握することは，実社会で起きる事象の背景や予兆を探る上で重要な課題となっており，以下のような状況で有用である．

1. ウェブにおけるトピックの履歴に関する質問に答える．
2. 新たな情報の発生を監視または観察し，トレンドを分析することで，市場調査などに応用する．
3. ウェブにおける社会学的な現象とその推移を調査する．

しかし，現状の主要な検索エンジンでは，最新のページをいかに検索するかに焦点が当てられており，こうした過去を紐解くような調査方法はほとんど提供されていない．膨大なアーカイブを基に，ページの内容を過去にさかのぼって見られるサービス [1] も始まっているが，未だその機能は限定的である．

本論文では，定期的に収集したウェブアーカイブから，リンク解析を用いてウェブページ間の関連が発展する過程を可視化するシステム，WebRelievo を提案する．図 1，2 に，本システムの画面スナップショットを示す．WebRelievo は，ユーザの指定したページに対して，各収集時期のアーカイブからページの相関図を表すグラフを抽出して，時系列グラフと呼ばれるグラフの列を作成し，時間に沿って漫画のコマ割のように表示することで，発展過程の閲覧を可能にする．各グラフは，同じページが概ね同じ位置に表示されるよう同期して自動的にレイアウトされるため，各時間でどのような変化が起きたかをグラフを比較しながら容易に把握することができる．また，注目する変化に応じて 2 種類のビューを提供する．現在の実装では，1999 年 8 月から 2003 年 7 月にかけて日本のウェブページを大規模に収集した 6 回分のウェブアーカイブにおける発展過程を閲覧可能である．

以下，第 2 章では関連研究に付いて述べ，第 3 章では WebRelievo の概要について説明する．第 4 章では同期レイアウトのアルゴリズムを解説し，第 5 章では実装について説明する．第 6 章で，まとめと今後の課題を述べる．

2　関連研究

2.1　発展過程の可視化

情報構造の発展過程を可視化する手法については，学術文献の関連可視化，グラフ描画などの分野で様々な研究がなされている．

Chen らは科学論文における著者間の共参照関係の発展過程を可視化する手法を提案している [3]．さらに，[4] において，発展過程の可視化に適した辺のフィルタリング手法に関する考察を行っている．Chen らの手法は，特にレイアウトの同期は行っておらず，変化の把握しやすいフィルタリング手法に焦点を当てている．

Chi らはタイムチューブと呼ばれる手法を提案し，1 つのウェブサイトにおけるページの階層構造，および訪問者のアクセスパターンの発展過程を可視化

* Masashi Toyoda and Masaru Kitsuregawa, 東京大学生産技術研究所 戦略情報融合国際研究センター

した [5]．この手法ではまずサイトの階層構造を，ルートページを中心として，子ページを同心円上に配置したディスクツリー手法で可視化する．さらに各時間についてディスクツリーを作成し，並べて表示することで時間による変化を閲覧可能にする．これに対し WebRelievo は，ウェブサイト間の関連を可視化しており，任意のグラフの発展過程を扱っている．

Diehl らは，変化するグラフの列からアニメーションを作成するための，グラフ描画手法を提案している [7]．Diehl らの手法は，グラフ列に含まれる全グラフの和を取ったスーパーグラフに対して事前に力学的レイアウトを施し，その結果を基に各グラフのレイアウトを決定することで，グラフ変化によるレイアウトの急激な変化を抑えている．また，Erten らは，力学的グラフレイアウトを拡張して，変化するグラフの列を並列配置したり，2 次元レイアウトを 3 次元的に重ねて表示する等，様々な形で描画する枠組を提案しており [9]，最近では学術文献の関係の進化を可視化する事例に応用している．WebRelievo はこれらの研究の延長上にあるが，同期レイアウトを動的に行いながらインタラクティブなグラフ操作を実現しており，さらにノードのクラスタリングを扱っている．

我々はこれまで，リンク解析を用いて抽出した互いに関連しあうページの集合，ウェブコミュニティー，を単位とした発展過程閲覧手法を開発してきた [14]．この手法では，主にコミュニティにおけるメンバー数の変化 (発生，合併，分裂，消滅による) に焦点を当てており，コミュニティの内部やその周辺における構造の発展過程を詳細に解析することはできなかった．WebRelievo は，コミュニティの解析により変化の概要を把握した後，個々のページ単位でより詳細な構造の変化を解析するツールという位置付けにある．

