5 Web更新モニタリング

山田　誠二　国立情報学研究所
seiji@nii.ac.jp

Webで最新情報をチェック

インターネットの普及に伴い，今や多くの個人や組織が自分のWebページを持つに至るほど一般に普及したWebだが，その最も役に立つ特長の１つとして，Webページが頻繁に更新されることによる「情報の新鮮さ」が挙げられる．株価，為替レート，ニュース，新聞，天気予報など数え上げればきりがないほど，時々刻々と更新され最新の情報を提供してくれるWebサイトが多く存在する．確かにこれらのWebページで提供される最新情報は，ユーザにとって非常に役に立つものであるが，最新情報がアップロードされればできるだけ早くそれを知る必要があるため，ユーザは常時それらのWebページをモニタリングしなければならない．しかし，2，3のWebページならまだしも，監視すべきWebページが多くなると，とてもいちいちチェックをすることはできない．そのため，せっかくの最新情報を見逃してしまう経験をされた方も少なくないであろう．このようなWebページの監視を「Web更新モニタリング」と呼ぶ．

Web更新モニタリングは，Webでの情報収集にとって非常に重要な問題であり，これまでもいくつものWeb更新モニタリングシステムが研究開発されてきた．本稿では，これまでの代表的なシステムの紹介，そこでの問題点と先端的なWeb更新モニタリングの研究例，そして今後の課題について説明していく．

Web更新モニタリング

代表的なWeb更新モニタリングシステム

Web更新モニタリングの基本機能である，Webページの情報の更新をチェックし，その更新部分を検出することは，基本的にはプログラムで実現できるため，これまでWeb更新モニタリングのアプリケーションがいくつか開発されてきた．まずは，これまでのWeb更新モニタリングを理解するために，現在広く使われているシステムとして，電子メールで更新を通知するフリーウェアであるChangeDetectと商用アプリケーションであるWebSpectorを紹介する．

ChangeDetect[2]は，NetMindから発展したアプリケーションであり，サーバが一括してWebページの更新をモニタ，管理する．ChangeDetectは，ユーザが更新を通知して欲しいURLと通知メールを送る宛先のメールアドレスをサーバに登録する．この登録は，**図-1**のようなインタフェースで行われる．そして，サーバは登録されたURLのWebページをモニタして，更新があるとメールでユーザに通知する．モニタリングの対象としているのは，基本的に１つのWebページであり，複数のWebページを登録することはできるが，Webサイト全体をモニタすることはできない．また，フレームを使ったページや動的にコンテンツを生成するページには対応していない．

また，ChangeDetectは，意味のない更新通知を避ける工夫がされている．たとえば，ユーザ登録（有料）により拡張される機能により，更新中に含まれるべきキーワードの設定が可能で，複数のキーワードをAND，OR，フレーズで設定できる．設定後は，その条件を満たす更

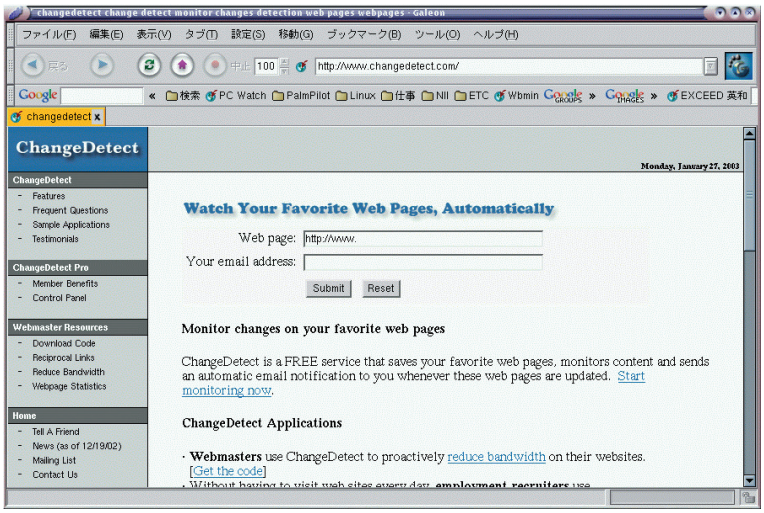


図-1 ChangeDetectのインタフェース[2)]

