卒業論文 2004 年度 (平成 16 年度)

ユーザ動作を基にしたデータ間関連性とデータ着目度算出機構

指導教員

慶應義塾大学環境情報学部

徳田　英幸

村井　純

楠本　博之

中村　修

南　　政樹

慶應義塾大学 環境情報学部

大澤 亮

ryo@sfc.wide.ad.jp

卒業論文要旨　　2004年度(平成16年度)

ユーザ動作を基にしたデータ検索用メタ情報生成機構の構築

近年，コンピュータ技術の進歩に伴い，様々な分野で情報の電子化が行われてきた．情報とは文書や画像，映像，音楽のデータなどが該当する．それらのデータが膨大化，多様化するに従い，ユーザが目的のデータを見つけるのが困難になってきた．そのため，ユーザのデータ発見を支援するために，様々な検索システムが実用化されてきた．

通常，ユーザは既読データの再検索を迅速にするため，検索したデータの中で再閲覧の可能性のあるデータを保存する．しかし，保存したデータ数が膨大になると，それらのデータ中から見つけたいデータを発見するのが困難になる．膨大になった履歴データの検索を効率化する手法には，フィルタリングやカテゴリ分けがある．しかし，これらの手法はユーザに対して手動での操作を要求する．従って，参照量が多くなるとユーザに対する負担が大きい．また，ユーザは付加情報の付け忘れや分類間違いを起こした場合，目的のデータを見つけられない．

本研究の目的は，ユーザの履歴データ検索を効率化する事である．具体的にはユーザのデータへのインタラクションとアプリケーションイベントを基にした連想検索機能とデータレーティング機能を提案する．人間の記憶は連想からなり，履歴検索においてはユーザの過去体験を基にした上記の検索手法は有効である．本研究ではユーザのデータへのインタラクションとアプリケーションイベントを監視，保存しデータのレーティングとデータ間の関連性算出を自動で行うミドルウェア **DMemFinder(Deep Memory Finder)** を構築する．DMemFinder を用いることで検索アプリケーションは関連検索とフィルタリング機能をユーザに提供できる．

本論文では，第 2 章において，既存のデータ検索手法と本研究の関連性について詳しく述べる．第 3 章ではまず，本研究の目的と機能要件を述べ，ユーザのデータへのインタラクションとアプリケーションイベントを監視，保存しデータのレーティングとデータ間の関連性算出を自動で行う手法について述べる．第 4 章で DMemFinder の設計について，第 5 章で実装について述べる．そして，第 6 章で本機構を評価し，第 7 章で本論文をまとめる．

慶應義塾大学　環境情報学部
大澤 亮

Abstract of Bachelor's Thesis

User Behavior Based Meta-data Generator for Data Search

This thesis, first, clarifies the method of data search. It, then, presents DMemFinder(Deep Memory Finder) which is a dynamic meta-data generator for data search, and describes the design and implementation of DMemFinder. Finally, it shows evaluation of the system and concludes.

Ryo OHSAWA
Faculty of Environmental Information Keio University

目 次

図 目 次

表 目 次

第1章　序論

1.1 背景

近年，コンピュータ技術の進歩に伴い，様々な分野で情報の電子化が行われてきた．情報とは文書や画像，映像，音楽のデータなどが該当する．それらのデータが膨大化，多様化するに従い，ユーザが目的のデータを見つけるのが困難になってきた．そのため，ユーザのデータ発見を支援するために，様々な検索システムが実用化されてきた．例えば，ユーザが Web ページを検索する際，Google や Yahoo!といった検索エンジンが利用できる．また，専門的なデータが欲しい場合，ユーザは個々のデータベースにアクセスする事でデータを得られる．具体的には，総務省統計局の Web ページ [1] や Cite Seer[2] などが挙げられる．

通常，ユーザは既読データの再検索を迅速にするため，検索したデータの中で再閲覧の可能性のあるデータを保存する．保存の手法は，データへのポインタを保存する手法とデータそのものを保存する手法が挙げられる．前者の例はブラウザのブックマーク機能を利用したり，リンク集を作成する手法が該当する．後者の例はローカルの PC に保存したり，プリントアウトする手法が該当する．