2.2 リンク解析

ウェブ上では，互いに関連するページはページとリンクからなるグラフにおいて近くに存在している傾向がある．この理由としては，同種類のページへのリンクを集めたリンク集が数多く作成されていること，ウェブページの作者は自分のページに関連する情報を持つページにリンクを張る傾向があること，が挙げられる．この特徴を利用して，密に結合されたページを抽出することで互いに関連するページの集合を得ることができる．Kleinberg [11] は，オーソリティおよびハブの概念に基づいて関連ページを計算する手法を提案している．オーソリティーとは，あるトピックについて良質な内容を持つページのことを指し，多くの良いハブからリンクを張られているページと定義される．ハブは，あるトピックに関するリンク集やブックマークページのことを指し，多くの良いオーソリティーにリンクを張っているページと定義される．HITS [11] は，この定義に基づいて，ウェブの部分グラフからオーソリティおよびハブを効率良く抽出するアルゴリズムである．

また Dean らは，シードページに対し関連するページのリストを結果として返す，Companion [6] という関連ページアルゴリズム (RPA) に HITS を応用している．Companion は，シードページ周辺のグラフからオーソリティを抽出する．我々は，Companion の再現率を下げる代わりに精度を上げる手法，Companion–[13]，を提案しており，ウェブ全体での関連ページ相関図の作成に利用している．WebRelievo は Companion–を利用して，ユーザの指定したページの関連ページを算出し，それらのページ間の関連の発展過程を可視化している．現在の実装はオーソリティ，ハブの概念に基づいているが，枠組としては入力ページに対して関連ページの順序付きリストを返すものであれば，どのようなアルゴリズムでも使用できるようになっている．

3 WebRelievo の概要

本章では，WebRelievo の概要を示す．まず本システムの扱う時系列グラフを導入し，次に時系列グラフの可視化手法等について説明する．

3.1 時系列グラフ

まず WebRelievo の扱う時系列グラフを導入する．時系列グラフとは，ある時間間隔をおいて作成された有向グラフの列であり，最初の時間を 1，最後の時間を T としたとき以下のように表される．

$$\{G_t = (V_t, E_t) | 1 \leq t \leq T\}$$

添字 t は各ウェブアーカイブが収集された時期を表す．V_t 中のノードはウェブページを表し，E_t 中の有向辺 (u, v) は，u を関連ページアルゴリズムに入力すると v が上位 N 件以内の関連ページとして出力されることを表している．我々の実装では関連ページアルゴリズムとして Companion– [13] を用いているが，入力ページに対して順序付けされた関連ページリストを返すアルゴリズムであれば何を使用しても良い．時系列グラフにおいて，ノードおよび辺は，各時間において発生および消滅する可能性があり，V_t および E_t の要素数は一定とは限らないことに注意されたい．

さて，各 G_t は非常に大きなグラフとなるため，一度に全てを表示することは不可能である．このため時系列グラフから，ユーザの指定したページ (p) の周辺に焦点を当てた部分時系列グラフを抽出する必要がある．部分時系列グラフは，

