

修 士 論 文

題 目

計算機上の仕事状態の保存と復元機能に関する研究

研究室担当教員の署名

提 出 者

小笠原 良

岡山大学 大学院自然科学研究科 電子情報システム工学専攻

平成 21 年 2 月 3 日 提出

# 要約

利用者が仕事で利用するアプリケーションソフトウェア (AP) は、利用者の仕事の再開を支援するために、過去に編集した文書ファイルなどの履歴情報を提供している。ここで、複数の AP の統一的な履歴情報を提供できれば、計算機内で過去に参照したデータの想起と参照を効果的に行うことができる。しかし、統一的な履歴情報を提供する既存のツールは、利用者に統一的なファイルアクセス履歴情報を提供するものの、履歴情報を具体的な仕事単位で集約できない。このため、履歴情報から個々の仕事の途中状態を想起することは難しい。そこで、私は、計算機上の仕事状態の保存と復元機能を提案する。提案機能は、履歴情報を利用者の意識できる仕事の単位で集約して提示する。本論文では提案機能の一実装である「デスクトップブックマーク」を利用することで、仕事状態から仕事の途中状態を容易に復元できるか、関連ツールと比較考察した。具体的には、各ツールが統一的履歴情報として取得したパスに着目し、評価尺度としてデータの再現率と適合率を導入した。そして、デスクトップブックマークは関連ツールと比較して、データの再現率は同程度であるものの、データの適合率は大きく改善されることを示した。

# 目次

1	はじめに	1
2	統一的履歴情報	3
2.1	概要	3
2.2	解決できる問題	3
2.3	取得方法	4
3	計算機上の仕事状態の保存と復元	5
3.1	既存手法の問題点	5
3.2	要求と対処	6
3.2.1	要求	6
3.2.2	仕事状態の保存機能	6
3.2.3	仕事状態の復元機能	7
3.3	提案機能	7
4	実現方式	8
4.1	実現位置	8
4.2	仕事状態として保存する情報	9
4.3	仕事状態の保存契機	10
5	関連研究	12
5.1	データ管理手法	12
5.1.1	データの分類	12
5.1.2	データ内容からの検索	12
5.1.3	統一的履歴情報の利用	13
5.1.4	各手法の比較	13
5.2	仕事を意識した統一的履歴情報の提示手法	14

<b>6 システム設計</b>	<b>15</b>
6.1 デスクトップブックマーク . . . . .	15
6.2 計算機外部の履歴情報の取得方法 . . . . .	15
6.2.1 概要 . . . . .	15
6.2.2 Private Proxy Server ログフィルタの設計 . . . . .	17
6.3 計算機内部の履歴情報の取得方法 . . . . .	18
6.3.1 概要 . . . . .	18
6.3.2 デスクトップブックマークへの要求 . . . . .	20
6.3.3 対処 . . . . .	20
6.3.4 更新履歴管理システムへの要求 . . . . .	21
6.3.5 利用する更新履歴管理システム . . . . .	21
6.4 保存するデータの選定規則 . . . . .	22
6.5 システム構成 . . . . .	23
6.6 動作概要 . . . . .	25
<b>7 評価</b>	<b>27</b>
7.1 比較対象研究 . . . . .	27
7.1.1 Google Desktop . . . . .	27
7.1.2 俺デスク . . . . .	28
7.2 評価概要 . . . . .	29
7.2.1 データの再現率と適合率 . . . . .	29
7.2.2 環境 . . . . .	30
7.2.3 手順 . . . . .	31
7.3 評価結果 . . . . .	31
7.3.1 評価 1：選定規則の閾値の妥当性 . . . . .	31
7.3.2 評価 2：仕事状態復元の容易さ . . . . .	34
<b>8 おわりに</b>	<b>38</b>
<b>謝辞</b>	<b>39</b>
<b>参考文献</b>	<b>40</b>
<b>発表論文</b>	<b>43</b>

# 目 次

2.1	統一的履歴情報 . . . . .	4
3.1	仕事状態 . . . . .	6
6.1	通信履歴の把握場所 . . . . .	16
6.2	システム構成 . . . . .	23
7.1	仕事状態の提示例 . . . . .	30
7.2	評価 1 の結果 . . . . .	32
7.3	評価 1 の結果 (個人) . . . . .	34

# 表 目 次

4.1	実現位置の比較 . . . . .	9
6.1	対処方法の評価 . . . . .	16
6.2	計算機内部の履歴情報取得位置 . . . . .	19
6.3	更新履歴管理システムの比較 . . . . .	21
7.1	デスクトップブックマークと関連ツールとの比較結果 . . . . .	35
7.2	各関連ツールのデータの再現率と適合率 . . . . .	35
7.3	各関連ツールのデータの再現率と適合率 (個人) . . . . .	36

# 第 1 章

## はじめに

計算機を利用して仕事を進める機会は増加している。利用者は、仕事で様々なアプリケーションソフトウェア（以下、AP）を利用する。例えば、利用者はエディタ、表計算ソフト、Web ブラウザなどの AP を仕事で利用する。これらの AP は、利用者の仕事の再開を支援するために、過去に編集した文書ファイルなどの履歴情報を提供している。履歴情報を提供する利点は、利用者にとって過去に参照したデータ（文書ファイルや Web ページ）の想起と参照が容易になる点である。ところが、多くの場合、これらの履歴情報は個々の AP に閉じており、複数の AP の履歴情報を同時に提供できない。利用者は多くの AP を組み合わせて仕事をしているため、一連の仕事に最適化された統一的な履歴情報を利用者に提供することは難しい。この問題を解決するためには、個々の AP の持つ履歴情報を収集して統一的に管理し、さらに集約して利用者に提示する仕組みが必要となる。

このような背景から、統一的履歴情報を提供するツールがいくつか提案されている [1][2][3]。これらのシステムの統一的履歴情報、および断片的に思い出せる利用者の記憶を組み合わせることで、利用者は失われた記憶を補完できる。具体的には、利用者は過去の計算機上での活動履歴を探索することで、過去に参照したデータの想起と参照に役立つ情報を取り出せる。しかし、これらのツールは、統一的履歴情報を具体的な仕事単位で集約できない。このため、統一的履歴情報から個々の仕事の途中状態を想起することは、利用者である人間の記憶に委ねられており、利便性に難がある。

一方、ある工作中に別の急な仕事が入り、現在行っている仕事を一時的に、例えば 2、3 日の間、中断することがある。仕事を再開するためには、仕事を中断する際どのようなデータを参照していたか記憶しておく必要がある。ここで、仕事を再開する際に、統一的履歴情報は仕事の中断時に参照していたデータの想起を支援できる。しかし、既存の統一的履歴情報

を提供するツールは、統一的履歴情報からデータを具体的な仕事単位で一括して再参照できない。このため、統一的履歴情報を用いて仕事を再開する場合、利用者は仕事を中断した際に参照していたデータを手作業で一つずつ再参照していく必要があり、利便性に難がある。

これらの問題に対処するため、私は計算機上の仕事状態の保存と復元機能を提案する。提案機能は、過去に行った仕事内容と統一的な履歴情報との関連付けを利用者の任意の時点でブックマークのように記録する。また、再開したい仕事の中にデスクトップ上で起動していた AP の起動、および参照していたデータの参照を一括して実行できる。



## 第 2 章

# 統一的履歴情報

## 2.1 概要

統一的履歴情報とは、複数の AP の履歴情報 (文書ファイルへのアクセス情報や操作情報) を共通する API で統一的に提供したものである。統一的履歴情報の基本的な考え方を図 2.1 に示し、以下に説明する。

まず、AP の履歴情報を利用した従来手法 (図 2.1 中の (a)) では、履歴情報は個々の AP によって個別に提供されている。このため、複数の AP にまたがった履歴情報を扱うことは、利用者に繁雑な操作を要求する。

一方、統一的履歴情報を利用した手法 (図 2.1 中の (b)) では、個々の AP の履歴情報を統一的インタフェースで提供する。このため、複数の AP にまたがった履歴情報を簡単に扱えることから、利用者は過去に参照したデータを容易に想起し、参照できる。例えば、エディタで参照した文書ファイルの履歴情報と、Web ブラウザで参照した Web ページの履歴情報との関連を時刻を基準に扱える。このように、統一的履歴情報は利用者にとって有益である。

## 2.2 解決できる問題

統一的履歴情報を利用することで解決できる問題について、人間の記憶の観点から明らかにする。

Margery Eldridge らは、仕事に取り組む間に直面する人間の記憶に関する問題を「回想的記憶問題」「展望的記憶問題」「アクションスリップ」の 3 つに分類した [4]。回想的記憶問題は、過去に記憶した情報を忘れる問題である。展望的記憶問題は、今後の予定を忘れる問題

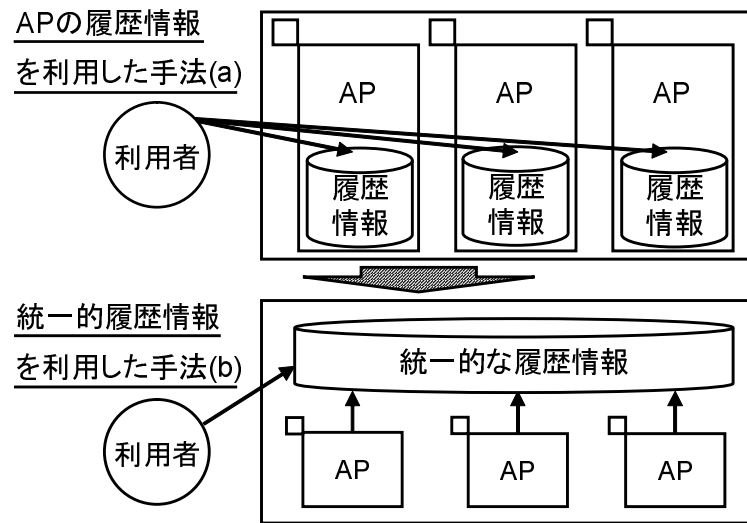


図 2.1 統一的履歴情報

である．アクションスリップは現在行っている行動で問題を引き起こす短期記憶の失敗である．統一的履歴情報を取得する研究はいずれも，文書に関する回想的記憶問題の解決に着目している．

