

メタ知識に基づくウェブ探索を容易にする FootprintTrailer と ウェブブラウザとの連携機構

A Mechanism of Linking a Web Browser with the FootprintTrailer Facilitating a Meta-knowledge-based Search

足立 吉隆[†] 佐藤 慶三[‡] 中島 誠[‡]

Yoshitaka Adachi Keizo Sato Makoto Nakashima

1. はじめに

ウェブを介した情報収集やウェブアプリケーションの利用に際し、ユーザの多くは、自らや他のユーザが過去にウェブ探索を経て見つけたウェブページを繰り返し再訪問する[6]。再訪問には、過去のブックマークなどの履歴の利用や、ログイン情報の入力などを繰り返し行う必要がある。また、変化の激しいウェブ上では、履歴が利用できなくなることも多く、過去にウェブページを見つけた方法を再度試し、新たにウェブページを探索することになる。このような、ウェブ探索や再訪問を経たウェブ探索をうまく行うには、日常的に利用するウェブブラウザに、繰り返し操作の軽減や新たなウェブページの探索を支援する仕組みが要る。

これまで我々は、ユーザの望むウェブページの見つけ方をユーザの持つメタ知識と捉え、それに基づいたウェブ探索およびそれを通して見つけたウェブページへの再訪問を容易にする機構 FootprintTrailer を提案してきた[8]。これは、ユーザが過去に見つけたページの URL、クリックしたリンク、あるいはログイン情報などの履歴をウェブ探索におけるユーザの知識 (Footprint) として記録し、これを用いてウェブ探索過程をウェブブラウザ上に再現する。再訪問に際し、ユーザによる繰り返し操作の必要がなく、ウェブページが移転している場合も、再現過程の途中からユーザ自身によるメタ知識に基づくウェブ探索を行える仕組みを有する。

メタ知識に基づくウェブ探索では、ユーザが自身の経験から得たやり方で、望むウェブページを見つけてゆく。一方で、知識に基づくウェブページ再訪問では、過去にウェブ探索で見つけたウェブページを、見つけた際の知識を基に再訪問するため、自動的に見える。ここで、メタ知識に基づくウェブ探索は、知識に基づいた再訪問のつづきとして、あるいは知識を更新するためにも行われる。例えば、過去に見つけたニュース記事のウェブページを再訪問する際、その記事が削除された後、そのウェブページ周辺のウェブページやさらにそこからのリンクを介して新たに記事を探して行ける。

このような、ウェブ探索を容易に行うためには、ウェブページ再訪問をウェブブラウザ上で任意のタイミングで利用でき、また、ウェブページの読み込み状況に合わせ適切な再訪問操作を行える仕組みを実現すればよい。そのためには、ウェブページ再訪問の仕組みとウェブブラウザとを

連携させ、ウェブ探索とウェブページの再訪問を容易に切り替えながらウェブページを参照できる仕組みが必要になる。

本稿では、FootprintTrailer とウェブブラウザと連携させてメタ知識に基づくウェブ探索を容易に行える機構を提案する。プラグインの技術を利用して、FootprintTrailer とウェブブラウザとの連携機構を実現することで、ウェブブラウザを利用する多くのユーザがウェブページの再訪問に際し、メタ知識に基づくウェブ探索を簡便に行えるようにする。大学生を被験者とした実験では、繰り返し再訪問する教務情報システムとニュース記事への再訪問を想定し、連携機構によって、ウェブ探索と再訪問いずれも容易に行え、ユーザのメタ知識に基づくウェブ探索の支援を行えることが確認できた。