$$\{G_t(p) = (V_t(p), E_t(p)) | 1 \leq t \leq T\}$$

と表され，以下の手順で抽出される．

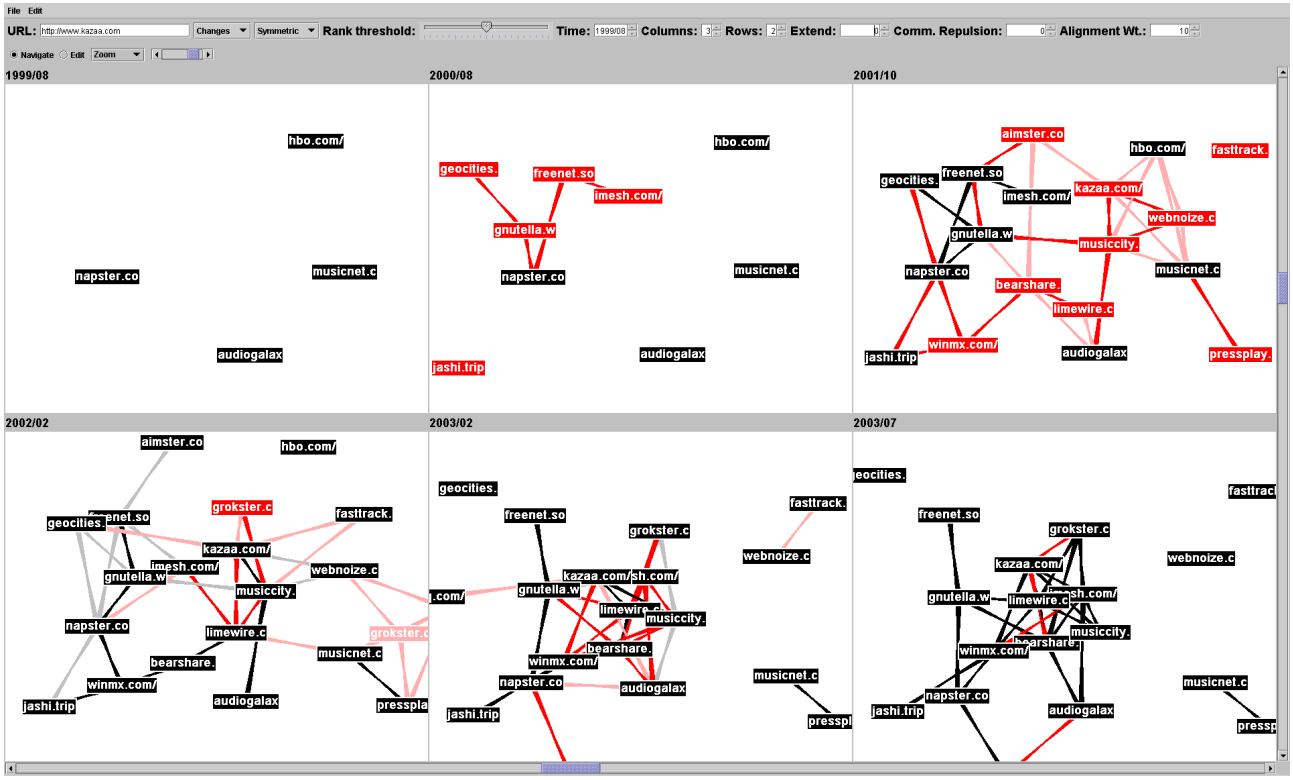

図 1. 差分ビューによる P2P ソフトウェアの変遷

1. 各時間 t において，ページ p に関連するページの集合 $R_t(p)$ を抽出する．つまり $R_t(p)$ は，G_t において p から指されているページの集合である．
2. 各 t において，$V_t(p) = V_t \cap (\cup_{1 \le t \le T} R_t(p))$.
3. 各 t において，$E_t(p) = \{(u, v) \in E_t | u, v \in V_t(p)\}$.

3.2 漫画型および紙芝居型可視化

WebRelievo は，漫画におけるコマ割のように画面を分割して，左から右，上から下の順番で各時間毎に時系列グラフを表示する (図 1 参照)．グラフを並べて表示することで，隣り合うグラフの比較を容易にしている．画面分割における行数，列数はユーザが自由に調整できる．特に，行数および列数を共に 1 にして開始時間を変化させると，紙芝居の様にグラフの変化を見ることになる．また，ユーザは表示を開始する時間を変化させることで，表示されるグラフの列を時間軸に沿ってスライドさせることができる．

3.3 差分ビューおよびクラスタビュー

WebRelievo は，ノード，辺の発生と消滅の表現に重点をおいた差分ビュー，および，辺で密に結合されたノードのクラスタがどのように変化したかを重点的に表現するクラスタビュー，の 2 通りのビューを提供する．

差分ビュー (図 1) は，ユーザが各時間のグラフから前後における変化をある程度把握できるよう，各ノードおよび各辺の発生および消滅を色で表現する[1]．各 $G_t(p)$ におけるノードおよび辺は，$G_{t-1}(p)$, $G_{t+1}(p)$ に存在するかしないかによって，以下のように色分けされる．