しかし，保存したデータ数が膨大になると，それらのデータ中から見つけたいデータを発見するのが困難になる．ユーザが閲覧する Web ページの内，58 %が既読ページであるという研究報告 [3] がなされており，膨大になった履歴データの中からデータを効率よく検索する手法が今後一層重要になってくる．

1.2 問題意識

膨大になった履歴データの検索を効率化する手法には，フィルタリングやカテゴリ分けがある．フィルタリングとは，検索システムがユーザのデータに対するレーティングを保存し，レーティングを基に以降の検索結果を変更する手法である．例えば，ユーザが重要だと入力したデータを検索結果の上位にソートし，重要でないと入力したデータを検索結果から排除する手法が挙げられる．カテゴリ分けとはユーザがデータをカテゴリごとに分けて保存し，後の検索を容易にする手法である．具体例としては，ブックマークを分類保存しておく手法が挙げられる．

しかし，これらの手法はユーザに対して手動操作を要求する．従って，参照量が多くなるとユーザに対する負担が大きい．また，ユーザは付加情報の付け忘れや分類間違いを起こした場合，目的のデータを見つけられない．

1.3 目的

本研究の目的は，普段のユーザに手動操作を要求しない手法で，履歴データ検索を効率化することである．本研究ではユーザ動作をデータのメタ情報として保存し，そのメタ情報を基にデータ間の**関連度**とデータ**重要度**を算出し，データ検索を支援する手法を提案する．

本研究においてユーザ動作とは，ユーザがデータへアクセスしたイベントとアプリケーション固有のイベントを対象とする．前者の例としては，データの表示，クリップボード利用，文章マーキングなどのイベントが該当する．後者の例としては「メッセンジャで同僚 A に話かけられた」や「音楽 B を再生していた」などのイベントが該当する．

人間の記憶は連想からなり，履歴検索においてはユーザの過去動作を基に関連度や重要度を算出する手法は有効である [4].

本研究ではアプリケーションイベントを取得し，データ間関連度とデータの重要度算出を動的に行うミドルウェア **DMemFinder(Deep Memory Finder)** を構築する．DMemFinder を用いることでデータ検索アプリケーションは，データ間の関連性を用いた関連検索や重要度によるソート機能をもった検索をユーザに提供できる．

1.4 本論文の構成

本論文では，第 2 章において，既存のデータ検索手法と本研究の関連性について詳しく述べる．第 3 章ではまず，本研究の目的と機能要件を述べ，ユーザのデータアクセスイベントとアプリケーションイベントを取得しデータのデータ間の関連性算出とデータのレーティングを自動で行う手法について述べる．第 4 章で DMemFinder の設計について，第 5 章で実装について述べる．そして，第 6 章で本機構を評価し，第 7 章で本論文をまとめる．

第2章　データの検索

本章ではまず，一般的なデータ検索手法と本研究との関連性を述べる．次にデータ内容を基にしたデータ間の関連性算出手法について述べ，本研究の手法と比較する．最後に，本研究の対象である履歴データ検索の効率化を行う既存手法とその問題点について述べる．

2.1 データ検索場所

データの検索は対象とする検索場所によって検索手法が異なる．そのため本節ではデータ検索場所ごとにシステムがどのような手法を用いて，ユーザの検索を効率化しているかについて述べ，本研究との関連性を述べる．データ検索場所を Web，データベース，ローカル PC の 3 つに分類する．

2.1.1 Web

ユーザは検索エンジンを用いる事で，Web からデータを検索できる．検索エンジンは全世界の Web ページをインデックス化し，ユーザのリクエストに対して適切な Web ページを返す．現在，Google から検索できる全世界の Web ページ数は 80 億以上あり，ユーザは非常に多くの情報を Web から得られる．検索エンジンは Web ページのレーティングやカテゴリ分類，パーソナライズ検索をユーザに対して提供し，ユーザの検索を効率化している．

