データベースシステム 128－51
（２００２．７．１９）

ウェブデータのセマンティックキャッシュ手法

成 凱　上林 彌彦

内容の近い或は関連の深い一部のデータがよくアクセスされるというような「セマンティック局所性」がウェブ利用の重要な特徴であるが，ウェブデータの利用局所性を適切に扱うセマンティックモデルがなかったため，これまでのキャッシュにはセマンティック情報がほとんど用いられていない．また，キャッシュに蓄えた情報は利用者に報せておらず，透過的にしか利用されなかかったため，キャッシュデータの利用率は低いくおよそ６割のデータが一度も再利用されずに捨てられてしまった．これらの問題を解決するために，本稿はウェブ利用を中心にキャッシュデータのセマンティック特性を取り扱う階層的モデルを提案しセマンティックキャッシュ手法の設計・検証を行った．また，プロキシキャッシュに基づいてキャッシュデータの明示的利用方式SVL（Shared Visited Links）についても述べた．

Semantic Caching Schemes for Web Data

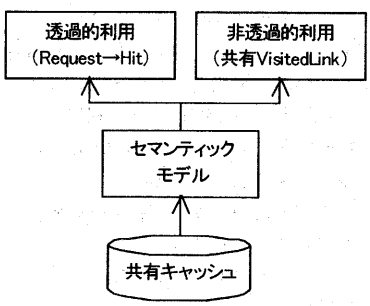
Kai CHENG　Yahiko KAMBAYASHI

Semantic locality is a salient feature of web access, which dedicates that a small part of similar or closely related web data tend to be accessed more frequently than others. However, as today's web is dominated by hypertext documents, the lack of a suitable semantic description of the cached contents makes it difficult to capture and exploit semantic locality in caching. Another problem is that cached data are not made explicitly retrievable and only a very small fraction of them may be often used. In this paper, we propose a hierarchical semantic model and define semantic region from the viewpoint of content consumers. Semantic caching schemed based on this model was described. Finally, we also demonstrate a new mechanism, called SVL (Shared Visited Links) based on the semantic caching for supporting information foraging of users subscribed to a shared caching proxy serve

1. まえがき

近年WWWの著しい発展に従い様々な情報を瞬時に発信でき必要な情報を容易に入手できるようになっている．一方，ウェブ利用の急増に伴いネットワーク通信量が年々倍増しつつありWWWがインターネットパフォーマンスのボトルネックとなっている．このパフォーマンスを向上するためにキャッシュがよく用いられているが，ウェブデータのハイパーテキスト構造及びウェブ利用のナビゲーション特性がうまく取り扱えなくてアルゴリズム改善に限界がある．その一、従来のキャッシュ手法は利用履歴からしかデータの人気度を推測できなかったため利用履歴の長いデータしか正確にキャッシュできず，新しいデータはいくら重要であっても必ずしも優先度が高くなるのではない．その二，キャッシュされたデータは利用者に知らせておらず透過的にしか利用できなかったためキャッシュデータの利用率が低く60%以上のデータが再利用されずに捨てられてしまった．

図1 ウェブデータのセマンティックキャッシュ



これらの問題を解決するため、これまでのようにウェブデータを単なる物理的なデータとして取り扱うのではなく，セマンティックな情報をキャッシュ管理に活用するべきである．更に，キャッシュの利用履歴により人気情報を判断し価値のある情報を利用者に報せることによって利用者に役立つ情報を提供しキャッシュデータの利用率を高めることができる．

本研究はキャッシュデータを重要な情報源としてモ

京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University

デル化しセマンティックな情報をキャッシュ管理に活用しウェブデータのセマンティックキャッシュ方式を提案した．さらに，キャッシュデータの透過的利用に加え非透過的な利用方式も試した。具体的に (1) 透過的利用の効率向上．透過的キャッシュ利用とはキャッシュの存在を気にせずウェブデータをアクセスする方式でありリクエストされたデータがキャッシュに入っていることを[ヒット]という。普段のウェブナビゲーションは透過的 Request-Hit 利用方式に属しており、このパフォーマンスを改善するため、セマンティック局所性を利用するセマンティックキャッシュを提案した。(2) 非透過的利用の導入．システムが把握している利用履歴に基づく人気コンテンツを提示することによって，利用者が意図的にキャッシュデータを利用する方式で，キャッシュデータ利用率を高めるだけでなく，利用者のウェブ検索に支援することもできる。非透過的利用を可能にするため，我々は利用履歴を意識する共有 VisitedLink 機能を実装した．本報告はこれらの内容について述べる．