- 黒色: $G_{t-1}(p)$ および $G_{t+1}(p)$ に存在する．黒で安定性を示す．
- 赤色: $G_{t-1}(p)$ に存在せず，$G_{t+1}(p)$ には存在する．t における発生を赤で示す．
- 灰色: $G_{t-1}(p)$ には存在するが，$G_{t+1}(p)$ には存在しない．$t+1$ における消滅を灰色で表現する．
- 薄赤色: $G_{t-1}(p)$ に存在せず，$G_{t+1}(p)$ にも存在しない．t において発生し，すぐ消滅することを薄赤色で示す．

差分ビューは，新規に現れたページや関連の把握に有用である．図 1 では，差分ビューを用いて，P2P ファイル共有ソフトの変遷を閲覧している．入力としては Kazaa のホームページ (www.kazaa.com) を

[1] カラーの図が CD-ROM に掲載されているので参照されたい．

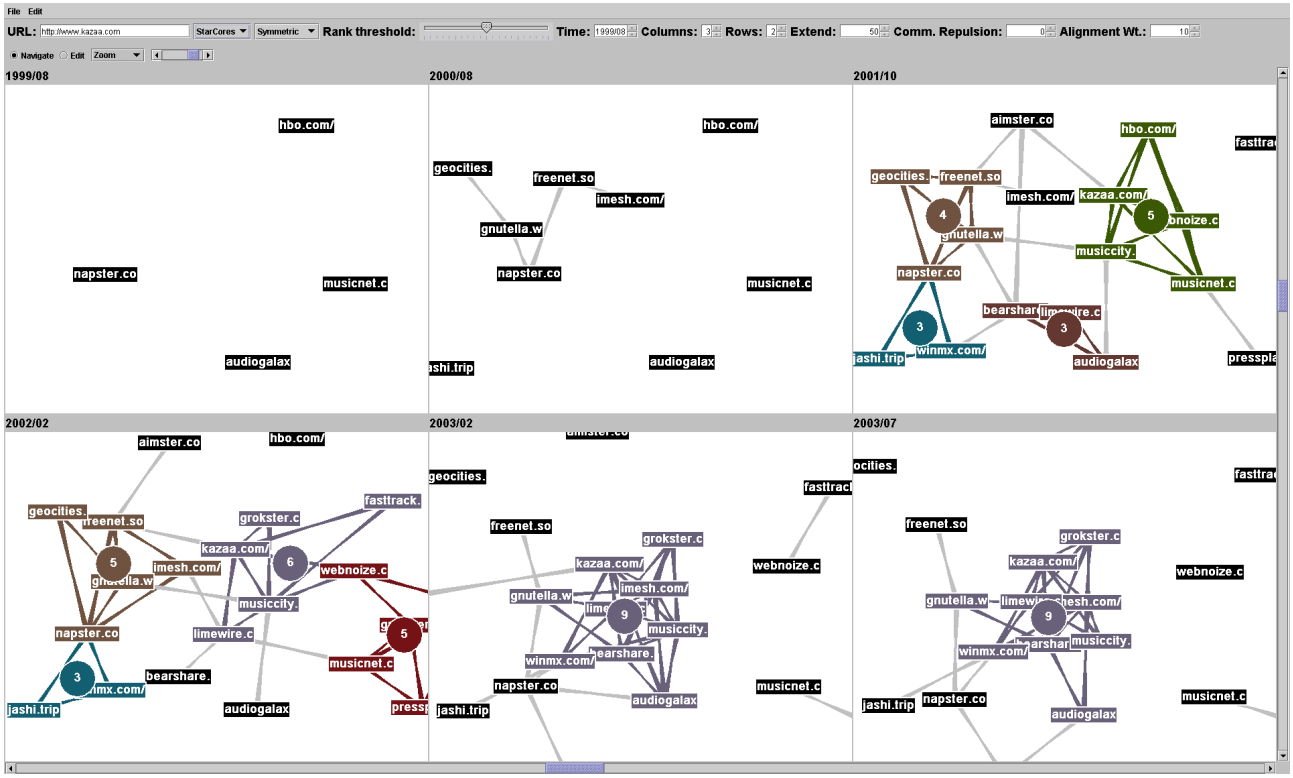

図 2. クラスタビューによる P2P ソフトウェアの変遷

与えている．1999 年に Napster が現れ，2000 年には Gnutella，Freenet 等が新しく現れている．2001 年にはさらに，Kazaa，Bearshare 等が現れ，その後は次第に関連が密になっていく様子が分かる．