Web ページのレーティングとは，独自のアルゴリズムを用いて Web ページをレーティングし検索結果に反映させる手法である．例えば，Google はそれぞれの Web ページごとに Page Rank という値を設定し，検索結果を Page Rank 順にソートしている [5]．Page Rank は様々な要素を基に算出されている．例えば，Web ページへ向けられるリンク数が増えるほど Page Rank が上昇する．

Web ページのカテゴリ分類とは，記載されている内容ごとに Web ページを分類することである．カテゴリごとに検索機能を提供している検索エンジンをディレクトリ型検索エンジンと呼ぶ．ユーザはカテゴリに絞って検索を行う事で，検索結果の精度を上げられる．しかし，人が主に分類作業を行うため，情報量が少ないという問題点がある．

パーソナライズ検索とは，ユーザの好みに応じて検索結果を変えてユーザの検索を効率化する手法である．具体的には Google Personalized Search[6] が挙げられる．ユーザがあらかじめ趣味を登録すると，ユーザの趣味に応じた Web ページが上位にソートされる．例えば通常 Kahn と入力すると建築家の Louis I. Kahn 氏が先頭にくるが，趣味をサッカーにするとサッカードイツ代表の Oliver Kahn 選手の記事が先頭にくる．

本研究ではユーザのデータアクセスイベントとアプリケーションイベントを基に，データのレーティングと関連性算出を行っている．過去の動作記憶を基に検索したい場合は本研究の方が適切である．逆に一般的な重要度や分類を基に検索したい場合は既存手法が有効である．

2.1.2 データベース

ある特定分野のデータのみを検索したい場合は，データベースにアクセスする手法が有効である．例えば，論文を検索する場合は Google で Web 全体を検索するより，Cite Seer や Google Scholar[7] などを用いた方が効率よく検索できる．

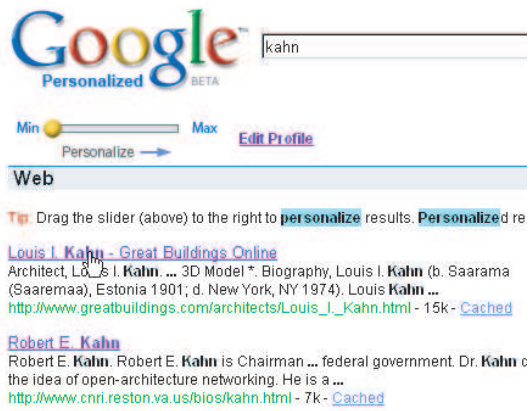

図 2.1: 通常の検索結果

図 2.2: パーソナライズされた検索結果

また，現在の検索エンジンでは，ユーザが入力した単語が含まれている Web ページを探すことしかできない．しかし，Web ページにメタ情報が付加されれば，検索システムはより多様なクエリをユーザから受け取れる．現在，Web 検索機能とメタ情報データベースを連携させるセマンティック Web の研究が進められている．セマンティック Web のメタデータは RDF(Resource Description Framework)[8] で記述される．例えば，各レストランのサイトに営業日，時間，場所といった共通のメタ情報を付加すれば，ユーザは「新宿駅」「金曜」「22 時以降も営業」というキーワードからレストランが探せる．

さらに近年，様々な分野のデータベースを統合し，ユーザに対して分野を横断した検索機能を提供し，ユーザの検索を効率化する手法の研究が進められている．データベースを横断的に扱う共通仕様は OWL(Web Ontology Language)[9] で記述される．OWL を用い，メタ情報間の関連を記述することで，ユーザは複数データベース間での検索が可能になる．例えば，会社の社員管理データベースとワークステーションのファイルシステムを統合することにより，人の名前からプロフィール，プロジェクト，ドキュメントなどの関連情報が検索できる．

本研究では，ユーザのデータへのインタラクションとアプリケーションイベントをメタ情報としてデータに付加し，検索に反映させている．これらのメタ情報と別のデータベースの情報を OWL を用いて仕様を定義すれば，他のメタ情報データベースと連携できる．