## 2.3 取得方法

利用者のデータ参照履歴の収集方法に関しては，関連論文でよく議論されている．

MyLifeBits[1] は人の生涯において価値のある情報を記録し、管理するためのシステムである．MyLifeBits は，利用者の Web ページ参照履歴，電話の通話記録，視聴したラジオやテレビなどの情報を収集する．Memory-Organizer[2] は，Web ページ参照履歴を収集することで利用者の知識空間を作成し，利用者が過去に蓄えた知識の想起を支援するシステムである．Forget-me-not[3] は，PDA 用の記憶補助システムである．利用者の実世界におけるイベント（出会った人の ID，ワークステーションの起動履歴，文書ファイルの操作履歴，電話通話の履歴）を収集し，時間順に表示することで利用者の過去の仕事内容の想起を支援する．

このように，統一的履歴情報自体は簡単に収集できるようになり，そして蓄えられるようになった．しかし，統一的履歴情報を有効活用できるように利用者に提示するのは難しい．統一的履歴情報を収集するシステムは蓄えたデータを利用者の使いやすい形式に変換し，利用者の情報活用を促進する必要がある．

## 第 3 章

# 計算機上の仕事状態の保存と復元

### 3.1 既存手法の問題点

これまでに述べたように，統一的履歴情報は利用者にとって有益である．しかし，統一的履歴情報を扱う既存手法には以下の問題点がある．

(問題点 1) 仕事とデータとの関連性を保存できない

利用者は過去に参照していたデータを，データの名前や参照していた時刻ではなく「仕事」単位で記憶している．例えば，仕事「論文 A 執筆」中に参照した Web ページを再参照したい場合，利用者は Web ページを参照した正確な時刻は記憶していないものの，「論文 A 執筆時に参照した」といったデータ参照時に行っていた仕事に関する情報（以下，仕事情報）は記憶している．このため，過去に参照したデータを想起するには，仕事情報からデータを想起し参照できる仕組みが必要である．しかし，既存手法はいずれも利用者に統一的なアクセス履歴情報を提供するものの，これらの情報を具体的な仕事単位で集約できない．このため，過去に参照していたデータの想起は難しい．

(問題点 2) 過去の仕事を容易に再開できない

利用者は過去に行っていた仕事の続きを行うことがある．このとき，過去に行っていた仕事の中にデスクトップ上で起動していた AP を起動し，参照していたデータを再参照する必要がある．例えば，「論文 A」を発展させた内容の論文を執筆する場合，過去に論文 A を執筆していた時にデスクトップ上で起動していた AP を起動し，参照していたデータを再参照する必要がある．しかし，既存手法はいずれも仕事を再開する場合，データを 1 つずつ手作業で参照していく必要がある．このため，過去の仕事を再開するには手間がかかる．

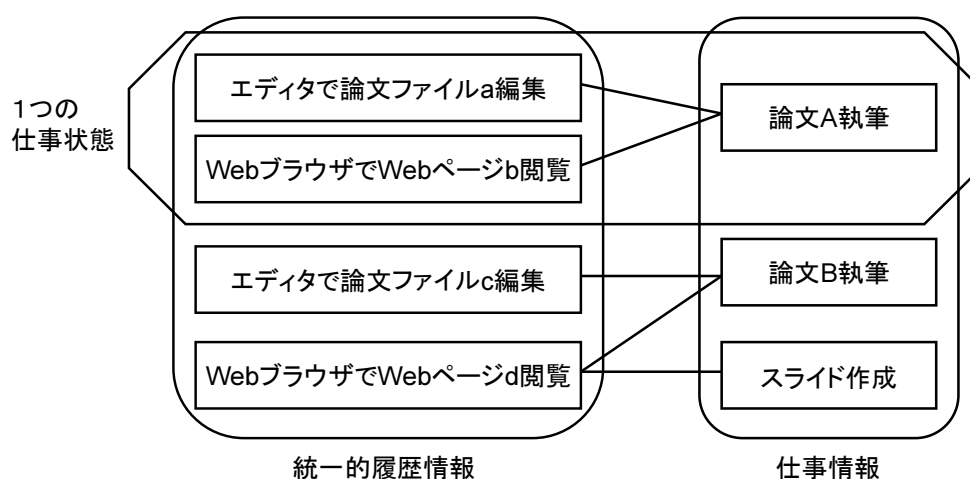


図 3.1 仕事状態

## 3.2 要求と対処

### 3.2.1 要求

統一の履歴情報を扱う既存手法の問題点から、新しい機能には以下の要求がある。

(要求 1) 仕事とデータを関連付けて仕事状態として保存

(要求 2) 過去の仕事状態を容易に復元

以降では、これらの要求への対処について述べる。

### 3.2.2 仕事状態の保存機能

(要求 1) への対処として、統一の履歴情報を仕事単位で保存する機能 (以降、仕事状態の保存機能) を実現する。ここで、仕事状態は統一の履歴情報を利用者の仕事単位で集約した情報とし、1つの仕事状態は仕事情報、および統一の履歴情報からなるとする。仕事状態の保存機能の概要について、図 3.1 を用いて、以下に説明する。例えば、「論文 A 執筆」という仕事情報に対して、「エディタで論文ファイル a 編集」、「Web ブラウザで Web ページ b 閲覧」という統一の履歴情報を関連付けて保存する。仕事情報と統一の履歴情報の具体的な内容は 4.2 節にて議論する。

### 3.2.3 仕事状態の復元機能

(要求 2) への対処として、過去に行っていた仕事中にデスクトップ上で起動していた AP の起動、および参照していたデータの参照を一括して行う機能 (以降、仕事状態の復元機能) を実現する。仕事状態の復元機能の概要について、図 3.1 を用いて、以下に説明する。例えば、「論文 A 執筆」という仕事情報に関連付けられた統一的履歴情報を用いてエディタと論文ファイル a、および Web ブラウザと Web ページ b の起動と参照を一括して行う。

## 3.3 提案機能

上記の要求への対処を満たす機能として、計算機上の仕事状態の保存と復元機能を提案する。計算機上の仕事状態の保存と復元機能の目的から、提案機能には以下の特徴を持たせる。

(特徴 1) 統一的履歴情報を仕事単位で集約した「仕事状態」を保存できる

(特徴 2) 仕事状態から仕事の途中状態を容易に復元できる

(特徴 3) 複数の仕事状態をブックマークのように保存管理できる

仕事状態の保存機能の目的は、過去に参照していたデータの想起にある。1 つの状態しか保存できない場合、過去の仕事情報をさかのぼって検索できない。このため、複数の仕事状態をブックマークのように保存管理可能なシステムを実現する。

(特徴 4) 過去に参照したデータが更新された場合、データ内容は最新のものを再参照できる

仕事状態の復元機能の目的は、計算機の状態を元に戻すことではなく、仕事の中断と再開を支援することである。つまり、ハードウェアや OS の状態の復元が目的でなく、過去に参照していたデータの再参照を目的としている。このため、過去に参照したデータが更新された場合、データ内容は最新のものを再参照可能なシステムを実現する。

(特徴 5) Web アクセス履歴に対応できる

インターネットを利用して仕事をする機会は増加している。このため、計算機内のファイルアクセスだけでなく Web アクセス履歴にも対応させることで、利便性の向上を図る。

## 第 4 章

# 実現方式

### 4.1 実現位置

提案手法の実現位置として、以下の 3 つが考えられる。

#### (1) OS による実現

ハイバネーション機能に代表されるように、仕事を途中で中断したい時、計算機上の仕事途中状態を完全に保存し、復元可能な機能を作成できる。また、チェックポイントリスタート [5] に代表されるように、保存した状態から AP の走行を再開できるように、走行中の AP の状態を保存可能にする機能を作成できる。

#### (2) 仮想計算機による実現

VMware[6] のスナップショット機能に代表されるように、仕事を途中で中断したい時、仮想計算機上の仕事途中状態を完全に保存し、復元可能な機能を作成できる。

#### (3) AP による実現

Google Desktop [7] や俺デスク [8] に代表されるように、過去に参照したデータの想起と参照を支援する機能を作成できる。

これらの比較結果を表 4.1 に示し、以下で詳細を述べる。

OS や仮想計算機上にて機能を提供する場合、OS の管理する計算機情報 (ハードウェアとソフトウェア両方の情報) を利用できるため、ハードディスクや OS の状態といった計算機上の全ての状態を保存可能な機能を作成できる。しかし、計算機上の全ての状態を過去の状

表 4.1 実現位置の比較

	OS	仮想計算機	AP
収集可能な情報量が多い			
導入にかかる工数が少ない	×	×	

態に復元する場合、「過去に参照したデータが更新された場合、データ内容は最新のものを再参照できる」という特徴を実現できない。これにより、例えば、OS のセキュリティアップデートを行う前の状態に復元すると、OS の状態まで元に戻るため、セキュリティ上問題となる。また、OS や仮想計算機上にて提供する場合、導入時に新しく作成した OS(もしくは仮想計算機)をインストールし、計算機環境を再構築する必要がある。このため、機能の導入にかかる工数は多い。

一方、AP にて機能を提供する場合、利用できる情報は AP で抽出可能な情報のみであるため、OS や仮想計算機上に実現する場合と比較して、収集可能な情報量は劣る。しかし、OS の提供する情報の取得、および利用者の操作履歴情報の取得により、個々の AP の持つ履歴情報と同等の情報を収集できる。このため、AP レベルにおいても提案機能に必要な情報を十分に収集可能といえる。また、AP にて提供されるを導入する場合、導入にかかる工数は既存の環境に新しい AP をインストールする作業のみである。このため、機能の導入にかかる工数は少ない。

導入にかかる工数が少なく、かつ個々の AP の持つ履歴情報と同等の情報を収集できることから、提案機能は AP として実現する。

## 4.2 仕事状態として保存する情報

ここでは、3.2.2 項で述べた仕事状態を具体化する。提案機能が保存する仕事状態は、利用者に提示する情報である。このため、提案機能は利用者に提示する目的で以下の情報を保有する。