クラスタビュー (図 2) では，ノード単体の変化ではなく，ノードの意味的な塊の変化を見ることができる．このビューでは，辺で密に結合されたノードの集合をクラスタとして抽出し，各クラスタに別な色を付けることで，クラスタの変化を表現する．WebRelievo におけるクラスタの抽出方法は，以下の通りである．

1. 各 $G_t(p)$ において有向辺で相互に結合されている部分 (辺 (u, v)，(v, u) が共に存在する) のみを抽出し，相互結合による無向グラフを作成する．図 1，2 では分かり易さのため，この無向グラフを表示しているが，元の有向グラフを表示することも可能である．
2. 上記無向グラフにおいて，辺で結合されたノードの 3 角形をすべて抽出する．
3. 辺を共有する 3 角形同士が同じクラスタに属するように分類する．この時点で各クラスタは，1-連結以上 (点，辺両方において) であることが保証される．ただし，1-連結以上の部分グラフ全てを抽出している訳ではない．
4. 2 個以上のクラスタに属するノードは，そのノードから最も多くの辺がのびているクラスタに属するものとする．これにより，クラスタは互いに素なノードの集合となる．

この方法は，我々のこれまでの研究 [13] に基づいており，抽出されたクラスタ内のページは，内容が類似している可能性が高いことが分かっている．クラスタリングの手法としては，極大クリークを列挙するなど他の方法を使用しても良く，以降の可視化手法には影響しない．ただし互いに素なクラスタリング手法を使用する必要がある．

クラスタを見分け易くするため，各ノードおよび辺には，クラスタ毎に異なる色を割り当て，クラスタの重心にはそのクラスタに含まれるノード数を表す丸いノードを表示する．また，時系列的な追跡を容易にするため，各クラスタについて，隣り合うグラフから最も多くのノードを共有するクラスタを探し出し，同じ色を割り当てている．

クラスタビューでは，クラスタの合併や分裂などの挙動を把握することが可能となる．図 2 は，図 1 と同じ時系列グラフをクラスタビューで表示したものである．2001 年の時点で，Napster，Gnutella など初期からあるソフトがクラスタを形成し，この時点で現れた Kazaa や Bareshare などは新規のクラスタを形成している．2003 年の 2 月にはこれら新旧のソフト群が合併して 1 つのクラスタとなっている．さらに 2003 年 2 月，7 月のグラフからは，Napster，

Gnutella など古いソフトの影響力の低下も見て取れる．Napster は 2003 年 2 月にクラスタから分裂しており，Gnutella もクラスタ内でのリンク数を減らしていることが分かる．

4 同期レイアウトアルゴリズム

WebRelievo におけるグラフレイアウトには，力学的アプローチ [8, 10] を時系列グラフ間の位置調整を行うように修正したアルゴリズムを用いている．すなわち，各 $G_t(p)$ を力学的アプローチでレイアウトしながら，時系列方向にノードの位置調整を行う．ユーザは対話的にノードをドラッグすることができ，その場合にもノードの位置が自動的に同期される．以下にその詳細を述べる．

4.1 力学的アプローチ

力学的アプローチは，グラフにおけるノードを質点，辺をばねとする力学モデルのシミュレーションを用いてグラフをレイアウトする手法である．辺でつながれた 2 点の間には引力 F_a が働き，全ての 2 点の間には斥力 F_r が働く．WebRelievo では点 u, v 間に働く引力および斥力として以下の定義を用いている．それぞれの力は u，v 間のユークリッド距離 d を引数とする関数で表される．

$$F_a(d) = d^2/c_1^2, \quad F_r(d) = -c_1/d$$

ただし，c_1 は 2 点間の望ましい距離を表す定数である．辺で結ばれた 2 点間の距離が c_1 となったときに引力と斥力がつりあうようになっている．我々の実装では $c_1 = 100$ をデフォルト値として用いている．