2. 関連研究

知識に基づくウェブ探索を支援する方法として、BrowsLine[3]や Semantic History Map[9]がある。それぞれは、ユーザが再訪問したいウェブページを発見しやすくするために、ユーザの探索履歴を時間軸に沿ってウェブページタイトルとアイコンを配置したり、検索質問と検索結果のブラウジング過程を追加されたキーワードとともに木構造で視覚化したりする。また、視覚化の際に過去に見つけたウェブページのサムネイルを利用することで、ユーザが再訪問したいウェブページを見つけやすくする Visual Snippets[10]、Contextual Web History[11]や、候補となるウェブページをランキングして提示するための PivotBar[4]がある。しかしながら、これらの手法はユーザのウェブ探索履歴の利用による支援方法で、ログイン操作が必要な場合や、ウェブページが変更された場合など、操作の再現や、メタ知識に基づくウェブ探索が必要な場合への対処がない。Memento [5]では、ユーザの探索履歴を視覚化する代わりに、検索質問とその検索結果のウェブページのタイトルや URL を提示することで、検索を通じた再訪問を支援している。しかしながら、ユーザのウェブ探索は検索エンジンによるものではないため、ウェブ探索支援としては不十分である。

FootprintTrailer[8]は、ユーザのメタ知識に基づくウェブ探索過程を通して得られた操作情報やウェブページの断片情報を知識として、ウェブブラウザ上で実際に操作を再現しながら、ウェブページを再訪問する。この機構では知識に基づく再訪問後、ユーザが任意に他のウェブページの探索へ移行することはできるが、ユーザがウェブブラウザから容易に利用するためのユーザインタフェースは備えられていなかった。

本稿で提案する連携機構は、ウェブブラウザから FootprintTrailer を任意に呼び出せる仕組みと、ウェブブラウザの状態を取得して FootprintTrailer とウェブブラウザを連携させる仕組みにより、ウェブ探索とウェブページの再

[†] 大分大学大学院知能情報システム工学専攻
Dept. of Computer Science and Intelligent Systems,
Graduate School of Engineering, Oita University

[‡] 大分大学工学部知能情報システム工学科
Dept. of Computer Science and Intelligent Systems,
Faculty of Engineering, Oita University

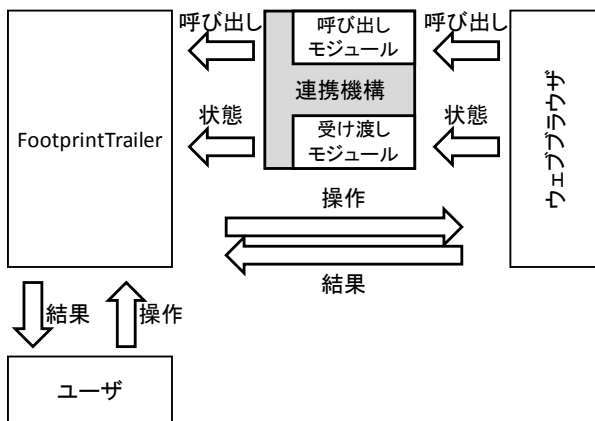


図1 FootprintTrailer とウェブブラウザとの連携機構

訪問の繰り返しを容易に行えるようにすることで、メタ知識によるウェブ探索を支援する。

3. FootprintTrailer とウェブブラウザとの連携機構

ここでは、FootprintTrailer[8]とウェブブラウザとの連携機構の概要について述べる。FootprintTrailer は、メタ知識に基づくウェブ探索の過程をユーザーの知識（Footprint）として整理・記録する動作と、その知識を辿る（Trail する）ことで知識に基づく再訪問を行う。Footprint は、ウェブ探索を開始したウェブページの URL に加え、ウェブ探索の過程で参照されたウェブページ上でリンクをクリックした際のウェブブラウザのウィンドウ上での座標とその周辺画像の色情報、ユーザーが検索フォームに入力したキーワードやログインのために入力したログイン名、パスワードで構成されており、ユーザーのウェブ探索過程に沿って整理・保存される。