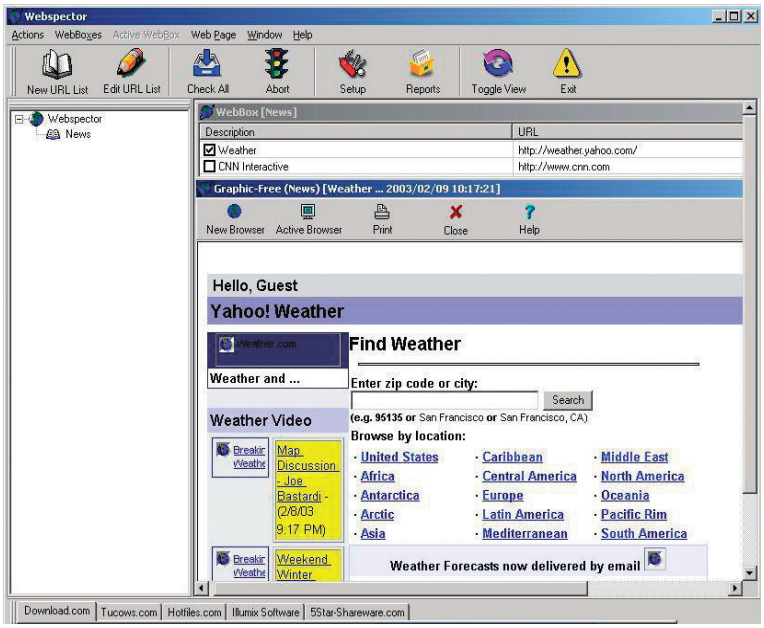


図-2 WebSpectorのインタフェース[5)]

新のみが通知される．

メールによる更新通知メールの一部を以下に示す．この通知メールから分かるように，単に更新されたURLを知らせてくれるだけであり，更新された個所の提示等はできない．

> *The ChangeDetect monitor has determined that the following web page has changed:*
> *http://research.nii.ac.jp/~seiji/index-e.html*

WebSpector™[5)]も，ユーザが登録したURLのWebページをモニタリングする．基本的には，クライアントアプリケーションがユーザのPCに常駐して，モニタリングを行う．更新が検出された場合は，eメールで通知をする．また，**図-2**のようなインタフェースが提供されており，これを通して，モニタリングしたいURLの登録，更新されたページのURLのリスト表示，そしてそのリストのURLをクリックすることにより，ブラウザでそのページを表示するなどの機能がある．

さらに，WebSpectorでは，更新のモニタリングをいつ行うかのスケジュールを設定可能で，また，更新された部分のハイライト表示，ユーザが指定したキーワードのハイライト表示などにより，可視性を向上させている．さらに，検出すべき更新の条件として，差分のサイズのしきい値を指定して，それ以上の大きさの更新があった場合のみ通知させることができる．

Webページ差分の抽出

Web更新モニタリングには，Webページが更新されたか否かを判定し，また更新された部分を抽出する技術が必要である．ここでは，Web更新モニタリングにおいて，Webページ間の差分の検出に広く使われている技術である最長共通部分系列LCSとそれを拡張したHCSについて説明する．

LCSは，直観的には，2つの記号列間の最大共通部分系列を意味するが，より厳密な定義を次に示す．記号系列$\sigma=\sigma_1\sigma_2\ldots\sigma_p$に対し，非連続であってもよいが，順序関係を保持している系列$\tau=\sigma_{i_1}\sigma_{i_2}\sigma_{i_l}(i_1<i_2<\cdots<i_l)$を$\sigma$の部分系列と呼ぶ．そして，2つの記号系列$\alpha, \beta$において，$\gamma$が$\alpha$，$\beta$両方の部分系列であるとき，$\gamma$を$\alpha$，$\beta$の共通部分系列とする．2つの記号系列$\alpha, \beta$における最長共通部分系列LCS（Longest Common Subsequence）とは，α，β間の最も長い共通部分系列を意味する．