この定義は，実装に用いている TouchGraph[12] に基づいており，[8, 10] における定義とは異なる．Eades の定義 [8] に変更することは可能だが，Fruchterman の定義 [10] は斥力が強く設定されているため振動が激しくなり対話的なレイアウトには向いていないことが経験上分かっている．

4.2 グラフ間の位置合わせ

時系列グラフの差分を理解しやすくするため，隣り合うグラフの間では同じノードができるだけ同じ位置にあることが望ましい．これを実現するため，引力，斥力に加えて隣り合うグラフ間で位置合わせを行う力を各点に与える．ノード $u_t \in G_t$ には，$u_{t-1} \in G_{t-1}$ および $u_{t+1} \in G_{t+1}$ に近付く方向の力，F_{t-1} および F_{t+1} が加えられる．u_t が隣のグラフに存在しない場合にはこれらの力は 0 となる．定義は，以下の通りである．F_{t-1} は，u_t と u_{t-1} 間の距離 d_{t-1} を引数とする関数で定義され，F_{t+1} も同様に定義される．

$$F_{t-1}(d_{t-1}) = d_{t-1}/c_2, \quad F_{t+1}(d_{t+1}) = d_{t+1}/c_2$$

ここで，c_2 は位置合わせの力の強さを調整する定数である．$c_2 = 1$ のとき，位置合わせの力は最大になり，u は各グラフでほぼ同位置に表示される．c_2 を大きくすると，各グラフでの引力，斥力の方が相対的に強くなり，位置合わせの力は弱まる．

直感的には，位置合わせの力を最大にすると最も理解しやすいレイアウトになると予想される．しかし，位置合わせを厳密に行うと，最初の時間における位置関係が，最後の時間における位置関係にも影響するため (逆も同様)，構造の変化が大きいときにはかえってグラフの形をいびつにしてしまう．このため，時間の経過と共に緩やかに位置の変更を許すほうがレイアウトは見やすくなる．我々の実装では $c_2 = 10$ をデフォルト値として使用している．ただし，この値はユーザが調整できる様になっている．

4.3 時系列的変化のレイアウトへの反映

クラスタビューにおいては，ノードおよび辺の色を所属するクラスタを表現するために使用しているため，発生および消滅が分かり難くなっている．このうち辺の発生または消滅のどちらかについては，辺の長さを調節することである程度表現することが可能である．具体的には，$t-1$ に存在しない辺を t で長くするか (発生の表現)，$t+1$ に存在しない辺を t で長くする (消滅の表現)．両方を同時に行うと，発生と消滅の区別がつかなくなるため，これを行う際には，どちらかを選択する必要がある．図 2 では，消滅を表現することを選択している．例えば，2001 年 10 月のグラフでは，右上の hbo.com が他のメンバーからは離れて配置され，次の時間に切り離されることが分かるようになっている．

アルゴリズム上は，発生または消滅した辺について，引力 F_a を以下のように変更する．

$$F'_a = d^2/(c_1 + c_3)^2$$

ここで c_3 は，発生または消滅した辺をどれだけ長くするかを表す定数である．WebRelievo では，デフォルト値として $c_3 = 50$ を使用しているが，ユーザが調節することも可能である．

本アルゴリズムの 1 イテレーション当たりの計算量は，各グラフにおけるノード数の最大値を n とすると，引力，斥力の計算に $O(Tn^2)$，位置合わせに $O(Tn)$ かかるため，全体では $O(Tn^2)$ のオーダーとなる．

5 実装

WebRelievo は，1999 年 8 月から 2003 年 7 月にかけて大規模に収集した 6 回分のウェブアーカイブを基にしている．表 1 に詳細を示す．収集ページ数はロボットを使用して実際に収集した HTML ファイルの数を示す．収集した各アーカイブからは，URL とリンクからなるウェブグラフを抽出し，リンク解

収集時期	収集ページ数	総 URL 数	総リンク数
1999/7–8	17M	34M	120M
2000/7–8	17M	32M	112M
2001/10	40M	76M	331M
2002/2	45M	84M	375M
2003/2	66M	384M	1058M
2003/7	98M	601M	1587M

表 1. ウェブアーカイブの詳細

析に使用するデータベースを作成する．このウェブグラフにはアーカイブ内のページの URL のみではなく，それらのページからリンクされているアーカイブの外側の URL も含まれる．結果として com や edu ドメインなどの URL もグラフに含まれることになる．表 1 には，グラフに含まれる URL の総数とリンクの総数も示されている．