ウェブ探索開始時に FootprintTrailer を起動しておけば、望むウェブページを見つけた時に、ウェブ探索を開始したウェブページから見つけたウェブページに至る探索過程を Footprint として記録しておける。再訪問の際は FootprintTrailer を呼び出し、再訪問したいウェブページの Footprint を選択することで、FootprintTrailer が以前再訪問した過程を自動的に Footprint に沿ってウェブブラウザ上に再現しウェブページを再訪問する。

メタ知識によるウェブ探索を支援するための、FootprintTrailer とウェブブラウザとの連携では、以下の2つの要件を満たす。

- (1) ウェブページ再訪問のための FootprintTrailer をウェブブラウザから呼び出す。
- (2) 再訪問を安定して行えるようにウェブブラウザからウェブページの読み込み完了情報を取得する。

項目1では、ユーザーが日常的にウェブブラウザを利用し行うウェブページの再訪問の繰り返しを容易に行えるように、FootprintTrailer の呼び出しをウェブブラウザから行えるようにする。項目2では、ウェブブラウザからウェブ上の各ウェブページを辿る際、通信状況によっては遅延が発生し、再訪問の途中で、ウェブページの読み込みを完了する前にリンクをクリックしてしまったりキーワード入力を試みたりすることがないよう、同期をとる。

図1に上記の要件を満たす、FootprintTrailer とウェブブラウザとの連携機構を示す。図中、連携機構は FootprintTrailer を呼び出すための呼び出しモジュールと、ウェブブラウザの状態を FootprintTrailer に渡すための受け渡しモジュールからなる。連携機構を介してウェブブラウザから FootprintTrailer を呼び出せるようにすることで、ユーザーはウェブ探索後、任意に FootprintTrailer を呼び出してウェブページを再訪問できる。また、ウェブブラウザの状態を FootprintTrailer に受け渡すことで、ウェブページを再訪問する際、通信状況の変動により、ウェブページ読み完了前にリンクのクリックや検索フォームへのキーワード入力などの操作が失敗することを防ぐ。

FootprintTrailer をウェブブラウザから呼び出せるようにする技術として、プラグインによる外部プログラムの呼び出し機能の利用が考えられる。プラグインを介して外部プログラム自体をプラグインの様に利用することで、ウェブブラウザからの利用が容易になる。また、ウェブブラウザのウェブページ読み込み完了の情報は、ユーザーのリンククリックなどの情報と同じ方法では取得できないため、専用のモジュールを作成し、FootprintTrailer に受け渡す必要がある。4.では、FootprintTrailer をウェブブラウザのプラグインとして利用する方法およびウェブブラウザからの状態取得について述べる。

4. 連携機構の実現

連携機構を実現するために、プラグインの利用による FootprintTrailer のウェブブラウザからの呼び出しを行う呼び出しモジュールと、ウェブブラウザのウェブページ読み込み完了の情報を仲介して FootprintTrailer へ受け渡し受け渡しモジュールの、2つのモジュールについて述べる。ここで、ウェブブラウザは Mozilla Firefox を用いる。

4.1 呼び出しモジュール

ユーザーのメタ知識に基づくウェブ探索の過程を記録することや、その知識に基づき再訪問を行う機構は、ウェブブラウザで動く Java Applet やスクリプト言語だけでは実装は難しい。それは、FootprintTrailer では、CollaboTray[1], [7]によるウェブブラウザのウィンドウイメージの取得やマウス・キーボードイベントの取得・送信、Template Matching[2]を利用したリンク位置移動への対応を必要とするためである。

FootprintTrailer とウェブブラウザとの連携機構を実装する方法として、ウェブブラウザに新しい機能を追加させるプラグインの機能を利用する。これにより、専用のウェブブラウザの開発や既存のウェブブラウザの改変なしに、FootprintTrailer を組み込むことができる。FootprintTrailer はウェブブラウザからロードされ、ユーザーは任意のタイミングでウェブ探索あるいはウェブページの再訪問する。