2. ウェブデータのセマンティックモデル

ウェブデータの多くはハイパーテキストでありおよそ 70%あまりのウェブデータは HTML 文書の形でネットワーク上に存在している．ハイパーテキストデータの利用方式はリンクを辿りながら必要とする情報を見つけるものであり従来のデータベースのように検索言語を使って必要な情報を探し出す方式と大いに違っている[3, 5]．ナビゲーションを中心とするセマンティック特性を取り扱うため，ウェブデータを適切にモデル化することが重要であるが，これまでの研究はハイパーテキスト文書の作成やハイパーメディア質問言語に向けており，キャッシュのような特定利用者の利用状況に応じて動的に構成されたデータには相応くない[3, 4, 5]．例えば，[3]と[5]はウェブ利用のナビゲーション性質を配慮しながら Web Machine や Web Automaton といった計算モデルを提案しウェブデータのモデル化を図ったが，ウェブ利用の局所性を取り扱うには適していない。本章はウェブキャッシュに適したセマンティックモデルを提案する。

2.1 利用局所性

データ全体には一部しかよくアクセスされていないという現象は利用局所性 (Locality of Reference) と呼ばれウェブだけでなくメモリ，ディスク，データベースなどあらゆるシステムにも利用局所性がよく知られている．利用局所性は次のように三種類に分けられる．(1) 時間的局所性(Temporal Locality)：最近利用された一部のデータがよく利用される；(2) 空間的局所性(Spatial Locality)：物理的距離の近い一部のデータがよく利用される．例えば、あるウェブサイトの関連データはよくアクセスされる；(3) セマンティック局所性 (Semantic Locality)：論理的距離の近い一部のデータがよく利用される．例えば FIFA サッカーワールドカップに関する文書は内容が近いので興味のある利用者によくアクセスされる可能性が高い．

利用局所性はアクセスの集中度を示すものでキャッシュ効率に決定的な影響を持つ．特にセマンティック局所性は最も高度的であり知的キャッシュ管理に重要である．しかし，ウェブデータがほとんどハイパーテキスト文書であり適切なセマンティックモデルがなかったため，これまでのウェブキャッシュ手法にはセマンティック局所性がうまく取り扱われていなかった。

2.2 HTTP オブジェクト(HTTP Objects)

ウェブは HTTP プロトコールに従い完全なファイルを伝送しないといけないので，キャッシュの受けたデータも完全なファイルが最小単位となる．それゆえ HTTP で伝送されるあらゆるタイプのファイルは HTTP オブジェクト（またはオブジェクト）と呼ばれ HTTP オブジェクトは次の属性をもつ．

URI	Type	Size	Frequency	Recency	TTL

HTTP オブジェクトの ID は URI (Uniform Resource Identifier) である．Size はオブジェクトサイズで，Type は HTTP メディアタイプでテキストタイプ(html, plain, xml, css)，イメージタイプ(jpeg, gif, png)，オーディオタイプ，ビデオタイプ（mpeg）などがある．オブジェクトの利用状況は利用頻度（Frequency），最近利用性（Recency），有効期限（TTL：Time-To-Live）である．

HTTP オブジェクトはキャッシュ置換の基本単位で効率的に扱わないといけない．それゆえ HTTP オブジェクトはよくハッシュ構造で管理される．ハッシュは URI（ハッシュキー）を引数としてのハッシュ関数の計算結果よりデータの物理場所（ディレクトリー）を決める．例えば MD5 は最もよく採用されるハッシュ関数で記号ストリングを引数とし 128 ビットの整数を出力する．与えられた URI からハッシュでオブジェクトの物理的場所は次のように求める．