- (1) 「仕事状態」は「仕事情報」と「統一的履歴情報」からなる
- (2) 「仕事情報」は以下の情報を含む

(2-a) 仕事名

仕事を一意に識別可能な名前を提示することにより，仕事を想起する引き金の役割を果たす．

#### (2-b) スケジュール

仕事を行った時刻を提示することにより，時系列順に考えた仕事の想起を可能にする．

### (3) 「統一的履歴情報」は以下の情報を含む

#### (3-a) データのパス (文書ファイル名や URL)

参照したデータのパスを提示することにより，過去に参照していたデータの再参照を可能にする．

#### (3-b) データのイメージ図 (サムネイル)

仕事中に参照していたデータのイメージ図を提示することにより，文字のみを提示した場合と比較して，より直感的にデータ内容を把握可能にする．

統一的履歴情報として保有する情報として，上記に加えてデータそのものの内容（以下，コンテンツ）が考えられる．しかし，提案機能自身は，コンテンツを保存対象としていない．なぜなら，コンテンツの保存に関しては既存ツール（更新履歴管理システム）が数多く存在しており，コンテンツの管理が必要な場合は別途これらの専用ツールが利用できるためである．更新履歴管理システムの具体例として，ファイルシステムの復元機能である Mac OS X Leopard の Time Machine [9] やバージョン管理システムである Subversion [10] がある．提案機能は，利用者の過去に参照していたデータの想起と参照の手助けを行うための項目のみを保存対象としている．

## 4.3 仕事状態の保存契機

仕事状態の保存契機として，以下の 2 つが考えられる．

### (1) 計算機が自動で仕事状態を保存

利用者が意識することなく自動で仕事状態を保存するため，仕事状態を確実に保存できる．しかし，利用者の「仕事」を意識した仕事状態の保存はできないため，後の検索精度が低下する．



## (2) 利用者が手動で仕事状態を保存

利用者が「仕事」を意識して Web のブックマークを作成する要領で仕事状態を保存できる。しかし、利用者が手動で仕事状態を保存するため、仕事状態を保存し忘れるおそれがある。

既存手法は計算機が自動で統一的履歴情報を保存する。しかし、「仕事」を意識するためには、利用者が手動で仕事状態を保存する必要がある。そこで、提案機能は Web ページのブックマークのように、「現在の仕事状態を保存」という操作を利用者に求める。

## 第 5 章

## 関連研究

### 5.1 データ管理手法

#### 5.1.1 データの分類

計算機上の文書ファイルを管理するため、フォルダ階層やファイル名による分類が行われている。しかし、文書ファイルを厳密に分類する方法は、データ管理に関する利用者の負担が大きい。また、複数のカテゴリに属しているような文書ファイルの分類は困難である。このため、情報を厳密に分類し管理するのではなく、より直感的で親しみやすい方式が求められるようになっている。

このような背景から、データの管理手法について、様々な研究が行われている。

実世界での利用者の仕事においては、空間的な位置による文書管理が多く利用されている[11]。計算機上においても、データのアイコンを計算機画面の中に配置して整理することは一般的に行われている。デスクトップは、場所に基づいた情報整理が便利だという理由で多くの利用者に使われている。例えば、やりかけの仕事に関するデータは、デスクトップの目に付く場所に置かれることが多い[12]。しかし、デスクトップに配置可能なデータ数はさほど多くなく、このような空間的な位置による分類は拡張性に問題がある。

#### 5.1.2 データ内容からの検索

大量のデータを厳密に分類することは困難であると、多くの論文で述べられている。この対処として、検索システムによってデータを管理する方法がある。

人気のある Web ページ検索サービスは、利用者計算機上から情報を発見するために、デスクトップ検索システム [13] を提供している。デスクトップ検索システムは、利用者の記憶しているキーワードから利用者計算機上の全データに対して全文検索し、検索結果を利用者に提示する。利用者は検索結果を基に、過去に参照したデータを再参照できる。

これは、抽象的に考えると、データ内容を基にデータを自動で分類しているといえる。

### 5.1.3 統一的履歴情報の利用

5.1.2 項でも述べたように、大量のデータを厳密に分類することは困難である。この対処として、2 章で述べた統一的履歴情報を時系列に視覚化する方法がある。

「超整理法 [14]」は現実のデータ (書類) において、時系列にデータを並べて情報を整理しようとする提案である。これは、計算機上のデータにも適応できる結論を出している。

いくつかの研究は Web ページの参照履歴に着目している。このアプローチの典型例として、WebTop [15] がある。WebTop は利用者のブックマーク、Web ページの参照履歴、および参照した Web ページの内容を基にデータの分類を行う。また、Temporally-threaded Workspace [16] は、プロキシサーバを用いることで、利用者に特別な操作を求めずに、複数の利用者の Web 閲覧履歴をまとめて保存する。

LifeStreams [17] は、超整理法 [14] に似た発想で、文書ファイルの内容やタイトルに関わらず、全てを時間順の 1 次元の列に並べて管理する。時間を通してワークスペースを視覚化する他のシステムとして、TimeScope [18] がある。このシステムは効果的なデータの組織化の手段として、時間軸で仮想化されたデスクトップの変動を提示する。

これは、抽象的に考えると、時刻を基にデータを自動で分類しているといえる。

### 5.1.4 各手法の比較

利用者自身の分類によるデータ管理手法は、分類規則をしっかりと決定しておけば、データの発見が容易になる。しかし、データの量の増加に伴い、データ管理にかかる手間が大きくなる問題がある。

コンテンツ内容による検索におけるデータ管理手法は、自動でデータを管理できるため、データ管理にかかる手間はほとんどない。また、検索ワードを上手く設定することで、簡単にデータを発見できる。しかし、データ同士の関連性をがわからないため、ある仕事に関連するデータを一括して想起したい要求に対処できない。

統一的履歴情報の利用におけるデータ管理手法は、自動でデータを管理できるため、データ管理にかかる手間はほとんどない。一方、データへの大体のアクセス時間を想起できないと、データを再参照できない。しかし、時系列にデータを並べて表示するため、データ同士の関連性がわかる。このため、再参照したいデータの参照中に参照していた別データを想起できれば、これを契機としてデータの想起が可能となる。

ここで、本研究の目的は、過去に参照したデータの参照だけではなく、仕事の中断と再開を支援することにもある。仕事を再開するには、再開したい仕事に参照していたデータを全て再参照する必要がある。このため、仕事を行っていた時刻を特定できるならば、ある時刻周辺で参照したデータを一覧できる「統一的履歴情報の利用におけるデータ管理手法」が仕事の中断と再開を支援するのに最適な手法であるといえる。

## 5.2 仕事を意識した統一的履歴情報の提示手法

5.1.2 項で述べた統一的履歴情報を用いたデータ管理手法に関する研究は、統一的履歴情報を時系列順に並べることにより、利用者にデータの想起を促している。一方、提案機能は「仕事」を意識して統一的履歴情報を利用者に提示する。本節では、統一的履歴情報の提示手法において、仕事を意識した研究について述べる。

統一的履歴情報の要約を利用者に提示するシステムとして、Desktop History [19] や Task-Tracer [20] がある。これらのはいずれも OS を改良してシステムを作成している。OS を改良してシステムを作成すると、システムは OS の管理する計算機情報 (ハードウェアとソフトウェア両方の情報) を利用できるため、より精密なシステムを作成できる。しかし、OS を改良してシステムを作成する場合、システム導入の際に OS を再インストールし、計算機環境を再構築する必要がある。このため、導入に掛かるコストは高くなる。一方、提案機能は 4.1 節で述べたように、AP にて機能を提供することから、導入に掛かるコストは低いと考えられる。

Google Desktop[7] は、デスクトップ上の計算機内部の文書ファイル更新履歴、および Web ページの参照履歴を関連付けて保存する。また、俺デスク [8] は、計算機内部のデータ参照をトリガとして、Web 参照履歴の検索を可能にする。これらのツールの保存する「計算機内部のデータ参照」は、一種の仕事情報とみなすことができる。俺デスク、および Google Desktop との比較については、7 章にて詳細を述べる。

## 第 6 章

# システム設計

### 6.1 デスクトップブックマーク

提案機能の一実装として、私は過去に提案した「デスクトップブックマーク」を計算機上の仕事状態の保存と復元機能として再構成した。本章の以降では、デスクトップブックマークのシステム設計について述べる。

### 6.2 計算機外部の履歴情報の取得方法

#### 6.2.1 概要

計算機外部にあるデータへのアクセス履歴情報（以下、計算機外部の履歴情報）を取得するには、外部計算機との通信情報を記録すればよい。ここで、保存する通信情報は、Web ページの閲覧履歴情報を考える。外部計算機との通信情報の把握は、図 6.1 に示す通り、以下の 3 つの方法が考えられる。

#### (対処 1) AP による通信情報の把握

Web ブラウザをデスクトップブックマーク用に修正し、Web ブラウザから通信情報を取得する方法である。これは Web ブラウザの改良を必要とする。このため、利用者がどの Web ブラウザを利用しても通信情報を把握できるようにするには、大きな工数を要する。一方、利用者計算機上で情報を扱うため、機密情報の扱いを含む仕事も安全に扱える。

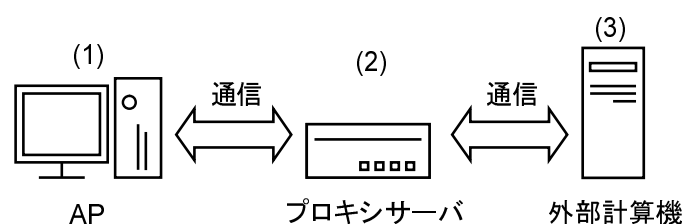


図 6.1 通信履歴の把握場所

表 6.1 対処方法の評価

	対処 1	対処 2	対処 3
実現の簡単さ	×		×
情報の取り扱いの安全さ		×	×

対処 1：AP による通信情報の把握

対処 2：プロキシサーバによる通信情報の把握

対処 3：外部計算機による通信情報の把握

#### (対処 2) プロキシサーバによる通信情報の把握

Web ブラウザと外部計算機との間の通信をプロキシサーバを仲介させ、プロキシサーバから通信情報を取得する方法である。Web ブラウザの設定をプロキシサーバ経由に変更するだけで導入可能であるため、簡単に実現できる。一方、利用者計算機外に情報を送信するため、情報の取り扱いは安全でない。