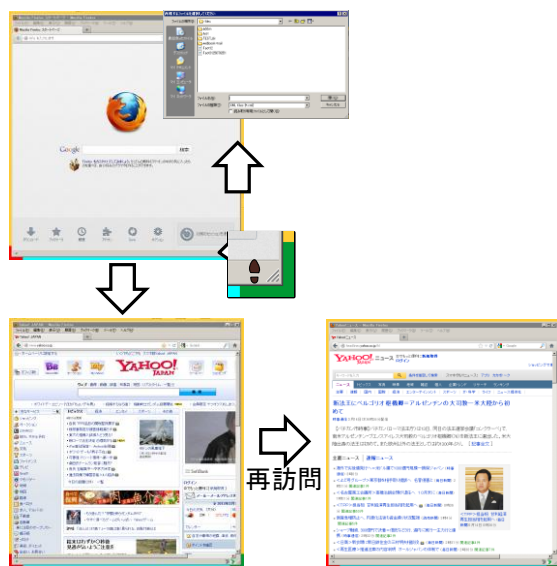
FootprintTrailer をウェブブラウザのプラグインとして利用できるようにするため、この外部プログラムを呼び出すための、スクリプトを記述し、それをプラグインとしてインストールする。FootprintTrailer とウェブブラウザを連携させるためのプラグインは JavaScript によって記述できる。FootprintTrailer の構造は変更することなく、それ自体をプラグインのように呼び出すことで、FootprintTrailer がウェブブラウザと連携した形で利用できるようになる。以下に、



図2 ウェブブラウザからのFootprintTrailer呼び出し



(a) メタ知識によるウェブ探索



(b) FootprintTrailerによるウェブページ再訪問

図3 ウェブ探索およびウェブページ再訪問過程

JavaScript による、FootprintTrailer を呼び出すスクリプトを示す。記述中、filePath が FootprintTrailer のインストールされたパスとなる。

スクリプト:

```
function createProcess(filePath, argc, argv)
{
    var file =
    Components.classes["@mozilla.org/file/local;1"].createInstance(
        Components.interfaces.nsILocalFile);
```

```
file.initWithPath(filePath);
var process =
Components.classes["@mozilla.org/process/util;1"].createInstance(
    Components.interfaces.nsIProcess);
process.init(file);
var args = [argc, argv];
process.run(false, args, args.length);
}
```

図2に、FootprintTrailer を、プラグインを利用して呼び出すためのユーザインタフェースを示す。図中、左のウィンドウは FootprintTrailer インストール済みのウェブブラウザで、ウィンドウ右下の Footprint アイコンが FootprintTrailer の起動ボタンである。図中、右のウィンドウは FootprintTrailer を起動後のウェブブラウザで、ウィンドウの左下の外枠はウェブ探索過程の記録、右下の外枠はウェブページの再訪問をそれぞれ開始するためのボタンの役割をもつ。ユーザにとって慣れているウェブブラウザの操作性を維持するため、Footprint アイコン以外をウェブブラウザの外枠に割り当てた。

図3は、Yahoo! Japan のサイトでのニュース記事の探索における、ウェブ探索およびウェブページ再訪問での FootprintTrailer 起動後の操作例である。図3(a)はウェブ探索での操作を示している。左下の外枠をクリックして探索を開始し、興味のあるウェブページが見つかったら探索完了としてウィンドウ左下の外枠を再度クリックする。図3(b)は再訪問での操作を示している。再訪問時は、ウィンドウ右下の外枠をクリック後、ファイル選択ダイアログから、再訪問したいウェブページの Footprint を選択する。ユーザは、ウェブ探索を完了するまでのリンクのクリックや検索エンジンへのキーワード入力を手動で行うが、再訪問では FootprintTrailer が自動的にこれらの操作を行うため、FootprintTrailer 起動後、再訪問に必要なユーザの操作は任意の Footprint を選択するだけである。