2つのWebページの差分を抽出する場合は，まずそれらのWebページファイルをHTMLパーザにかけて，タグ構造を構文解析し，2つの記号系列を生成する．この構文解析では，基本的には，対応するタグに挟まれたプレーンテキストが1つの記号と見なされて，記号系列が生成される．ただし，表（<table>タグ）やリスト（<ul>, <ol>, タグ）などは，直観的な解釈に合うように特別の処理がされ，行やアイテムが1つの記号として抽出される．そして，得られた2つの記号系列からLCSを抽出する．古いページとLCSの差分が削除された部分であり，新しいページとLCSの差分が追加された部分となる．また，削除された部分と追加された部分の対応をとることで，古いページのどこが新しいページのどこに更新されたかを説明できる．なお，このLCSの検出は，UNIXのdiffコマンドで使われている．diffでは，行を単位とした記号系列においてLCSを抽出し，LCS以外を差分として取り出す．後述するWebBeholderでは，このLCSを基にして，更新部分の検出を実現している．

HCS（Heaviest Common Subsequence）は，LCSを拡張したもので，マッチした記号とその位置に基づいた重みを導入し，その重みが最大になるように拡張された共通部分系列である．近似的なマッチングや行の長さを考慮することができる．代表的な差分エンジンであるTopBlend[3]では，HTMLファイルを構文解析して，HCSに基づく差分を抽出することで，より精度の高い更新部分の特定を実現している．

問題点

ここまで，Webモニタリングシステムとその機能を見てきたが，実際に利用してみるといくつかが不満な点があることに気付く．その代表的なものを以下に挙げる．

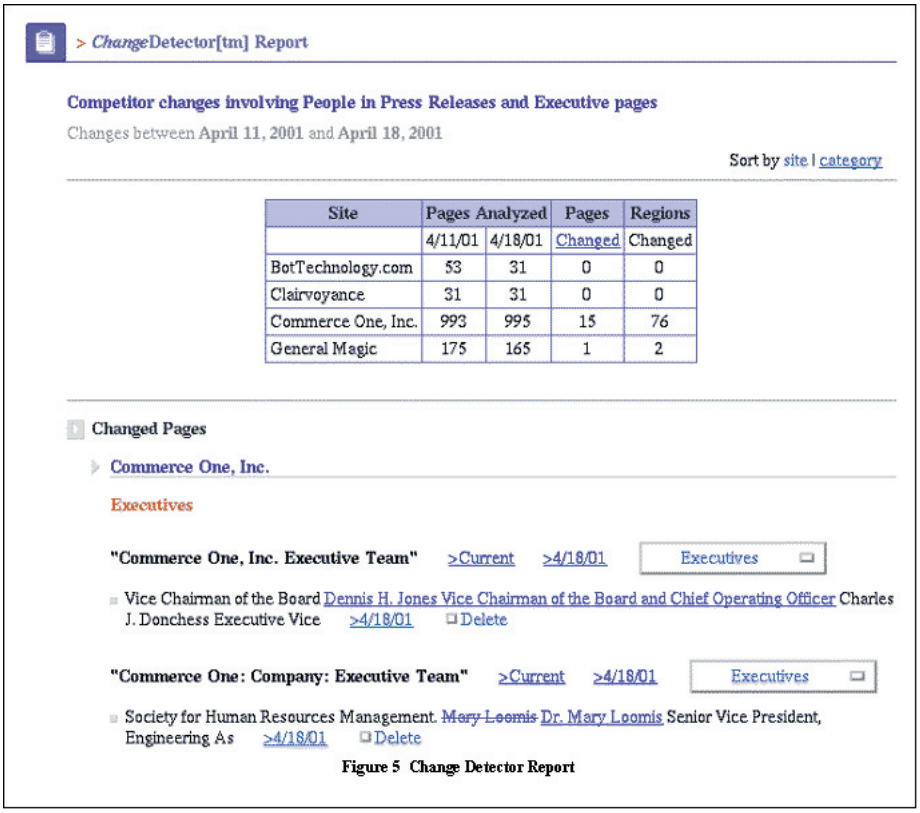
- モニタリングが，Webページ単位でしかできない：Webサイト全体をモニタリングしたいことも多いが，それが可能なシステムはほとんどない．モニタリング対象となるページが多いこと，不要な更新を取り除く必要があることなどが，Webサイト単位のモニタリングを難しくしている．
- ユーザが自分の必要な更新を十分に指定できない：ある特定の更新があったときだけ通知して欲しいことがよくある．それに対して，更新が含むキーワードを設定することで対応しているシステムがほとんどである．しかし，キーワードだけの条件では，数値の変動に対する条件，予測できないキーワードを含む更新の条件など，さまざまな記述できない更新の条件が残る．
- クライアント間の連携がない：すべてのクライアントが1つのサーバで走っており，またクライアントは独立に動作している．本来なら，すでにモニタリングされているWebページなどを共有することが望ましい．