#### (対処 3) 外部計算機による通信情報の把握

外部計算機から通信情報を取得する方法である。利用者計算機の設定変更は不要であるものの、特別な外部計算機を用意する必要がある。また、利用者計算機外に情報を送信するため、情報の取り扱いは安全でない。

各対処の比較を表 6.1 に示す。どの対処においても実現の簡単さと機密情報の安全さの両方を満足することは難しい。しかし、プロキシサーバを利用者計算機上に設置できれば、プロキシサーバを用いる際の欠点を解消できる。

上記の理由より、デスクトップブックマークでは、(対処 2) であるプロキシサーバによる方法を用いる。つまり、外部計算機との通信情報は、利用者計算機上にプロキシサーバを設置し把握することにより取得する。これにより、計算機外部の履歴情報を安全、かつ容易に取得できる。

### 6.2.2 Private Proxy Server ログフィルタの設計

利用者計算機上に設置したプロキシサーバから相手計算機との通信情報を取得する機構を実装する。

ここで、利用者計算機上に設置するプロキシサーバの一実装として、HTTP プロトコルに着目したプロキシサーバ(以降、Private Proxy Server)を実装する。今回は計算機外部の履歴情報として「Web ブラウザを利用した Web 閲覧履歴」を保存することを目的とする。

ここで、Private Proxy Server の保存する Web アクセス履歴情報は、「Web ブラウザを利用した Web 閲覧履歴」を把握する上で不要な情報がある。一般的な Web ページ(以降、メインページ)は画像など他の Web ページ(以降、サブページ)を参照していることが多い。メインページを GET メソッドで取得する場合、メインページに付随するサブページを取得するために GET メソッドが呼ばれる。Private Proxy Server の Web アクセス履歴情報は「HTTP リクエストの実行」に関するログであることから、メインページ(ユーザのクリックしたページ)の URL の他に、サブページの URL も一緒に保存してしまう問題がある。

このような問題を解決するために、Private Proxy Server の Web アクセス履歴情報からメインページの URL を抽出するフィルタを作成する必要がある。既存のフィルタは、以下のような手順で「Web ブラウザを利用した Web 閲覧履歴」を抽出する [21][22]。

- (1) リファラとなっている URL を抽出し、そのリファラがリクエストされた時の履歴を MainURL 参照履歴候補とする。リファラが重複している場合、先に参照された履歴のみを扱う。またリファラが “-” である履歴は後で扱うのでここでは除外しておく。
- (2) 深い階層構造を判別するために、MainURL 参照履歴候補とその 1 つ前の履歴との参照時間差を見て、両者を参照した時の時間差が利用者が URL を直接参照する間隔より十分に短い場合、その MainURL 参照履歴候補は深い改造構造の TOP ではないと判別し、MainURL 参照履歴候補から除外する。
- (3) MainURL 参照履歴候補にリファラが “-” の時の履歴を加え、MainURL 参照履歴候補同士の参照時間差を利用者の閲覧時間と見て、利用者の閲覧時間が十分に長ければ、その MainURL 参照履歴候補を閲覧 URL 履歴候補と判定する。このとき、一番最後の MainURL 参照履歴候補は利用者が最後に閲覧した URL なので閲覧 URL 履歴候補と考える。
- (4) 閲覧 URL 履歴候補からサービス状態コードが “200” 以外の履歴を除外する。
- (5) 残った閲覧 URL 履歴候補のリクエストを利用者が閲覧した URL と判別する。

しかし，既存手法では，メインページの URL の見逃しが多く発生した．また，サブページ URL を多く誤検出した．ここで，Private Proxy Server からメインページの URL を抽出する際に起こる問題について以下で述べる．

(問題 1) サービス状態コード 304 のページを抽出できない

既存フィルタはサービス状態コードが 200 以外の履歴を除外する．このため，Web ブラウザの持つキャッシュと比較して，Web ページが更新されていない場合に返ってくる 304(Not Modified) にマッチしない問題がある．

(問題 2) Content-Type を意識しない

利用者が閲覧する Web ページの多くは Content-Type が決まっている (text/html である) と考えられる．しかし，現在は Content-Type を意識していないため，CSS ファイルや XML ファイルなど，サブページを多く取得してしまう問題がある．

これらの問題を解決するため，Private Proxy Server ログフィルタの抽出条件を以下のように変更する．

(条件 1) リファラになっている履歴 or リファラが<sup>1)</sup>である履歴

(条件 2) サービス状態コードが 200 or 304 である履歴

(条件 3) Content-Type が text/html である履歴

(条件 4) < 前の候補ページ > を参照してから一定時間経過している履歴

## 6.3 計算機内部の履歴情報の取得方法

### 6.3.1 概要

計算機内部にあるデータへのアクセス履歴情報 (以下，計算機内部の履歴情報) を取得する方法として，以下の 3 つが考えられる．

(1) 個々の AP の持つ履歴情報の利用

(2) 個々の文書ファイルの持つ情報 (最終アクセス時間) の利用

(3) OS の提供する情報 (最近使った文書ファイルの最終アクセス時間) の利用



表 6.2 計算機内部の履歴情報取得位置

	AP	文書ファイル	OS
実装にかかる工数の少なさ	×		
仕事状態保存のオーバーヘッドの少なさ		×	

個々の AP の持つ履歴情報を利用する場合、個々の AP によって AP から履歴情報を取得する方法は異なる。このことから、使用する AP 全ての履歴情報を収集するためには、実装において大きな工数を必要とする。

個々の文書ファイルの持つ情報（最終アクセス時間）を利用する場合、AP によらず同一の手法で AP 情報を取得できる。ここで、文書ファイルの最終アクセス履歴は利用者の操作によって変化するため、仕事開始時と仕事終了時における「各文書ファイルの最終アクセス時間」を比較することで、個々の AP の持つ履歴情報と同等の情報を得られる。しかし、この方法は仕事の開始時と終了時に 1 回ずつ、計算機内部に存在するデータ全てを走査しなければならず、処理負荷はとても大きい。

OS の提供する情報（最近使った文書ファイルの最終アクセス時間）を利用する場合も、AP によらず同一の手法で AP 情報を取得できる。また、OS の提供する情報は、利用者の参照したことのデータの一覧であるため、余分なデータ（例えば、OS のシステムファイルなど、利用者が参照しないと思われるデータ）の情報を取得しない。ここで、先ほども述べたように、仕事の開始時と終了時に「各文書ファイルの最終アクセス時間」を比較することにより、個々の AP の持つ履歴情報と同等の履歴情報を取得できる。しかし、OS にデータ参照を通知しない AP も存在し、このような AP を利用してデータを参照した場合、データを履歴情報として保存できない問題がある。

これらの比較結果を表 6.2 に示し、以下で詳細を述べる。

個々の AP の持つ履歴情報の利用による計算機内部の履歴情報取得は、実装にかかる工数がとても大きいことから、実現は困難である。また、個々の文書ファイルの持つ情報の利用による計算機内部の履歴情報取得は、一回の仕事状態保存にかかるオーバーヘッドがとても大きいことから、現実的ではない。

一方、OS の提供する情報を利用する場合、実現の工数は少なく、また、一回の仕事状態保存にかかるオーバーヘッドもそれほど大きくない。このため、デスクトップブックマークでは OS の提供する情報から、計算機内部にあるデータの履歴情報を収集する。

### 6.3.2 デスクトップブックマークへの要求

デスクトップブックマークを単体で利用する場合，以下のようなケースで問題となる．

(ケース 1) デスクトップブックマークを用いて過去の仕事で用いたデータを再参照しようとしたが，仕事状態の保存後データを書き換えていたため，データから欲しい情報を取得できなかった

(ケース 2) デスクトップブックマークを用いて過去の仕事で用いたデータを再参照しようとしたが，仕事状態の保存後データを移動させたため再参照できなかった

ここで，更新履歴管理システムとデスクトップブックマークを連携させることで，上記のケース，つまり，仕事状態保存時から現在までの「コンテンツの更新」と「データのパス移動」に対処できる．しかし，更新履歴管理システムとデスクトップブックマークを手動で連携させる場合，以下の手順を踏む必要がある．

(手順 1) デスクトップブックマークを用いて過去の仕事で用いたデータのパスを取得する

(手順 2) 更新履歴管理システムを利用して特定の時刻のデータの状態に復元する

上の手順 2 において，手順 1 で入手したデータのパスをもとに手動でコンテンツを復元する必要があることから，非常に手間がかかる．ここで，更新履歴管理システムとデスクトップブックマークを統一的に扱えば，これらの手間を低減できる．具体的には，更新履歴管理システムとデスクトップブックマークを連携させることで，以下の機能を実現したい．

(要求 1) コンテンツを保存：ケース 1 に対応

(要求 2) データのパス移動に対処：ケース 2 に対応

### 6.3.3 対処

上記の要求に対処するため，デスクトップブックマークの機能拡張として，更新履歴管理システムとの連携機構を実装する．具体的には，デスクトップブックマークの仕事状態からコンテンツを復元したいデータを選択すると，デスクトップブックマークはデータ管理システムを利用してコンテンツを復元する．また，データ管理システムはファイルシステム中における文書ファイルの移動をデスクトップブックマークに伝え，デスクトップブックマークは伝えられた情報を基に保存した仕事状態を更新する．これにより，デスクトップブックマークにコンテンツを復元する機能，およびデータのパス移動に対処する機能を持たせる．

表 6.3 更新履歴管理システムの比較

	Time Machine	Subversion
OS 状態の復元なし		
全てのコンテンツを保存できる		
OS 依存少ない	x	

### 6.3.4 更新履歴管理システムへの要求

デスクトップブックマークと更新履歴管理システムとの連携機構の目的から，デスクトップブックマークと連携させる更新履歴管理システムは以下の要求を満たす必要がある．

(要求 1) 仕事状態に直接関係のないコンテンツは元に戻さない

仕事状態に直接関係のないコンテンツを元に戻すと，問題を引き起こす場合がある．例えば，カーネル内部のライブラリデータなど，OS に関わるファイルのコンテンツが復元されると問題となる．このため，仕事状態に直接関係のないコンテンツはそのままに，仕事に利用したコンテンツのみ復元できることが望ましい．