URI ⇒ MD5_hash(URI)=ABCD ⇒ /A/BC/D

即ち，オブジェクトの ID は URI であればファイルはローカルに/A/BC/D に蓄えることになっている．

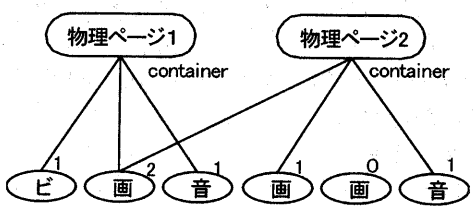
2.3 物理ページ(Physical Page s)

HTTPオブジェクトはHTTPが正しく取り扱える最小単位であるが，ウェブはハイパーテキストであるため利用者にとって個々のHTTPオブジェクトだけでは完全な情報をえるはずはない．利用者にとって完全な情報単位として複数のHTTPオブジェクトを物理ページとしてまとめる必要がある．

定義1：物理ページ

HTMLファイル及びそこに埋め込まれたメディアファイルを一つの物理ページである．HTMLファイルはコンテナー（Container），埋め込まれたファイルはコンポーネント（Components）と呼ぶ．

図2：埋め込みHTTPオブジェクト共有



物理ページは次のような性質を持つ

（1） コンポーネントは複数の物理ページに共有できる（**図2**）．コンポーネントを埋め込んでいる物理ページの数は「引用数」という．

（2） 物理ページのサイズはコンテナーとコンポーネントのサイズの和とする．

（3） 物理ページの利用状況（利用頻度、最近利用性など）はコンテナーの利用状況に従う．

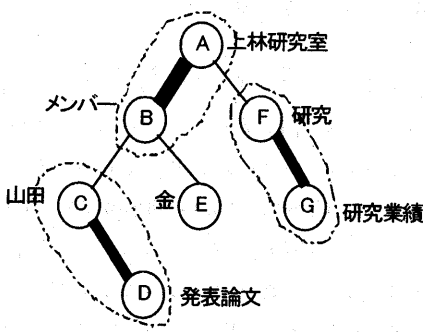
物理ページはHTML文書を中心として定義されたが，テキスト，ワード，PDFなどテキストを取り出せるデータもコンポーネントなしの物理ページとして認める．また，物理ページはコンテナーのアクセスに従ってページ全体がアクセスされるので，利用履歴はコンテナーであるHTTPオブジェクトに従うことになる．このため，多くの物理ページに共有されたコンポーネントは利用頻度が高まるはずである．

2.4 論理ページ(Logical Pages)

物理ページは独立のものではなく関連ページがハイパーリンクでつながっており，利用者はリンクを辿ってページの間に移動しながら必要な情報を見つける．大きな物理ページは読みにくく伝送時間も長くてパフォーマンスに問題があるので，ウェブ著者やサイト管理者はできる限り大きなドキュメントを複数の小さな物理ページに分割し元の論理関係をリンクで維持する傾向が強まる．

しかしドキュメントの分割やリンクの埋め方はあくまでも著者（コンテンツ提供者）側の意志を反映するもので，利用者（コンテンツ消費者）の認識とは必ずしも一致するのではない．著者側はできる限り各種アクセス経路を提供するようにリンクを埋めるが，しかし利用者は必ずしもすべてのアクセス経路を同じように辿るわけではない．利用者がよく訪ねたのは数少ない一部のリンクだけであるし，たとえ同じページを訪

図3 利用状況による論理ページ



れるとしても利用者によってアクセス経路が違うことがある．例えば，図3で示しているのはある研究室のウェブサイト構造でありホームページAを初め関連情報（メンバー構成、研究内容など）をアクセスできるようにしているが，人を探す利用者はよくホーム（A）からメンバー（B）へナビゲーションするが，研究に関心を持つ利用者は研究関係のページ（F）から研究業績（G）の経路をよく訪れる．