これらの問題に対し，研究システムレベルのものも含むが，解決法を模索している先進的なWeb更新モニタリングシステムを次章で紹介していく．

先進的なWeb更新モニタリング

サイト全体の更新モニタリング

ChangeDetector[1]は，Webページ単位ではなくWebサイト単位で更新モニタリングを行えるシステムである．Webサイト単位で更新をモニタすることにより，組織内のグループの再構成や生産ラインの終了などの通常では見つけにくい変化を効果的に検出できる．ChangeDetectorは，さまざまな機械学習の技術を基盤にしている．知的クローリングにより，膨大なサイト全体から効率的に関連するWebページを収集し，集まったページから分類学習によりトピックに依存した更新抽出を行い，エンティティ抽出により，意味的にフィルタリングされた更新抽出が可能になっている．プロトタイプシステムで，毎週2,000以上のサイトをモニタリング



> *Change*Detector[tm] Report

Competitor changes involving People in Press Releases and Executive pages

Changes between April 11, 2001 and April 18, 2001

Sort by site | category

Site	Pages Analyzed		Pages	Regions
	4/11/01	4/18/01	Changed	Changed
BotTechnology.com	53	31	0	0
Clairvoyance	31	31	0	0
Commerce One, Inc.	993	995	15	76
General Magic	175	165	1	2

Changed Pages

Commerce One, Inc.

Executives

"Commerce One, Inc. Executive Team" >Current >4/18/01 Executives

Vice Chairman of the Board Dennis H. Jones Vice Chairman of the Board and Chief Operating Officer Charles J. Donchess Executive Vice >4/18/01 Delete

"Commerce One: Company: Executive Team" >Current >4/18/01 Executives

Society for Human Resources Management ~~Mary Loomis~~ Dr. Mary Loomis Senior Vice President, Engineering As >4/18/01 Delete

Figure 5 Change Detector Report

図-3　ChangeDetectorの更新通知 [1)]

可能である．

ChangeDetectorは，まずWebサイトのWebページを収集するが，1,000ページを超えるWebサイトもあるので，サイト内すべてのWebページを収集することは難しいし，ユーザはサイト内のすべての更新に興味があるわけではない．そこで，収集されたページを機械学習の文書分類を用いて，あらかじめ設定されたクラスに分類してラベルづけを行い，ディレクトリ構造とそのラベルを基にサイトマップを自動生成する．そのサイトマップを用いて，優先的に収集すべきWebページを記述し，クローラはそこを中心にページを収集する．文書分類に用いられるクラスは，一般的かつユーザの興味あるWebページを表現できるように，プレスリリース，コンタクト情報，就職情報，経営情報などの12のクラスが定義されている．

Webページが分類されると，XMLベースの差分検出が行われる．そして，検出された変化から意味のない変化を除去するフィルタリングを行う．残った更新個所が，文書分類で使われたWebページのクラスに関連づけられる．このクラスにより，どのような更新かの情報が得られる．また，クラスを使って，通知すべき更新の条件を記述し，フィルタリングできる．