(要求 2) 利用したコンテンツを全て保存できる

仕事で利用したコンテンツを全て保存できなければ，コンテンツを復元できない可能性がある．このため，仕事で利用したコンテンツを全て保存できることが望ましい．

(要求 3) 動作する OS への依存が少ない

デスクトップブックマークは AP として実現されているため，動作する OS への依存度が低い．デスクトップブックマークと連携する更新履歴管理システムが OS に強く依存する場合，システム全体も OS に強く依存することとなる．システム全体が OS に強く依存すると，連携機構を利用できるプラットフォームを限定してしまう問題が起こる．このため，連携する更新履歴管理システムの動作する OS への依存は低いことが望ましい．

### 6.3.5 利用する更新履歴管理システム

更新履歴管理システム「Time Machine [9]」「Subversion [10]」を，6.3.4 項で述べた要求を軸に比較する．これらの比較結果を表 6.3 に示し，以下で詳細を述べる．

TimeMachine は、システムデータも含めた全データを保存する。このため、ある時刻の計算機上のファイルの状態を全て復元しようとする、OS 等の状態まで元に戻してしまう可能性がある。また、TimeMachine は 1 ヶ月以上前のデータはディスク容量がなくなり次第、順次消去することから、過去の全てのコンテンツを復元できるとは限らない。さらに、Time Machine は、Mac OS 特有の機能であり、デスクトップブックマークと連携させる更新履歴管理システムをこの機能にした場合、連携機構が動作する OS を限定してしまう問題がある。

Subversion は、通常、オープンソース開発のプロジェクトのように、複数の計算機から 1 つのソースコードを改良する場合に用いられる。このため、一般の利用者計算機上のデータ管理には向いていない問題がある。そこで、デスクトップブックマークの仕事状態保存時に、リポジトリに更新内容を自動でコミットすることで、この問題に対処できると考えた。さらに、Subversion のリポジトリを個々の利用者が占有する形で利用することで、利用者計算機上の文書ファイル管理を行えと考えた。具体的には、リポジトリを利用者のアクセスできる任意の場所に置き、このリポジトリをある利用者が占有して利用する。このリポジトリを計算機上にチェックアウトし、チェックアウトしたディレクトリ以下（以下、ワーキングコピー）で利用者は作業を行うものとする。こうすることで、利用者計算機上の重要データのバックアップをとれる。このとき、リポジトリは蓄えたコンテンツを消去しないことから、過去の全てのコンテンツを復元できる。このリポジトリに OS に関する情報を除外して保存することで、OS の状態を元に戻すことなくコンテンツを復元できる。しかし、OS に関する情報を除外して保存すると、全てのデータを元に戻せるとは限らない。また、Subversion はオープンソースであり、主要な OS（Windows, Mac, Linux）上でなら問題なく動作する。

この Subversion とデスクトップブックマークを連携させることで、「何の仕事で利用したか」という曖昧な情報から、仕事を行っていた時刻やリビジョン番号を検索し、容易に重要データを復元可能になる。そこで、デスクトップブックマークと連携させる更新履歴管理システムを Subversion とする。

## 6.4 保存するデータの選定規則

デスクトップブックマークの実現にあたり、統一的履歴情報として保存するデータの選定規則を設ける。デスクトップブックマークは統一的履歴情報として、以下のいずれかの規則を満たしたデータのパスとイメージ図を保存する。

(規則 1) 利用者が参照した計算機内部にある文書ファイル

(規則 2) 利用者が Web ブラウザで一定時間以上参照した Web ページ

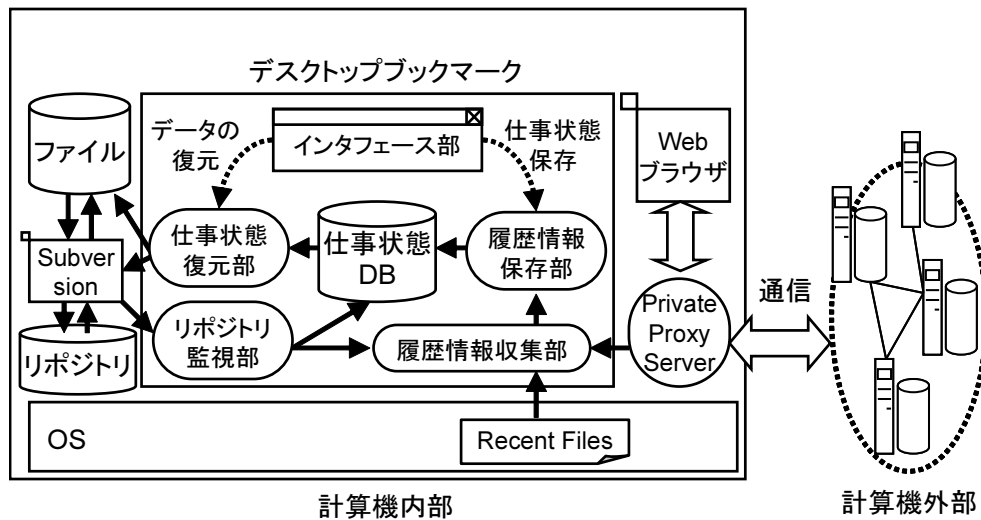


図 6.2 システム構成

計算機内部の履歴情報に関しては取得した情報を全て保存する．しかし，計算機外部の履歴情報に関しては，取得した履歴情報をフィルタリングして保存する．

計算機外部の履歴情報をフィルタリングする理由は，Web ブラウジングの際には，目的の Web ページに到達するまでに，目的の Web ページ以外の Web ページ（例えば，情報検索サイト）を経由することが多く，これらの経由ページの履歴情報は利用者にとって価値の低いページと考えられるためである．

ここで，利用者にとって重要な Web ページを閲覧する場合，利用者は一定時間以上その Web ページを見続けると考えられる．しかし，経由ページの場合，利用者はすぐに他の Web ページに遷移すると考えられる．このため，Web ブラウザで一定時間以上参照した Web ページのみを取得することにより，不要なパスの取得を最小限に抑える．

## 6.5 システム構成

デスクトップブックマークのシステム構成を図 6.2 に示し，以下で各システム構成要素について説明する．

- (1) インタフェース部：インタフェース部は，利用者がデスクトップブックマークを利用する際に用いる Web インタフェースである．インタフェース部は統一的履歴情報を列挙し，利用者に提示する．また，統一的履歴情報を利用者の分かりやすい形式に変換し，

利用者に提示する．そして，利用者はインタフェース部から仕事の開始と終了をデスクトップブックマークに通知する．

- (2) 仕事状態復元部：仕事状態復元部は，インタフェース部から利用者の再参照したいデータのパスを受け取り，そのデータを再参照する処理部である．仕事中に参照していたデータを一括して再参照する場合は，インタフェース部から仕事状態を受け取り，利用者が過去に参照していたデータを一括して再参照する．
- (3) 仕事状態 DB：仕事状態 DB は，仕事状態を永続的に保存しておくファイルである．
- (4) 履歴情報保存部：履歴情報保存部は，インタフェース部から仕事状態保存要求を受け取ると，履歴情報収集部から統一的履歴情報を受け取り，仕事情報と関連付けて DB に保存する処理部である．仕事情報を収集するために，履歴情報保存部は保存時の仕事名を入力として要求する．
- (5) 履歴情報収集部：履歴情報収集部は，統一的履歴情報を収集する処理部である．計算機外部の履歴情報を収集するために，履歴情報収集部は Web ブラウザと外部計算機との通信を仲介するプロキシサーバ (Private Proxy Server) からログを取得する．また，計算機内部の履歴情報を収集するために，OS から仕事開始時と終了時に AP 情報を取得する．
- (6) 仕事状態復元部：仕事状態復元部は，インタフェース部から利用者の再参照したいデータのパスを受け取り，そのデータを再参照する処理部である．仕事中に参照していたデータを一括して再参照する場合は，インタフェース部から仕事状態を受け取り，利用者が過去に参照していたデータを一括して再参照する．また，コンテンツの復元要求を受け取ると，インタフェース部から仕事時のリビジョン番号と復元したいデータのパスを受け取り，それを基にコンテンツを過去の状態に復元する．
- (7) リポジトリ監視部：リポジトリ監視部は，Subversion を利用してリポジトリを監視する処理部である．仕事の終了時，リポジトリ監視部はリポジトリに文書ファイルの更新内容をコミットし，履歴情報保存部へコミット後のリビジョン番号を伝える．また，計算機内部のデータの移動情報を取得し，仕事状態 DB を書き換える．

## 6.6 動作概要

デスクトップブックマークの動作概要を以下に示す (A-1) から (A-4) に仕事状態の保存時の処理の流れを示し (B-1) から (B-3) に仕事状態の復元時の処理の流れを示す。また、(C-1) から (C-4) に紛失したデータの復元時の処理の流れを示す。

(A-1) 利用者は計算機上で新しく仕事を開始する前に、Web CGI である「インタフェース部」から「仕事状態保存部」に仕事開始を通知する。

(A-2) 仕事を中断する際、利用者は現在の仕事状態を保存するために、「仕事状態列挙部」を通じて「仕事情報」(仕事名とキーワード)を入力し、「履歴情報保存部」に仕事状態の保存開始を指示する。

(A-3) 「履歴情報収集部」は「リポジトリ監視部」から「統一的履歴情報」(リビジョン番号)を取得する。このとき「リポジトリ監視部」はリポジトリに文書ファイルの更新内容をコミットする。また、プロキシサーバから「統一的履歴情報」(計算機外部のデータ)を取得する。さらに、仕事開始時と終了時の「最近使った文書ファイル情報の差分」から「統一的履歴情報」(計算機内部のデータ)を取得する。

(A-4) 「履歴情報保存部」は (A-2) で得た「仕事情報」と (A-3) で得た「統一的履歴情報」を関係付けて、DB に保存する。

(B-1) 「インタフェース部」は「仕事情報」の一覧を利用者に提示する。利用者は「仕事状態」を復元するために、提示された「仕事情報」の一覧から復元する「仕事状態」を指定する。