キャッシュは利用者のためにウェブデータを蓄えるので，リンク構造全体ではなく利用者の興味のありそうな部分をモデル化することが重要である．我々は利用者のよく訪れる経路を「論理ページ」と呼ぶ．論理ページを定義するには以下のような概念が必要である．

定義2：アンカー

p_i におけるアンカー a は次のような三つの項 $< p_i, n_1, n_2 >$ で定義する．ここで $p_i \in P$ は物理ページであり $n_1 \le n_2 \in N$ は自然数でアンカー範囲（Anchor Scope）と呼ぶ．n_1, n_2 のいずれかが未

定義の場合アンカー範囲はページ p_i 全体とする．アンカー範囲のテキストは「アンカーテキスト」という．

ひとごといえばアンカーは物理ページの n_1, n_2 の間の部分でありリンクの埋める始点と終点を指定する役割がある．

定義 3：リンク

リンク l は二つの項 $< a_i, a_j >$ で定義する．ここで a_i, a_j はいずれもアンカーである．a_i のアンカーテキストはリンク l のラベルであり *label(l)* で現す．リンク l を辿って p_i から p_j をアクセスする確率は $\Pr(p_j \mid p_i)$ とする．

利用履歴から次のように $\Pr(p_j \mid p_i)$ を求める．

(1) まず利用履歴からユーザセッションを決める．ユーザセッションは特定の利用者の継続的アクセス活動である．「継続的」とは時間間隔が 20 分以内とする．

(2) 次に $\Pr(p_j \mid p_i) = \Pr(p_i p_j) / \Pr(p_i)$ を求める．ここで $\Pr(p_i p_j)$ は p_i に続いて p_j をアクセスする確率でありと $\Pr(p_i)$ は p_i をアクセスする確率である．再び図 x を例として説明する．ユーザセッション数は 100 と仮定しうち，A を含むセッションが 20 で，A の後 B がアクセスされたセッションが 9 とすれば，$\Pr(A)$ =20/100, $\Pr(AB)$ =9/100, 結果として $\Pr(B \mid A)$ =0.09/0.2=0.45. つまり A がアクセスされると A からリンクを辿って B をアクセスするコンフィデンスは 0．45 である．

定義 4：k 次元巡回経路

(1) $L^{(0)} = P$ は 0 次元巡回経路である．

(2) $l =< a_i, a_j >$ が物理ページ $p_i, p_j \in L^{(0)}$, $(p_i \neq p_j)$ の間リンクとする．もし $\alpha = \Pr(p_j \mid p_i) > \lambda$ であれば，$l^{(1)} =< p_i, l, p_j >$ はコンフィデンス α の 1 次元巡回経路と呼ぶ．1 次元巡回経路の全体を $L^{(1)}$ で表示する．

(3) $l^{(i)} =< p_1, l_1, \cdots, p_{i+1} > \in L^{(i)}$ がコンフィデンス $\alpha^{(i)}$ の i 次元巡回経路で $l^{(1)} =< p_1', l_1', p_2' > \in L^{(1)}$ はコンフィデンス $\alpha^{(1)}$ の次元巡回経路とする．もし $\alpha = \alpha^{(i)} \cdot \alpha^{(1)} > \lambda$ かつ $p_1' = p_{i+1}, p_2' \notin l^{(i)}$ もしくは $p_2' = p_1, p_1' \notin l^{(i)}$ であれば $< p_1, l_1, \cdots, p_{i+1}, l_1', p_2' >$ もしくは $< p_1', l_1', p_1, l_1, \cdots, p_{i+1} >$ はコンフィデンスαの i+1 次元巡回経路である．

定義 5：部分経路

下記の条件を満す i(0≤i≤n-m) が存在すれば，$p_j = p_{i+j}'$ (j=1,...,m+1), $l_k = l_{i+k}'$ (k=1,...,m), m 次元巡回経路 $< p_1, l_1, \cdots, l_m, p_{m+1} > \in L^{(m)}$ は n (m≤n) 次元巡回経路 $< p_1', l_1', \cdots, l_n', p_{n+1}' > \in L^{(n)}$ の部分経路と呼ぶ．