以上で，Webサイト単位の更新モニタリングが実現され，後は抽出された更新が，電子メールやWebブラウザを用いてユーザに通知される．Webブラウザによる更新表示の例を**図-3**に示す．まず，上にある表に，いつ，どのサイトで，何ページが更新されたかの情報が提示され，その下には，Commerce One, Inc.のサイトにおける更新の詳細がリストで表示されている

対話的部分更新モニタリング

これまでのWeb更新モニタリングは，基本的にWebページで何らかの更新があれば通知するシステムだった．つまり，通知して欲しい更新の条件をシステムに知らせて，その条件を満たす更新のみ通知することは難しい．更新個所に含まれるキーワードを設定できるシステムもあるが，キーワードだけでは，たとえば，「ある企業の株価が9,000円を超えるか，5,000円を下回ったら通知」というような条件は記述しにくい．また，このような条件を一般のユーザが直接的に正確に記述することも容易ではない．

PUM[6)]は，ユーザが更新の位置と内容に関する条件づけを対話的にできるWeb更新モニタリングシステムである．全体の構成は，**図-4**のようになっている．最初に，ユーザは更新個所をマウスでハイライトしてPUMに与え，後はPUMからの更新通知をユーザが評価するだけ

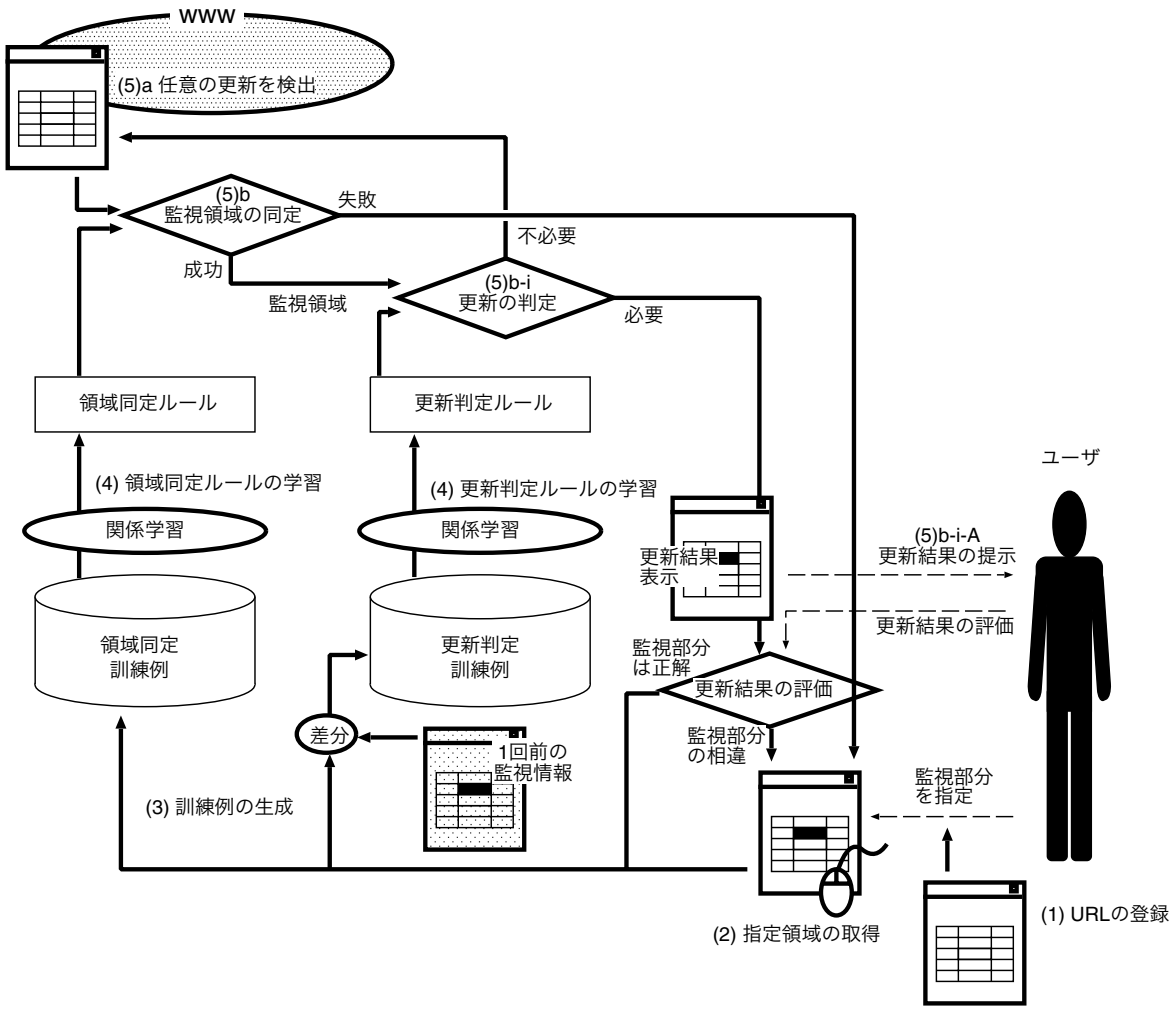


図-4 PUMのシステム構成

で，PUMが分類学習により，更新の位置を同定する領域同定ルール，更新の内容を判定する更新判定ルールの双方を学習する．そして，学習された判別ルールに基づき，更新の通知を行う．