(B-2) 「インタフェース部」は指定された「仕事状態」の詳細を利用者に提示する。

(B-3) 利用者が提示されたデータから再参照したいデータを指定すると、「仕事状態復元部」は仕事状態として保存した「データのパス」にあるデータを再参照する。

(B-4) 提示されたデータを一括して再参照したい場合、利用者は「仕事状態の復元」を指定する。ここで、「仕事状態復元部」に仕事状態の復元開始を指示する。

(B-5) 「仕事状態復元部」は「仕事状態表示部」から受け取った「統一的履歴情報」を基に、過去に参照していたデータを一括して再参照する。

- (C-1) 「インタフェース部」は「仕事情報」の一覧を利用者に提示する．利用者はコンテンツを復元するために，提示された「仕事情報」の一覧から，コンテンツを復元したいデータを利用していた仕事に関係する「仕事情報」を指定する．
- (C-2) 「インタフェース部」は「仕事情報」と関連付けられた「統一的履歴情報」を利用者に提示する．利用者は提示された統一的履歴情報から，コンテンツを復元したいデータを選択する．
- (C-3) 「仕事状態復元部」は「インタフェース部」から受け取ったデータのパスとリビジョン番号を基にコンテンツを復元する．



## 第 7 章

## 評価

### 7.1 比較対象研究

#### 7.1.1 Google Desktop

Google Desktop [7] は、統一的履歴情報を Google の検索技術と組み合わせることにより、統一的履歴情報からのデータ検索を容易に実現する。例えば、「評価 論文」などのキーワードからデータを検索できる。Google Desktop は以下の特徴を持つ。

- (1) 利用者に提示する情報は以下の通り
  - (A) データにアクセスした時間
  - (B) データのパス
  - (C) データのキャッシュ
  - (D) データのイメージ図
- (2) 自動で計算機上の履歴情報を保存する
- (3) 取得した全ての履歴情報を保存する
- (4) 履歴情報の取得方法は以下の通り
  - (A) 計算機外部の履歴情報

計算機外部の履歴情報は対応する Web ブラウザから個別に入手している。このため、代表的な Web ブラウザを利用しなければならない制約を持つ。

## (B) 計算機内部の履歴情報

計算機内部の履歴情報はファイルシステムから入手している．具体的には，計算機のファイルシステムの最終更新日時を定期的にチェックし，更新があれば，データ内容を保存し，牽引を更新する．

## (5) AP にて機能を提供している

Google Desktop は，計算機上で利用者の行った行動を監視し，計算機外部のデータへのアクセス履歴と計算機内部のデータの更新履歴を自動で DB に保存する．このため，利用者が意識することなく履歴情報を収集できる利点を持つ．また，AP にて機能を提供していることから，機能の導入に掛かるコストは低い．

しかし，計算機内部のデータに関しては，参照履歴ではなく更新履歴を保存するため，重要な履歴情報を収集できない可能性がある．また，Google Desktop は対応していない Web ブラウザの利用者の Web の操作履歴を保存できないことから，重要な履歴情報を収集できない可能性がある．さらに，取得した履歴情報を全て保存することから，利用者にとって重要でないデータを大量に保存する可能性がある．

Google Desktop はデータのパスやイメージ図の他に，データのキャッシュを保存する．デスクトップブックマークはデータのキャッシュを保存しないものの，4.2 節で議論したように，これは仕事状態の復元に影響を与えないと考えられる．

### 7.1.2 俺デスク

俺デスク [8] は，計算機内部のデータをトリガとして，計算機外部のデータの検索を可能にする．これにより，利用者は「PDF ファイル A を参照時に参照していた Web ページを検索」といった要求が可能になる．俺デスクは以下の特徴を持つ．

## (1) 利用者に提示する情報は以下の通り

- (A) データにアクセスした時間
- (B) データのパス
- (D) データのイメージ図

## (2) 自動で計算機上の履歴情報を保存する

## (3) 取得した全ての履歴情報を保存する

#### (4) 履歴情報の取得方法は以下の通り

##### (A) 計算機外部の履歴情報

計算機外部の履歴情報は Web ブラウザのアドオンから入手している．このため，代表的な Web ブラウザを利用しなければならない制約を持つ．

##### (B) 計算機内部の履歴情報

俺デスクは計算機内部のデータと Web 参照履歴との関連付けを行うものの，計算機内部の履歴情報を保存しない．

#### (5) AP にて機能を提供している

俺デスクは，計算機上で利用者の行った行動を監視し，計算機外部のデータへのアクセス履歴を自動で DB に保存する．このため，利用者が意識することなくデータを収集できる利点を持つ．また，AP にて機能を提供していることから，機能の導入に掛かるコストは低い．

しかし，計算機内部のデータの参照履歴を保存しないため，重要な履歴情報を収集できない可能性がある．また，俺デスクは対応していない Web ブラウザの利用者の Web の操作履歴を保存できないことから，重要な履歴情報を収集できない可能性がある．さらに，取得した履歴情報を全て保存することから，利用者にとって重要でないデータを大量に保存する可能性がある．

## 7.2 評価概要

### 7.2.1 データの再現率と適合率

提案機能の評価として，デスクトップブックマークは 3.3 節に示した特徴を満たしているか評価する．ここで，(特徴 1)，(特徴 3)，(特徴 4)，および (特徴 5) については，提案機能の実現方針，およびデスクトップブックマークのシステム設計より，特徴を満たしていることは自明である．このため，デスクトップブックマークを利用することで，仕事状態の復元にかかる手間は少なくなるか評価する．

ここで，私は統一的履歴情報として各ツールによって取得されたパスに着目した．各ツールの取得した統一的履歴情報に不要なパスが多く含まれる場合，仕事状態を復元する際に仕事状態からデータを想起できなくなる．一方，各ツールの取得した統一的履歴情報にパス取得漏れが多い場合，仕事状態を復元する際に過去に参照したデータを検索する手間は増大す

## Desktop Bookmark

DPSWS用スライド作成 (2007年10月18日22時07分)

仕事名

スケジュール

デスクトップ復元

開いたファイル

パス

[C:\ogasawara\doc\thesis\DPSWS\(カメラレディ\)\dpsws2007-1018.ppt](#)  
[C:\ogasawara\doc\thesis\DPSWS\(カメラレディ\)\dpsws15 final\dpsws15.pdf](#)  
[C:\ogasawara\doc\thesis\DPSWS15\image.htm](#)

閲覧したWebページ

サムネイル



図 7.1 仕事状態の提示例

る．このため、「パス取得漏れの少なさ」と「不要なパス取得の少なさ」は仕事状態の復元に大きく影響する．

そこで、「パス取得漏れの少なさ」と「不要なパス取得の少なさ」の評価尺度として、データの再現率と適合率を導入した．データの再現率は、利用者が有用と判断した全パスのうち、ツールの取得したパスの割合である．データの適合率は、ツールの取得した全パスのうち、利用者が有用と判断したパスの割合である．データの再現率と適合率が大きいほど、仕事状態の復元にかかる手間は少ないといえる．

評価ではまず、「デスクトップブックマークの選定規則で利用する閾値」と「データの再現率と適合率」との関連性を評価する．次に、デスクトップブックマークと既存手法の「データの再現率と適合率」を比較評価する．

### 7.2.2 環境

デスクトップブックマークを Microsoft Windows XP 上に Ruby on Rails を用いて実装した．1つの仕事状態の提示画面を図 7.1 に示し、以下に説明する．

デスクトップブックマークは、仕事情報として仕事名、スケジュール(仕事を行った時刻)を保存する．また、デスクトップブックマークは1つの仕事情報と関連付けて、計算機内部と外部のデータの閲覧履歴(データのパス、およびイメージ図)を提示する．ここで、データ

のイメージ図は，80 %以上の利用者が Web ページの詳細を認知 [23] できる「250 × 200 ピクセル」とした．

評価では，仕事状態として保存したデータの閲覧履歴と「利用者に有益なデータのパス一覧」を比較することにより，デスクトップブックマークのデータの再現率と適合率を測定した．

### 7.2.3 手順

(手順 1) 被験者はデスクトップブックマークを起動した状態で 30 分間かけてレポートを作成する

レポートの題目は，被験者が興味を持っていて，かつ特別詳しい知識を持ち合わせていないような題目（「日本の電気メーカ」）とした．このような題目を与えることで，レポート作成は被験者の日常的な計算機上の作業と同等な作業となる．また，作業時間を 30 分と短くした理由は，被験者の負担を最小限に抑えるためである．

(手順 2) 被験者はレポート作成の際に有用だったデータのパスをレポートに添付する  
ここで，レポートに添付されたパスから，利用者に有益なパス一覧を作成する．

(手順 3-A) デスクトップブックマークの収集した統一的履歴情報におけるデータの再現率と適合率を測定する

「デスクトップブックマークの閾値」と「データの再現率と適合率」との関連性を測定することで，デスクトップブックマークの最適な閾値を求める．

(手順 3-B) 各ツールの収集した統一的履歴情報におけるデータの再現率と適合率を測定する  
デスクトップブックマークは関連ツールと比較して，どの程度「利用者に有益なデータのパス」を収集できたか調査する．

## 7.3 評価結果

### 7.3.1 評価 1：選定規則の閾値の妥当性

デスクトップブックマークは計算機外部の履歴情報に関して，利用者が一定時間参照したデータのみを保存する．そこで，閾値とデータの再現率と適合率の関連性について，7.2.3 節