定義 6：論理ページ

論理ページは部分経路を除いた任意次元の巡回経路である．つまり，$l^{(i)} \in L^{(i)}$ (i>0) しかし $l^{(i)}$ は $l^{(j)} \in L^{(j)}$, (i<j) の部分経路である j が存在しないと $l^{(i)}$ は i 次元論理ページという．すべての論理ページを L で表示する．

以上の定義から次のような事実がわかる．

(1) 論理ページは巡回経路であり木構造ではない．経路の終点は「目的ページ」と呼ぶ．

(2) 任意の物理ページは少なくとも一つの論理ページに属する；

(3) 0 次元以上の論理ページは他の論理ページに含まれない．

(4) 論理ページの次元数に制限はないが，コンフィデンスの閾値λが大きくすると，高次元の論理ページがほぼ不可能となる．閾値λは 0.5 以上が必要である．

ウェブナビゲーションはリンクを辿ってページとページの間に移動する活動であり，リンクテキストとリンク先のページの内容が最も重要である．論理ページのセマンティック特性は次のように取り扱う．

定義 7：論理ページのセマンティック構成

論理ページ $l^{(m)} =< p_1, l_1, \cdots, l_m, p_{m+1} > \in L^{(m)}$ のセマンティックな内容は次のように構成される．

(1) タイトルは論理ページを構成するリンクのラベル

を連結した文に目的ページのタイトルを加えるものである．すなわち

$$title(l^{(m)}) = title(p_{m+1}) + \sum_{i=1}^{m} label(l_i)$$

(2) ボディー：論理ページの目的ページの

$$body(l^{(m)}) = body(p_{m+1})$$

(3) $l^{(m)}$ に対して単語 t_i の重みは

$$weight(t_i, l^{(m)}) = weight(t_i, body(l^{(m)})) \times 10$$

$$y_i = \begin{cases} 1, & t_i \in title(l^{(m)}) \\ 0, & t_i \notin title(l^{(m)}) \end{cases}$$

このように論理ページの VSM(Vector Space Model)による特徴ベクタルが表示し論理ページのセマンティック距離及び類似性が計算できる．式 (3) タイトルは文書内容を強く示すので，ここは因子 10 で強調する．この式で論理ページが動的に変わってもタイトル部分の重みだけ計算し直す必要があるので，大きな負担がかからない．

最後に，この後で説明しやすいように次の記号を用いる．$\vec{v}(l)$ は l の特徴ベクタルで $d(l_i, l_j)$ は $\vec{v}(l_i), \vec{v}(l_j)$ のセマンティック距離を表示する．さらに L_i は論理ページの部分集合の中心 $\bar{l}_i$ の特徴クタルは各要素の平均特徴ベクタルとする．

$$\vec{v}(\bar{l}_i) = \frac{1}{n} \sum_{l \in L_i} \vec{v}(l)$$

2.5 セマンティック領域 (Semantic Regions)

ウェブ利用の主な目的は必要な情報を探すことであるため，キャッシュされたデータをセマンティック類似度によりセマンティック局所性を取り扱うことができる．セマンティックに緊密に関連している論理ページはセマンティック領域を構成する．

定義 8：セマンティック領域

$L/_{\approx} = \{L_1, L_2, \cdots, L_k\}$ は論理ページのセマンティック類似性に基づく L の次の条件を満す分割でであれば $L_i (i = 1 \cdots, k)$ がいずれもセマンティック領域という．

(1) $L = \bigcup_{i=1}^{k} L_i$，かつ i≠j であれば $L_i \cap L_j = \phi$

(2) i≠j であれば $d(l_i, \bar{l}_i) < d(l_i, \bar{l}_j)$．

類似するウェブデータをセマンティック領域にまとめることによってセマンティック局所性を取り扱うことができるようになりキャッシュ管理に活用できる．さらに，セマンティック領域はキャッシュのためだけでなく，もう少しあとで説明するようにキャッシュの非透過的利用にも役立てる．