たとえば，**図-5** に週間天気予報のWebページを表示しているPUMのインタフェースを示す．この例では，ユーザが通知して欲しい更新は，「栃木の次の日曜日の降水確率が30%以下になる」更新である．ユーザは，図-5の網掛け部分をマウスにより領域指定する．この教示だけでは，PUMはユーザが通知して欲しい部分更新を正確に特定できないため，別の曜日での更新や降水確率が上昇した更新もユーザに通知してしまう．しかし，ユーザがそれらの更新通知を評価していくことにより，徐々に正しい更新を判別するルールをPUMが学習していき，最終的には正しい部分更新のみを通知するに至る．このように，PUMは，ユーザが欲しい更新の条件を直接記述することなく，通知された更新を評価するだけで，正しい更新を学習する点が優れている．

マルチエージェントシステムとしてのWeb更新モニタリング

WebBeholder[4] は，Webページの更新通知を行う先駆的な研究である．全体の構成は，**図-6** のようにマルチエージェントシステムとなっている．更新モニタリングを行うサービス提供エージェント，個々のユーザに対応するパーソナルエージェント，エージェント間の調整を行うメディエータから構成される．ユーザは，自分の更新モニタリングの要求を持ったパーソナルエージェントをメディエータに渡すことで，更新モニタリングが可能になる．

メディエータは，要求ブローカ，ナビゲータ，ファシリティの各モジュールを持つ．要求ブローカは，パーソナルエージェントのためにクエリをサービス提供エージェントに渡したり，パーソナルエージェント間の調整を行う．ナビゲータは，パーソナルエージェントに，他のWebBeholderコミュニティの情報を提供する．また，フ