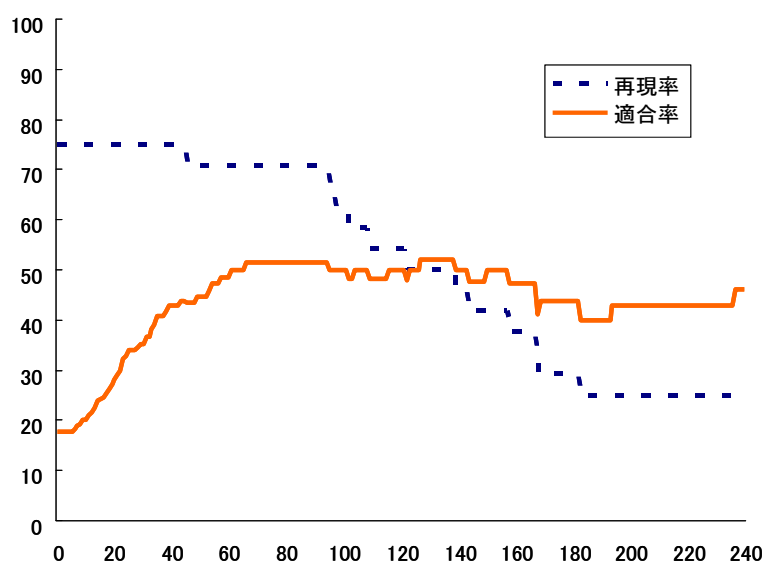


図 7.2 評価 1 の結果

で述べた評価方法を用いて評価を行った。本評価の被験者は私の所属する研究室の学生 (5 人) である。なお、ここでは計算機外部の履歴情報のみに着目して評価を行う。評価の結果を図 7.2 に示し、以下で考察を述べる。

閾値を 0 秒にすると、全ての履歴情報を保存することから、データの再現率は 100 % になるはずである。しかし、閾値を 0 秒に設定した場合もデータの再現率は 100 % にならなかった。この原因は、6.2.2 項で述べた Private Proxy Server ログフィルタの抽出精度が 100 % ではないためである。

閾値が 45 秒未満の場合、データの再現率に変化はなかった。また、データの適合率は閾値に応じて大きくなった。つまり、利用者に有益な全データは 45 秒以上参照されたため、利用者に有益なデータは全くフィルタリングされなかった。このため、データの再現率は低下しなかった。これに加えて、27 秒未満参照された経由ページは存在したことから、経由ページをフィルタリングする割合は閾値によって増加した。このため、データの適合率は閾値に応じて大きくなった。

閾値が 45 秒以上 95 秒未満の場合、データの再現率は閾値 45 秒を境に一旦低下するものの、それ以降で変化はなかった。また、データの適合率は、2 % 未満の下げ幅で低下する時はあるものの、閾値に応じて大きくなる傾向にあった。つまり、利用者に有益なデータの一部 (4 %) は 45 秒参照されたため、閾値 45 秒を境に利用者に有益なページの一部はフィルタリングされた。このため、データの再現率は一旦低下するものの、それ以降は低下しなかった。しかし、利用者に有益なデータの大部分は 95 秒以上参照されることから、利用者にとって

有益なページをフィルタリングする割合よりも、経由ページをフィルタリングする割合が大きくなる。この結果、データの適合率は、2 %未満の下げ幅で低下する時はあるものの、閾値に応じて大きくなる傾向となった。

閾値が 95 秒以上の場合、データの再現率は大きく低下する傾向にあった。また、データの適合率はゆるやかに低下する傾向にあった。つまり、利用者に有益なデータの多く (50 %) は 95 秒以上参照されたため、利用者に有益なページの多くがフィルタリングされた。このため、データの再現率は大きく低下した。また、経由ページもフィルタリングされるものの、これ以上に利用者に有益なデータをフィルタリングした。このため、データの適合率はゆるやかに低下した。

つまり、閾値には正解データを不正解とみなさない限界値 (今回の場合は 45 秒) が存在し、限界値を超える値に閾値を設定すると、データの再現率は大きく低下する。閾値が限界値以下の場合、経由ページを削除できる割合は増加し、さらに、閾値を上げて利用者にとって有益なデータを削除しないことから、データの適合率は閾値に応じて増加する。しかし、閾値が限界値を超えると、経由ページを削除できる割合は増加するものの、利用者にとって有益なデータまでも経由ページとして削除してしまうことから、データの適合率は閾値に応じて増加するとは限らない。

このことから、データの再現率をあまり低下させることなく、データの適合率を一番高めることができる最良の閾値は、多くの正解データを不正解とみなすようになる限界値 (今回の場合は 44 秒) であることがわかった。

しかし、デスクトップブックマークの最良の閾値 (44 秒) が一般的に最適な閾値であるかどうかは分からない。なぜなら、この評価は仕事の種類、仕事にかけられる時間、利用者特有の癖などで変化することから、今回の評価の最良の閾値 (44 秒) が一般的に最適な閾値であるかどうかは、より詳細な検討を必要とするためである。そこで、被験者のうち 4 人の評価結果を図 7.3 に示し、以下で考察を述べる。

同じ作業内容で、仕事にかけられる時間にもかかわらず、被験者によってデータの再現率と適合率のグラフは大きく異なった。これは、デスクトップブックマークの閾値は、利用者特有の癖によって変化することを示している。

このため、最適なデスクトップブックマークの閾値も、被験者によって異なることがわかった。具体的には、A の最適な閾値は 44 秒であるものの、B の最適な閾値は 94 秒、C の最適な閾値は 143 秒、D の最適な閾値は 108 秒である。

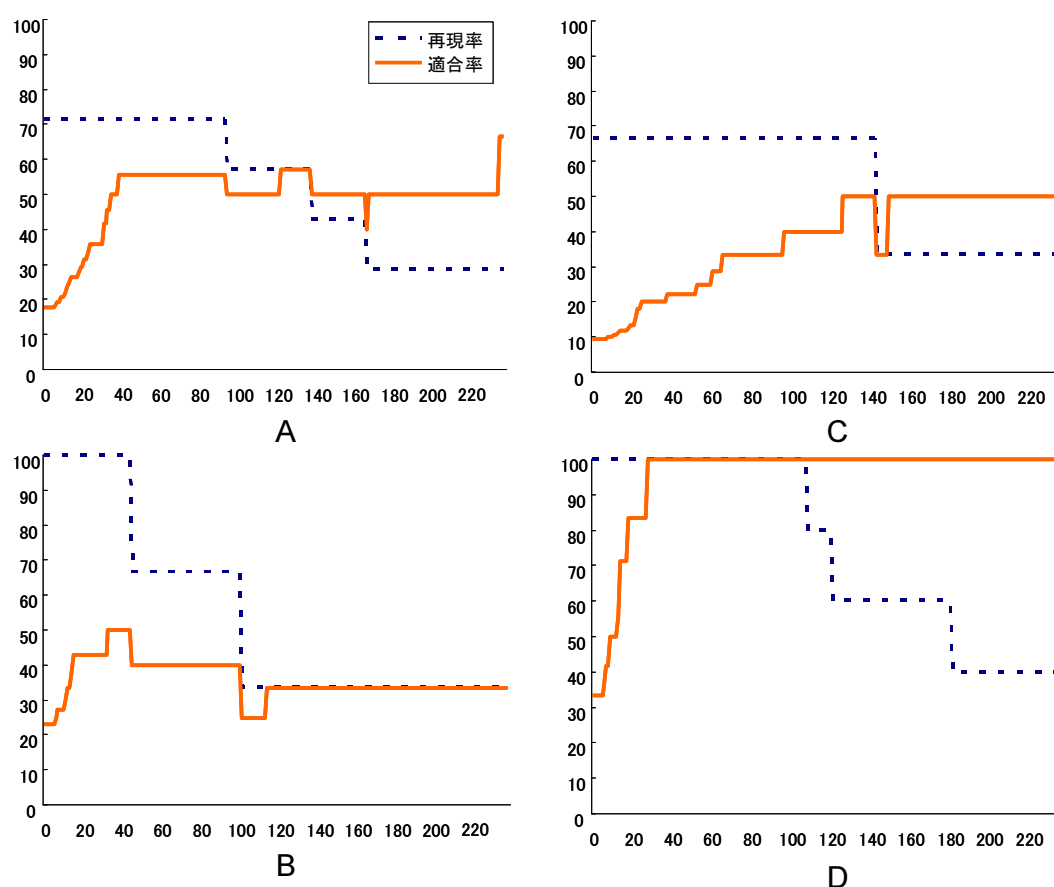


図 7.3 評価 1 の結果 (個人)

### 7.3.2 評価 2：仕事状態復元の容易さ

デスクトップブックマークと同じく AP にて機能を提供している Google Desktop と俺デスクを評価対象とし，7.2.3 節で述べた評価方法を用いて評価を行った．本評価の期待される効果は，デスクトップブックマークのデータの再現率と適合率が関連ツールよりも大きくなることである．評価の結果を表 7.1 と表 7.2 に示し，以下で考察を述べる．なお，評価において，デスクトップブックマークの選定規則の閾値は，各個人の最適値でなく，一般的な最適値である 44 秒としている．こうすることで，利用者特有の閾値ではなく，この一般的に最適な閾値を利用しても，デスクトップブックマークは他の関連ツールよりも有用であることを示す．

データの再現率に関しては，デスクトップブックマークも既存手法も 100 % にならず，ほぼ同じ値となった．この理由として，以下の 2 つの要因によるデータの再現率の低下は同程度であるためと考えられる．



表 7.1 デスクトップブックマークと関連ツールとの比較結果

	データ種類	個数 (計算機内部/外部)
正解データ DTB	正解	34(10/24)
	不正解	23(5/18)
俺デスク	正解	24(0/24)
	不正解	101(0/101)
Google Desktop	正解	28(4/24)
	不正解	314(212/102)

DTB = デスクトップブックマーク

表 7.2 各関連ツールのデータの再現率と適合率

	DTB	俺デスク	Google Desktop
データの再現率 (%)	68	71	82
データの適合率 (%)	49	19	8

DTB = デスクトップブックマーク

(プラス要因) デスクトップブックマークは計算機内部のデータの参照履歴を取得する．このため，利用者にとって有益な計算機内部のデータを履歴情報として全て保存できる．一方，Google Desktop は計算機内部のデータの更新履歴を取得するものの，参照履歴を取得できない．また，俺デスクは計算機内部の履歴情報を取得できない．つまり，関連ツールは利用者にとって有益なデータのうち，いくつかを参照履歴として保存できない．この結果，関連ツールのデータの再現率は若干低くなる．

(マイナス要因) 関連ツールは計算機外部の履歴情報をフィルタリングすることなく全て保存する．このため，利用者にとって有益な計算機外部のデータを履歴情報として全て保存できる．一方，デスクトップブックマークは Private Proxy Server のログから計算機外部の履歴情報を抽出する．この抽出精度は 100 % でないことから，デスクトップブックマークは利用者にとって有益なデータのうち，いくつかを履歴情報として取得できない．この結果，デスクトップブックマークのデータの再現率は若干低くなる．