セマンティック領域セマンティック距離の定義，k の決め方，クラスタリングの方式によるものであるが，関連研究は多かったため詳しい分析は本研究の範囲外になる．

ここまで，ウェブデータのセマンティック利用局所性を取り扱うためのデータモデルを提案した．これより提案モデルをどのようにキャッシュ管理，そしてキャッシュの利用者を支援するのかについて検討する．

3. セマンティックを意識するキャッシュ

セマンティック局所性を生かすため，キャッシュ管理にセマンティック領域，論理ページ、物理ページ，HTTP オブジェクトを取り組んで次のようなセマンティックを意識するキャッシュ手法を提案できる．

3.1 セマンティックキャッシュの管理方式

キャッシュデータは**図 4** が示す 3 種類の優先度キューで管理を行っている．(1) セマンティック領域の優先度キューはセマンティック領域の重要度，利用状況から決められた優先順位を管理するキューである。(2) 論理ページの優先度キューは各セマンティック領域における論理パージの優先順位を管理するキューである。(3) 物理ページの優先度キューは論理ページを構成する物理ページの優先順位を管理するキューである。

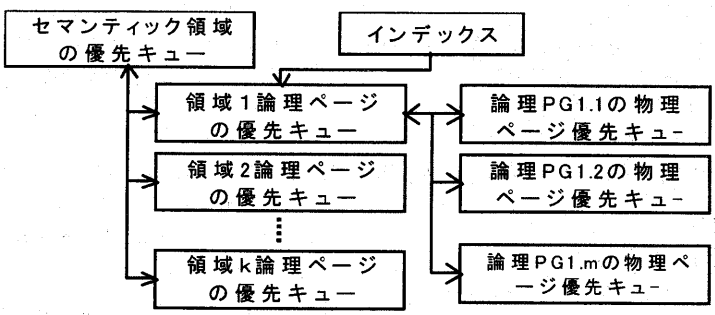
また，論理ページの類似性を求めるため，キャッシュメンテナンスの際にインデックスをする．

新しいデータが入る時点で，まずデータの該当物理ページがどの論理ページに属するかを決める．もし新規の論理ページであれば，ひとまず未定のセマンティック領域に入れておく．もし該当する論理ページがすでに存在すれば，該当する論理ページ及びセマンティック領域の利用状況を更新する．

3.2 キャッシュ置換

キャッシュ管理には利用履歴とその関連情報からキャッシュデータに優先順位をつけることが最も複雑だが重要である．優先順位が決められたら，キャッシュスペースを上げなければならない場合は優先順位の最も低い順にデータをキャッシュから追い出し，新規データを適当にキャッシュに入れる．

図4：セマンティックキャッシュ管理方式



キャッシュ置換はHTTPオブジェクトの単位で次のよう手続きとおり行う.

（1） セマンティック領域優先度キューから優先度の最も低いセマンティック領域 L^* を選び出する.

（2） L^* に相当する論理ページ優先度キューから優先度の最も低論理ページ l^* を選び出する.

（3） l^* に相当する物理ページ優先度キューから優先度の最も低物理ページ p^* を選む.

（4） p^* から「引用数」0のHTTPオブジェクトを追い出す.残りのHTTPオブジェクトの「引用数」を1減らす.

3.3 キャッシュメンテナンス

キャッシュ管理のオーバーヘッドを軽減するため,キャッシュ管理のなかに時間やメモリの消耗量の大きな操作をシステムの余裕のある時間帯に行うことが重要である.キャッシュメンテナンスは（1）インデックス更新,（2）セマンティック領域更新がある.

まず,未定のセマンティック領域に一時預かった論理ページを解析しインデックスに更新する.次にこれらの論理ページをクラスタリングし,該当するセマンティック領域を求める.

4. キャッシュ内容の提示方式：SVL機能

キャッシュされたデータを明示的にアクセスできるように我々は共有のキャッシュプロキシサーバに基づいてキャッシュ内容の提示方式SVLを開発してきた.