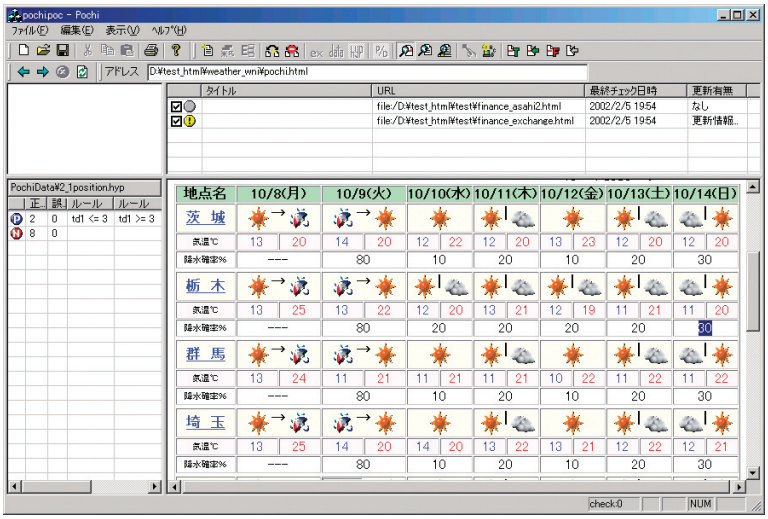


図-5 PUMのインタフェース

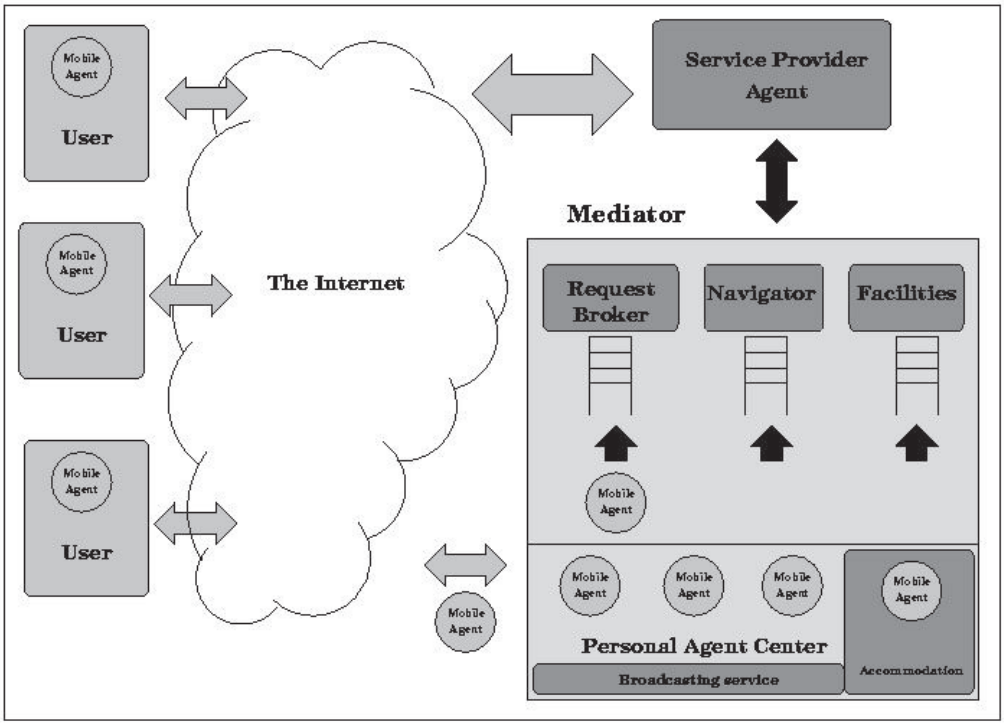


図-6 WebBeholderのシステム構成 [4)]

ァシリティは，新たに追加されたパーソナルエージェントに実行可能な諸機能を提供する．

サービス提供エージェントが行う更新モニタリングは，HTMLを構文解析した結果に対して，LCSを基に差分を抽出して，更新部分を特定する．さらに，HTMLタグを基に重み付けを行う．この重み付けは，たとえば，アンカータグ中のURLは重要であり，プレーンテキストは重要でない，というように設定される．抽出された差分に対し，この重みを用いて更新を重み付けして，重要度を評価している．**図-7**が，WebBeholderの更新通知画面である．削除された部分が線で消されており，その近くに追加された部分が表示されている．

WebBeholderの特徴として，サービス提供エージェント，メディエータ，そして複数のパーソナルエージェントが1つのコミュニティを構成して，コミュニティ間での協調が実現されていることがある．新しいパーソナルエージェントがコミュニティAに追加された場合，その更新モニタリング対象のWebページが他のコミュニティBですでにモニタリングされていれば，そのパーソナルエージェントをコミュニティBへ割り当てることができる．このように，WebBeholderでは，マルチエージェントコミュニティを使って，既存のクライアントとの連