一方，データの適合率に関しては，関連ツールはそれぞれ 8 %，19 % だったのに対して，デスクトップブックマークは 49 % と大きく改善された．これは，計算機外部の履歴情報の

表 7.3 各関連ツールのデータの再現率と適合率 (個人)

	DTB	俺デスク	Google Desktop
データの再現率 (%)	100	83	100
データの適合率 (%)	100	28	11
データの再現率 (%)	60	60	60
データの適合率 (%)	30	18	10
データの再現率 (%)	67	78	89
データの適合率 (%)	60	18	7
データの再現率 (%)	44	67	78
データの適合率 (%)	29	19	8
データの再現率 (%)	80	60	80
データの適合率 (%)	57	16	7

DTB = デスクトップブックマーク

取得精度が大きく関わっている．デスクトップブックマークは不要なデータを 24 個しか保存していないのに対して，既存手法はそれぞれ 101 個，314 個取得している．このような結果になった理由は，デスクトップブックマークは計算機外部の履歴情報をフィルタリングするのにに対して，Google Desktop と俺デスクは計算機外部の履歴情報を全て保存するためだと考えられる．

計算機外部の履歴情報の中には経由ページを多く含んでおり，これらのページは「利用者に有益なデータのパス」に含まれなかった．デスクトップブックマークはこの経由ページを除外するように計算機外部の履歴情報をフィルタリングしている．一方，関連ツールは計算機外部の履歴情報を全て保存する．このため，関連ツールは「利用者に有益なデータのパス」に含まれない経由ページを履歴情報として大量に保存する．この結果，関連ツールと比較してデスクトップブックマークのデータの適合率は高くなったと考えられる．

また，各被験者におけるデータの再現率と適合率を表 7.3 に示し，以下で考察を述べる．

デスクトップブックマークのデータの適合率は，全ての被験者において既存手法よりも高くなった．一方，データの再現率に関しては，既存手法と同程度であるといえない場合もあり，被験者によってばらつきがあった．これは，利用者の Web ブラウジングの方法に関係している．

今回の評価では，利用者はタブ機能付の Web ブラウザを用いて Web ブラウジングを行った．ここで，ある利用者は，検索ページからレポート作成に有用そうなデータをあらかじめ

タブに開いておき，そこから順番に開いたタブを閲覧する方法をとっていた．デスクトップブックマークはプロキシサーバが Web ページを GET する履歴を見ているため，タブ機能付の Web ブラウザで先ほど述べたような操作を行うと，デスクトップブックマークは「Web ページを閲覧した時間」を上手く取得できない．このため，データの抽出精度は低下し，データの再現率が低下したと考えられる．しかし，このような問題はあまり発生していない．

以上から，デスクトップブックマークは既存手法と比較して，データの再現率は同程度であるものの，データの適合率は大きく改善されることを示せた．適合率は「不要なパス取得の少なさ」であることから，デスクトップデスクトップブックマークは既存手法と比較して，大量の不要なパスを提示することなく，利用者に有益なパスを的確に提示できることを示している．このため，デスクトップブックマークは既存手法と比較して，仕事状態の復元に有益であるといえる．

## 第 8 章

### おわりに

計算機上の仕事状態の保存と復元機能を提案した．まず，統一的履歴情報を提供する既存手法について，その特徴と問題点を述べ，問題点への対処法を示した．次に，提案機能の特徴を示し，その実現方式について述べた．提案機能は仕事情報と統一的履歴情報を関連付けて保存することで，統一的履歴情報を仕事単位で集約する．

さらに，提案機能の一実装として「デスクトップブックマーク」を実装した．関連ツールとの比較評価により，デスクトップブックマークは保存した仕事状態から仕事の途中状態を容易に復元できることを示した．評価では，デスクトップブックマーク，および関連ツールが統一的履歴情報として取得したパスに着目し，評価尺度としてデータの再現率と適合率を導入した．そして，デスクトップブックマークは既存手法と比較して，データの再現率は同程度であるものの，データの適合率は大きく改善されることを示した．

残された課題として，仕事の切り替えへの対応手法の検討がある．

## 謝辞

本研究を進めるにあたり，懇切丁寧なご指導をして頂きました乃村能成准教授に心より感謝の意を表します．また，研究活動において，数々のご指導やご助言を与えて頂いた谷口秀夫教授に心から感謝申し上げます．さらに，日ごろから絶えずご指導やご助言を頂きました田端利宏准教授に深く感謝を申し上げます．

また，日頃の研究活動において，お世話になりました谷口研究室，田端研究室，および乃村研究室の皆様には感謝いたします．

最後に，本研究を行うにあたり，経済的，精神的な支えとなった家族に感謝いたします．

## 参考文献

- [1] Jim Gemmell, Gordon Bell, Roger Lueder, “MyLifeBits: a personal database for everything,” *Commun. ACM*, Vol. 49, No. 1, pp. 88–95 (2006).
- [2] 村上晴美, 平田高志, “記憶を中心とする人生の記録: ユーザの知識空間の作成による Web ブラウジング履歴の想起支援,” 情報処理学会人文科学とコンピュータ研究会報告, Vol. 2004, No. 7, pp. 19–24 (2004).
- [3] M. Lamming, M. Flynn, “Forget-me-not: intimate computing in support of human memory,” *Proceedings FRIEND21 Symposium on Next Generation Human Interfaces* (1994).
- [4] Margery Eldridge and Abigail Sellen and Debra Bekerian “Memory Problems at Work: Their Range, Frequency and Severity,” *Technical report, EPC-1992-129* (1992).
- [5] Laadan, O. and Nieh, J. “Transparent Checkpoint-Restart of Multiple Processes on Commodity Operating Systems,” *2007 USENIX Annual Technical Conference*, pp. 323–336 (2007).
- [6] VMware Inc., “*VMware*,” <http://www.vmware.com/>.
- [7] Google Inc., “*Google Desktop*,” <http://desktop.google.com/ja/>.
- [8] 大澤 亮, 高汐一紀, 徳田英幸, “俺デスク: ユーザ操作履歴に基づく情報想起支援ツール,” 情報処理学会第 47 回プログラミング・シンポジウム (2006).
- [9] Apple Inc., “Mac OS X Leopard - Features - Time Machine,” <http://www.apple.com/macosx/leopard/features/timemachine.html>.
- [10] Tigers.org, “*Subversion*,” <http://subversion.tigris.org/>.

- 
- [11] Thomas W. Malone, “How do people organize their desks?: Implications for the design of office information systems,” *ACM Trans. Inf. Syst.*, Vol. 1, No. 1, pp. 99–112 (1983).
- [12] Deborah Barreau, Bonnie A. Nardi, “Finding and reminding: file organization from the desktop,” *SIGCHI Bull.*, Vol. 27, No. 3, pp. 39–43 (1995).
- [13] Microsoft Corporation., “Windows デスクトップサーチ,”  
<http://www.microsoft.com/japan/windows/desktopsearch/default.mspx>.
- [14] 野口悠紀雄, “「超」整理法,” 中公新書 (1993).
- [15] David Wolber, Michael Kepe, Igor Ranitovic, “Exposing document context in the personal web,” *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, ACM, pp. 151–158 (2002).
- [16] Koichi Hayashi, Takahiko Nomura, Tan Hazama, Makoto Takeoka, Sunao Hashimoto, Stephan Gumundson, “Temporally threaded workspace: a model for providing activity-based perspectives on document spaces,” *HYPERTEXT '98: Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, ACM, pp. 87–96 (1998).
- [17] Eric Freeman, David Gelernter, “Lifestreams: a storage model for personal data,” *SIGMOD Rec.*, Vol. 25, No. 1, pp. 80–86 (1996).
- [18] Jun Rekimoto, “Time-machine computing: a time-centric approach for the information environment,” *UIST '99: Proceedings of the 12th annual ACM symposium on User interface software and technology*, New York, NY, USA, ACM, pp. 45–54 (1999).
- [19] Sean Uberoi Kelly, John P. Davis, “Desktop History: Time-based Interaction Summaries to Restore Context and Improve Data Access,” *International Conference on Human-Computer Interaction*, pp. 204–211 (2003).
- [20] Anton N. Dragunov, Thomas G. Dietterich, Kevin Johnsrude, Matthew McLaughlin, Lida Li, Jonathan L. Herlocker “TaskTracer: a desktop environment to support multi-tasking knowledge workers,” *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, ACM, pp. 75–82 (2005).

- 
- [21] 栗原 聖治, “Private Server の通信履歴を利用した Web ページ閲覧情報の抽出法,” 岡山大学工学部情報工学科卒業論文, 2007.
- [22] 栗原 聖治, 乃村 能成, 谷口 秀夫, “プロキシサーバによる効率的な Web 閲覧履歴の取得方法の検討,” 電子情報通信学会 2009 年総合大会講演論文集,(掲載予定).
- [23] Shaun Kaasten, Saul Greenberg, Christopher Edwards, “How people recognize previously seen Web pages from titles, URLs and thumbnails,” *Proceedings of Human Computer Interaction 2002*, pp. 247–265 (2002).



## 発表論文

- [1] 小笠原良, 乃村能成, 谷口秀夫, “デスクトップブックマーク：計算機上の仕事状態の保存と復元機能の提案,” マルチメディア通信と分散処理ワークショップ論文集, pp. 177–182 (2007.10).
- [2] 小笠原良, 乃村能成, 谷口秀夫, “デスクトップブックマーク：計算機上の仕事状態の保存と復元機能の評価,” マルチメディア・分散・協調とモバイルシンポジウム論文集, pp. 1418–1423 (2008.07).
- [3] 小笠原 良, 乃村 能成, 谷口 秀夫, “更新履歴管理システムを用いたデスクトップブックマークの機能拡張,” マルチメディア通信と分散処理ワークショップ論文集, pp. 73–78 (2008.12).
- [4] 小笠原 良, 乃村 能成, 谷口 秀夫, “仕事の引き継ぎ支援に関する検討,” 電子情報通信学会 2009 年総合大会講演論文集,(掲載予定).