ウェブ探索時に FootprintTrailer を起動しておくことで、望むウェブページを見つけた段階で、ウェブサイトのホームページなど、ウェブ探索を開始したウェブページから、見つけたウェブページまでの Footprint がバックグラウンドで記録されている。ウェブブラウザのウィンドウ左下の外枠をクリックし、後で再訪問したいウェブページとして登録する。Footprint は外部ファイルとして自動的に保存される。ウェブページの再訪問では FootprintTrailer 起動後、ウェブブラウザのウィンドウの右下の外枠をクリックすることで表示されるダイアログから、Footprint を選択する。Footprint の選択後、FootprintTrailer はウェブ探索開始のウェブサイトから再訪問対象のウェブページへとリンクのクリックや必要に応じてユーザ名やパスワードなどを入力しながら、再訪問したいウェブページに至るまでの個々のウェブページを自動的に参照してゆく。

4.2 受け渡しモジュール

ウェブブラウザでは、内部でウェブページの読み込みが完了しているかどうかを判断する情報を JavaScript などから取得できる。この情報を FootprintTrailer へ受け渡すことで、リンクのクリック等の操作でウェブブラウザと同期をとることができ、FootprintTrailer による再訪問を安定して

行える。ウェブブラウザの状態取得は、JavaScript によるウェブブラウザの、ウェブページ読み込み完了情報の参照と、その参照結果を、FootprintTrailer に受け渡すプログラムの呼び出しの 2 段階に分けられる。

JavaScript の ready 関数を利用することで、ウェブブラウザが現在ウェブページの読み込みを完了しているかの情報を取得できる。FootprintTrailer への受け渡しでは、JavaScript から直接受け渡すことも考えられるが、ウェブページ読み込み状況を保持するアプリケーションプログラムを用意し、このプログラムを JavaScript から起動することで、JavaScript で得たウェブブラウザのウェブページ読み込み状況の情報を FootprintTrailer に受け渡すことができる。FootprintTrailer に受け渡すプログラムを MS Windows 上で稼働するアプリケーションとして実装することで、同様に MS Windows 上で稼働するアプリケーションであるウェブブラウザ上のリンククリック等の情報と同じ仕組みで、ウェブページ読み込み状況を FootprintTrailer が取得できるようになる。

FootprintTrailer では、再訪問時にウェブページ間を移動することに状態を受け渡し上記のプログラムと接続を行い、ウェブページの読み込み完了を示す情報を取得できた時に、再訪問を進めるのに必要な操作をウェブブラウザに送信し、読み込み完了の情報を取得できなかった時や受け渡しプログラムが起動していない場合は、再訪問を一時中断する。この手法により、FootprintTrailer は従来のウェブブラウザとのクリック操作等の操作情報の受け渡しに利用するのと同じ仕組みで、新たにウェブページ読み込み状況の情報を利用できる。

5. 実験的考察

本稿で提案した、FootprintTrailer とウェブブラウザとの連携機構によるメタ知識に基づくウェブ探索の支援について、ユーザビリティ評価を行った。実験では大学生 12 名を被験者とし、大学の教務情報システムでのアルバイトの求人に関する掲示板と、Yahoo! Japan のウェブサイトからのニュース記事の 2 種のウェブページについて、ウェブ探索および再訪問を行ってもらった。教務情報システムでのウェブ探索では、ウェブサイト内の掲示板に紹介されているアルバイトの求人情報一覧から、興味のある求人情報ページを探してもらった。Yahoo! Japan では、ウェブサイト内のニュースカテゴリから興味のあるニュース記事を探してもらった。いずれのページも過去に訪問したことがあり、日常的にこれらを再訪問する状況を想定した。教務情報システムでは、ログインのためのユーザ名とパスワードが必要であった。その後、それぞれのウェブサイト上で、見つけた記事を FootprintTrailer を利用して再訪問してもらい、FootprintTrailer の操作性を評価してもらった。