リンク解析を効率的に行うためこれらのウェブグラフは，指定された URL から隣接 URL を検索できるメインメモリデータベースとして実装してある．実装方式は，connectivity server [2] と同様である．さらに，被リンク数の多い主要なページについては，関連ページの計算をあらかじめ行っておき，関連ページのデータベースを作制してある．これで時系列グラフを高速に取り出すことが可能になる．

時系列グラフ表示部分については，Java を用いて実装しており，自動グラフ描画ソフトウェアの TouchGraph[12] をベースにグラフの同期に関する変更を加えている．Pentium III (1.7GHz)，1GB RAM のマシン上で 6 つのグラフを同時に表示させると，1 グラフあたり 50 ノード程度であれば，秒間 5 フレーム程度のアニメーション表示が可能である．

6 まとめと今後の課題

定期的に収集したウェブアーカイブから，リンク解析を用いてウェブページ間の関連が発展する過程を可視化し閲覧可能にするシステム，WebRelievo を提案した．WebRelievo は発展するグラフの様子を時系列的に並べて，同期しながらレイアウトすることで，発展過程の解析を容易にしている．

本論文では，時系列グラフの可視化として，差分ビューおよび，クラスタビューという 2 通りの発展過程表現手法を提案した．差分ビューは，ノードおよび辺の発生および消滅の観察に適しており，クラスタビューは，ページの意味的な塊が合併，分裂する様子を観察するのに適している．また，同期レイアウトに関してアルゴリズムの詳細を述べ，レイアウトパラメタについては，位置合わせ，および，グラフの時系列変化の反映に関する調整が重要であることを示した．

今後は，アーカイブの収集間隔を縮めながらより詳細な解析を行えるツールを構築していく予定である．収集間隔が，1 週間毎程度まで縮まると，より連続的な変化を表現する手法を開発する必要がある．

参考文献

[1] Wayback Machine, The Internet Archive. http://www.archive.org/.

[2] K. Bharat, A. Broder, M. Henzinger, P. Kumar, and S. Venkatasubramanian. The Connectivity Server: fast access to linkage information on the Web. In *Proceedings of the 7th International World Wide Web Conference*, pp. 14–18, 1998.

[3] C. Chen and L. Carr. Visualizing the evolution of a subject domain: A case study. In D. Ebert, M. Gross, and B. Hamann eds., *IEEE Visualization '99*, pp. 449–452, San Francisco, 1999.

[4] C. Chen and S. Morris. Visualizing Evolving Networks: Minimum Spanning Trees versus Pathfinder Networks. In *IEEE Visualization 2003*, pp. 67–74, 2003.

[5] E. H. Chi, J. Pitkow, J. D. Mackinlay, P. Pirolli, R. Gossweiler, and S. K. Card. Visualizing the Evolution of Web Ecologies. In *Proceedings of ACM SIGCHI '98*, pp. 400–407, 1998.

[6] J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. In *Proceedings of the 8th World-Wide Web Conference*, pp. 389–401, 1999.

[7] S. Diehl and C. Görg. Graphs, They are Changing. In *The 10th Symposium on Graph Drawing*, pp. 23–30, 2002.

[8] P. Eades. A heuristic for graph drawing. *Congressus Numerantium, 42*, pp. 149–160, 1984.

[9] C. Erten, S. G. Kobourov, V. Le, and A. Navabi. Simultaneous Graph Drawing: Layout Algorithms and Visualization Schemes. In *The 11th Symposium on Graph Drawing*, pp. 437–449, 2003.

[10] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

[11] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pp. 668–677, 1998.

[12] A. Shapiro. Touchgraph. http://www.touchgraph.com/.

[13] M. Toyoda and M. Kitsuregawa. Creating a Web Community Chart for Navigating Related Communities. In *Conference Proceedings of Hypertext 2001*, pp. 103–112, 2001.

[14] M. Toyoda and M. Kitsuregawa. Extracting evolution of web communities from a series of web archives. In *Proceedings of the Fourteenth Conference on Hypertext and Hypermedia (Hypertext 03)*, pp. 28–37, August 2003.