2.1.3 ローカル PC

通常 OS はローカル PC 内のデータを検索する技術を提供している．しかし，近年より高速に検索する技術が実用化されてきた．具体的には Google Desktop Search[10] や Copernic Desktop Search[11] などが挙げられる．

Google Desktop Search はユーザがアイドル状態の時，データの場所をインデックス化し，データへの高速検索機能を提供する．またファイルの変更，移動，削除などのイベントを監視し，インデックスを更新する．現在，Microsoft Word，Excel，Power Point，テキ

ストファイル，Outlook，Outlook Express の受信メールのデータを検索可能である．2004 年 12 月 Ask Jeeves Desktop Search[12] と MSN Desktop Search[13] のβ版がリリースされた．また，Yahoo!も X1 社 [14] の技術を使い近日中にリリースする予定である．各社の検索システムはまだβ版であり今後様々な改良が行われると思われる．

本機構においても検索の高速化を行うため，データのデータの変更，移動，削除をインデックス化する．また，そうすることにより，データの保存場所変更にも対応可能になる．

2.2 データ間の関連性算出手法

本研究では電子化された情報を検索対象としているが，それらの情報はそのデータ形式によって検索手法が異なる．本節ではまず，それぞれのデータ形式における検索手法について述べる．次に，それぞれのデータ形式において，データ内容からデータ間の関連性を算出する手法について述べる．データ形式は大きくテキストデータとバイナリデータに分けられる．

本研究ではユーザのデータアクセスイベントとアプリケーションイベントを基に，データの関連性を算出している．過去の動作記憶を基に検索したい場合は本研究の方が適切である．逆に一般的な内容の関連性を基に検索したい場合は本節で述べた手法が有効である．

2.2.1 テキストデータ

テキストデータとは，文字データをある特定の文字コードでエンコードしたデータである．テキストデータのみでできたデータを特にプレーンテキストと呼ぶ．Microsoft Word で作成した文書や HTML ファイルなどはテキストデータと付加情報からなる．

検索エンジンは，ユーザが入力したキーワードをもとにテキストデータを検索し，そのキーワードが出現するデータを探し出す．検索エンジンではテキストデータを形態素解析を用いて検索精度の向上を行っている．形態素解析とは，自然言語処理の基礎技術の 1 つであり，自然言語で書かれた文章から品詞を見分ける作業である．現在，日本語や英語などで普及しているのは隠れマルコフモデルを用いた手法である．現在，Google や Yahoo!, Microsoft Search など多くの検索エンジンでは Basis Technology 社の Rosette 形態素解析システム [15] を採用している．

2.2.2 バイナリデータ

バイナリデータとは実行可能形式のコンピュータプログラムや，画像や音声，動画などのデータが該当する．これらのデータを検索する手法は，バイナリデータに付随したテキストデータを基に判断する手法とバイナリデータの特徴を付加情報としてデータに加える手法からなる．

前者の手法は HTML ファイルのようにテキストデータとバイナリデータからなるデー

タを検索する際に利用可能である．この手法を用いて Google は画像検索機能を提供しており，Yahoo!は画像，音声，映像の検索機能を提供している．

後者の手法はデータ種類ごとに様々な属性要素を定義し，データに付加情報を加える手法である．画像を例として挙げると物体の配置，形状，色特徴からメタ情報を抽出する．画像，映像，音声などのマルチメディア情報においては，メタ情報共通規格として MPEG-7[16] が定められている．MPEG-7 で採用されている技術規格の具体例として，NEC とサムスンが開発した顔認識技術が挙げられる [17].