ウェブ探索時とウェブページ再訪問時それぞれで、どの程度被験者による操作を必要としたか調べた。最後にアンケートに回答してもらい、ユーザビリティを評価した。アンケートの質問は以下の 4 つである。それぞれ 5 段階のリッカート尺度（非常に同意できる: 5, 同意できる: 4, どちらでもない: 3, 同意できない: 2, 全く同意できない: 1）で評価してもらった。

質問 1: FootprintTrailer の起動が分かりやすかった

表 1 ウェブ探索および再訪問における操作コスト

	FootprintTrailer による再訪問	ウェブ探索
教務情報システム	2.0 (0.0)	19.2 (0.2)
Yahoo! Japan	2.0 (0.0)	2.5 (0.5)

* 括弧内の数値は標準偏差

** 教務情報システムでは、ログイン時のキークリックも含む

表 2 アンケート回答結果

	スコア				
	5	4	3	2	1
質問1	7	5	0	0	0
質問2	6	5	1	0	0
質問3	6	5	1	0	0
質問4	2	8	2	0	0

質問 2: FootprintTrailer の操作が分かりやすかった

質問 3: 全体的に使いやすかった

質問 4: 今後もこのシステムを利用したい

表 1 に、ウェブ探索時とウェブページ再訪問時の被験者の操作コストの平均を示す。表中、FootprintTrailer による再訪問では、ユーザの操作は FootprintTrailer の起動および Footprint の選択のみのため、再訪問の対象ウェブページに関係なく操作は 2 回で済んでいる。教務情報システムでは、アルバイト以外の、学生呼び出しや奨学金に係る連絡事項などと混在したリストからの探索であったため、操作コストが上がっていた。ニュース記事のウェブ探索では、ログインがないのに加え、Yahoo! Japan はポータルサイトとして、様々なニュースを紹介しているため、ニュース記事を見つけるまでの操作数が少なく、相対的に FootprintTrailer によるコストの低減度が小さかった。また、Yahoo! Japan ではニュース記事に至るまでの経路が多いため探索の操作コストの個人差が大きかった。

アンケートの回答結果を表 2 に示す。表の値は、各質問の各スコアについて、何人の大学生がどのスコアに回答したかを示している。表中、質問 1, 2 についてはすべての被験者から 4 以上の同意回答を得られた。また、全ての質問について、ウィルコクソンの符号順位検定で統計的に有意に同意に属する回答が多かった ($p < 0.05$)。

実験結果から、FootprintTrailer を、プラグインを介してウェブブラウザと連携させることで、ユーザがメタ知識によるウェブ探索あるいは再訪問の際に容易に呼び出せるようにする機構は、独自のユーザインタフェースを用いる既存の手法に比べて、ユーザに理解しやすいユーザインタフェースで、ウェブ探索およびウェブページの再訪問を支援するのに有効であった。

また、連携機構について、「ログイン名、パスワードも知識として利用できて便利だった」、「ウェブ探索過程の記録の取り方など、FootprintTrailer の操作が分かりやすかった」など、肯定的な意見があり、ユーザが提案手法によるウェブページ再訪問を便利と判断したことが分かる。

一方で、「Footprint の記録中かを示す情報や再訪問の過程が分かりにくかった」、「再訪問するウェブページの指定が分かりにくかった」というコメントも得られたことから、ウェブ探索や再訪問の過程での入力操作の見やすさや、Footprint 自体のユーザへの提示方法の工夫が必要であることが分かった。前者の FootprintTrailer の再訪問過程については、クリックしたリンクアンカーのズームングや別ウィンドウ表示を検討している。後者の Footprint の提示については、記録時に抽出したウェブページのタイトルをリスト化し、ユーザが視覚化に再訪問対象を選べるユーザインタフェースを整備してゆく。