現在のウェブクライアントソフトには Visitedlinkという機能が含まれている。利用者の最近アクセスしたページはクライアントのキャッシュに一時格納され,後ほどこのページへリンクしているページを閲覧する時、Visitedlinkとして特別な色で表示される。例えば,Microsoft社のIE, Netscape社のNavigatorではリンクが「表示済み」か「未表示」かによって違う色で表示できる. Visitedlink 機能により利用者が既に辿ったリンクであることや,リンク先のページは自分のコンピュータに入っていることが分かる。

しかしこのような Visitedlink 機能はあくまでも個人レベルであり,ほかの利用者が同じページをアクセスしても他人がこのページに興味があることを意識できずお互いに経験が共有しないまま終わってしまった。この問題を解決するため、VisitedLink を共有することが重要である。以下我々が開発しているSVL(Shared Visited Links) という利用履歴の共有機能について述べる.

$\vec{V}_i$ ページi	本日 利用頻度 $v_{i,0}$	今週平均 利用頻度 $v_{i,1}$	今月平均 利用頻度 $v_{i,2}$
ページ1	4	5.5/日	4.3/日
ページ2	0	8.0/日	3.9/日
…	…	…	…

表1：利用履歴管理の仕組み

4.1 利用履歴による人気ページ

SVLを実現するためキャッシュが把握している利用履歴に基づいてページの人気度を判断し共有する価値があるかどうかをきめる必要がある。人気度は物理ページ i の当日, 今週と今月の一日平均利用頻度 $\vec{V}_i = (v_{i,0}, v_{i,1}, v_{i,2})$ によって決められる. 表1はこのような利用履歴データである. この利用履歴に基づいてページの人気度を判断するには利用者の好み, データな内容に応じて重み $\vec{W} = (w_0, w_1, w_2)$ を付ける. それでページiの人気度は以下のように計算できる

$$v_i = \vec{W} \cdot \vec{V}_i = \sum_{j=0}^{2} (w_i \cdot v_{i,j})$$

$\vec{W}$	w_0	w_1	w_2
長期重視型	0.10	0.10	0.80
短期重視型	0.70	0.30	0.00
安定利用型	0.33	0.33	0.33

表2：内容，利用者の好みに応じた重み

4.2 SVL (Shared Visited Links)機能

(1) 利用者u：ページiをリクエストする（デフォールトにSVLトップページ）

(2) プロキシ：このリクエストを受けキャッシュかオリジナルなサーバからページiを検索する

(3) ページiを解析し各リンクkは次のように処理する

① kが履歴表に記録がなければ次へ

② kが履歴表に記録があれば，式[1]でkの人気度v_k，そしてkの外観属性を求める

③ kのラベル部分の属性を書き換える

(4) 書き換えたページiをuに返す

5. まとめ

本研究ではウェブデータの利用率およびキャッシュ効率の向上を実現するため，利用状況を取り組むセマンティックモデルを提案した．このモデルに基づくセマンティックキャッシュの設計を行った．さらにっキャッシュ内容を利用者に報せるための Shared Visited Links (SVL)機能についても述べた．

参考文献

[1] K. Cheng and Y. Kambayashi. "Enhanced Proxy Caching with Content Management", Knowledge and Information Systems, An International Journal. vol.4, no.2 pp.202-218 April 2002

[2] K. Cheng, Y. Kambayashi, "A Semantic Model for Hypertext Data Caching". Proc. 21st International Conference on Conceptual Modeling (ER2002), Tampere Finland, October 7-11, 2002 (to appear)

[3] S. Abiteboul and V. Vianu. Queries and Computation on the Web. In Proc. 6th International Conference on Database Theory (ICDT'97), pp.262-275, January 8-10, Delphi, Greece, 1997.

[4] F. Afrati and C. Koutras. A Hypertext Model Supporting Query Mechanisms. In Proc. of European Conference on Hypertext (ECHT'90), pp.52-66, 1990.

[5] A. Mendelzon and T. Milo. Formal Models of Web Queries. In Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), pp. 134-143, Tucson, Arizona, 1997.