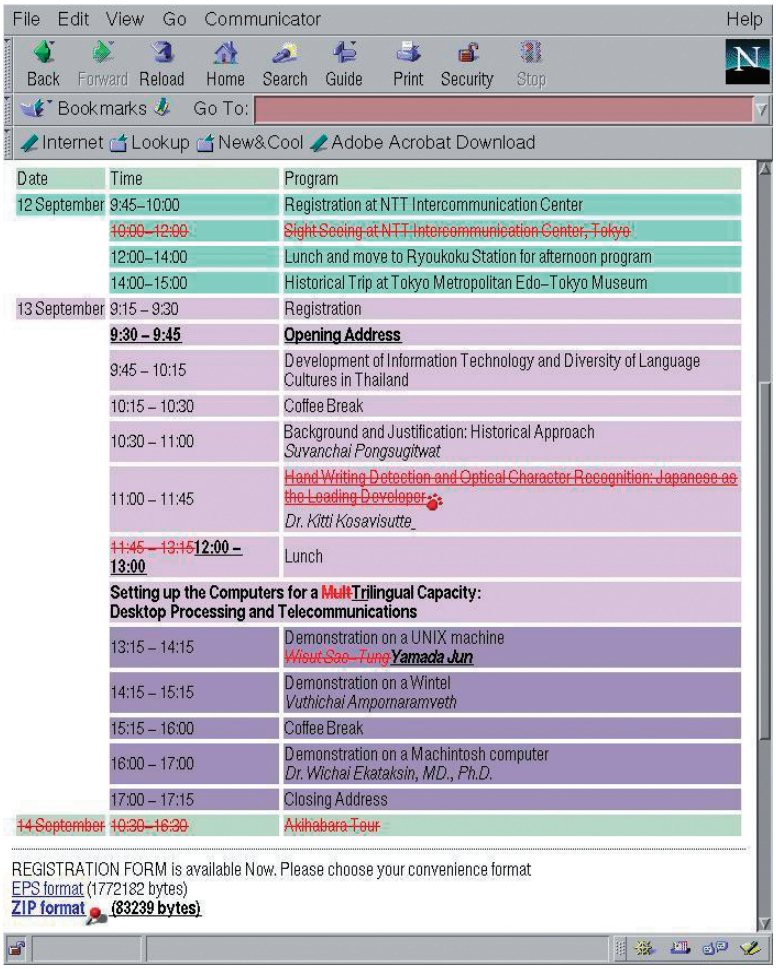


図-7　WebBeholderの更新提示 [4)]

携を実現している．

今後の方向

本稿では，現在普及しているWeb更新モニタリングシステムを紹介し，その機能，問題点について，議論した．また，それらの問題点を克服する先進的なWeb更新モニタリングのメカニズムを，具体的なシステムを例として説明した．

先進的なシステムでは，いくつかの問題点が解決されているものの，基本的には，更新があった場合，ブラウザで更新部分をリスト形式で表示するものがほとんどである．

今後は，ユーザの知りたいさまざまな更新の意図をどのようにユーザに負荷をかけることなく獲得するかというユーザモデリング，そして，検出された更新をいかに効果的に統合するかという情報統合，さらに統合された結果をいかに分かりやすくユーザに提示するかのユーザインタフェースなどの問題を，人工知能，ヒューマンインタフェースなどの関連分野の技術を使いながら解決していくことが課題であろう．

参考文献

1）Boyapati, V., Chevrier, K., Finkel, A., Glance, N., Pierce, T., Stokton, R. and Whitmer, C.: ChangeDetector: A Site-Level Monitoring Tool for the www, In WWW 2002, pp.570-579 (2002).
2）ChangeDetect, http://www.changedetect.com/
3）Chen, Y., Douglis, F., Huang, H. and Vo, K.-P.: TopBlend: An Efficient Implementation of HtmlDiff in Java, In WebNet 2000 (2000).
4）Saeyor, S. and Ishizuka, M.: WebBeholder: A Revolution in Tracking and Viewing Changes on the Web by Agent Community, In WebNet 1998 (1998).
5）WebSpector: http://www.illumix.com/
6）山田誠二，中井有紀：対話的分類学習によるWebページの部分更新モニタリング，人工知能学会論文誌，Vol.17, No.5 (2002).

（平成15年4月1日受付）