6. おわりに

本稿では、FootprintTrailer とウェブブラウザとの連携の必要性について述べ、FootprintTrailer をウェブブラウザから利用できる連携機構を提案した。この仕組みは FootprintTrailer をウェブブラウザから呼び出すスクリプトを実装するとともに、ウェブブラウザのウェブページ読み込み状況を取得し FootprintTrailer に受け渡す機構を実装することで、実現できる。評価実験では、FootprintTrailer とウェブブラウザとの連携機構を実装したウェブブラウザを利用した、メタ知識に基づくウェブ探索および知識に基づく再訪問を実際に大学生に行ってもらった。結果として、提案手法のユーザビリティについて高い評価を得たとともに、再訪問過程の視覚化などの面で改善が必要なが分かった。

今後は、この機構を広く利用できるように、様々なウェブブラウザで利用可能なプラグインの実装を進めていながら、公開ウェブサイトを通じて容易にダウンロード、インストールできる仕組みを整備することで、ユーザがこの機構を利用してウェブ探索を効率的に行うための環境を構築してゆく。また、マウスジェスチャなどの、ウェブブラウザ固有の操作にも対応し、個々のユーザによってカスタマイズされたウェブブラウザでも FootprintTrailer を使えるようにしてゆく。

参考文献

- [1] Y. Abe, K. Matsusako, K. Kirimura, M. Tamura, M. Nakashima and T. Ito: Tolerant sharing of a single-user application among multiple users in collaborative work, Companion Proc. CSCW2010, Savanna Georgia, USA, pp. 555-556, Feb. 6-10, 2010.
- [2] R. Brunelli, Template matching techniques in computer vision: theory and practice, Wiley, 2009.
- [3] O. Hoerber and J. Gerner: BrowseLine: 2D timeline visualization of Web browsing histories, Proc. IV'09, Barcelona, Spain, pp. 156-161, July 15-17, 2009.
- [4] R. Kawase, G. Papadakis, E. Herder and W. Nejdl: Beyond the usual suspects: context-aware revisitation support, Proc. HT'11, Eindhoven, The Netherlands, pp.27-36, June 6-9, 2011.
- [5] C.E. Kulkarni, S. Raju and R. Udupa: Memento: unifying content and context to aid webpage re-visitation, Proc. UIST'10, New York, NY, USA, 2010, pp 435-436, Oct. 3-6, 2010.
- [6] D. Morris, M.R. Morris and G. Venolia: SearchBar: a search-centric web history for task resumption and information re-finding, Proc. CHI'08, Florence, Italy, pp. 1207-1216, Apr. 5-10, 2008.
- [7] S. Sakamoto, M. Nakashima and T. Ito: Dodging window interference to freely share any off-the-shelf application among multiple users in co-located collaboration, Proc. HCI'11, Orlando, Florida, USA, pp. 305-314, July 9-14, 2011.
- [8] K. Sato, Y. Adachi, M. Nakashima and T. Ito: A Mechanism of Trailing the Footprint for the Previously Visited Web Pages to Ease a Meta-knowledge-based Search, Proc. NBIS2012, Melbourne, Australia, pp.298-305, Sept. 2012.
- [9] J. Šimko, M. Tvarozek and M. Bielikova: Semantic history map: graphs aiding web revisitation support, Proc. DEXA'10, Bilbao, Spain, pp. 206-210, Aug. 30 – Sept. 3, 2010.
- [10] J. Teevan, E. Cutrell, D. Fisher, S.M. Drucker, G. Ramos, P. Andre and C. Hu: Visual snippets: summarizing web pages for search and revisitation, Proc. CHI'09, Boston, Massachusetts, USA, pp.2023-2032, Apr. 4-9, 2009.
- [11] S.S. Won, J. Jin and J.I. Hong: Contextual web history: using visual and contextual cues to improve web browser history, Proc. CHI'09, Boston, Massachusetts, USA, pp.1457-1466, Apr. 4-9, 2009.