2.3 既存の履歴検索効率化手法とその問題点

本研究では履歴検索の効率化を目的とする．本節では初めに既存の効率化手法について述べ．次にそれらの問題点について述べる．既存の効率化手法は大きく分けて，ユーザのフィードバックを基にデータを履歴データを分類する手法，レーティングする手法，履歴をビジュアル化する手法からなる．

2.3.1 履歴データの分類

ユーザがデータを保存するとき，データの種類によって分類することで，既読データの検索を容易にできる．具体的には Internet Explorer のお気に入りや保存フォルダの分類が該当する．しかし，保存データの種類が多様になるとこの作業は非常に困難になる．なぜならば，どちらにも分類できないデータや複数カテゴリに分類できるデータが出現するからである．また，ユーザが分類し間違えたデータは検索できない．

また，別のアプローチとして時系列で分類する手法が挙げられる [18]．この手法を用いれば，ユーザは手動で分類する必要はない．Internet Explorer の履歴検索や Google Desktop Search などほとんどの検索システムで時系列での検索が可能である．しかし，ある短期間内において，履歴データ中の重要なデータの割合が低くなればなるほど，重要なデータの検索が困難になる．

2.3.2 データのレーティング

データのレーティングとはユーザがデータの重要付けを行い，以降の検索に役立てる手法である．具体的には UJIKO[19] や My Yahoo! Search[20] が挙げられる．UJIKO を用いることで，ユーザは Web ページへのレーティングができ，検索結果をレーティング順に変えられる．また，必要のないページは次回以降の検索から非表示にできる．

しかし，この手法を用いるにはユーザが手動でデータのレーティングをしなければならない．従って，参照量が膨大になればなるほどそれらの作業が大変になる．

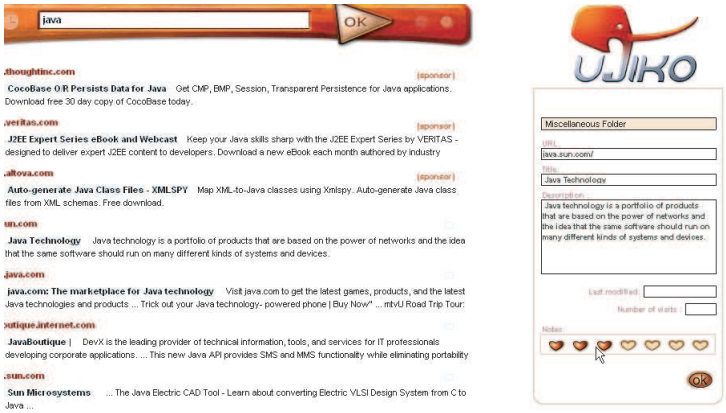

図 2.3: UJIKO

2.3.3 履歴のビジュアル化

ユーザが履歴データ検索を容易にするための手法の 1 つとして，履歴データをビジュアル化する手法が挙げられる．Internet Explorer のお気に入りをビジュアル化する手法としては Visual Marks[21] が挙げられる．Visual Marks を用いることでお気に入りのサムネイル表示ができる．またユーザが Web ページへアクセスした履歴をビジュアル化する研究として，Pad Prints[22] が挙げられる．Pad Prints はユーザがハイパーリンクをたどってアクセスした Web ページを木構造として表示できる．

履歴のビジュアル化は直感的でわかりやすいが，一般的に GUI を中心とするインターフェースはテキストデータを中心とした検索手法より動作が遅い．さらに必ずしもあらゆる状況で過去データへのアクセス速度を向上させる技術とは言えない．

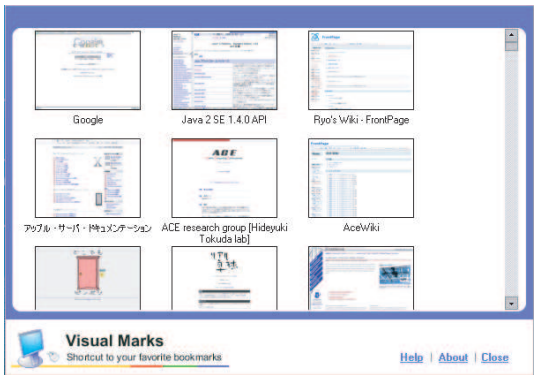

図 2.4: Visual Marks

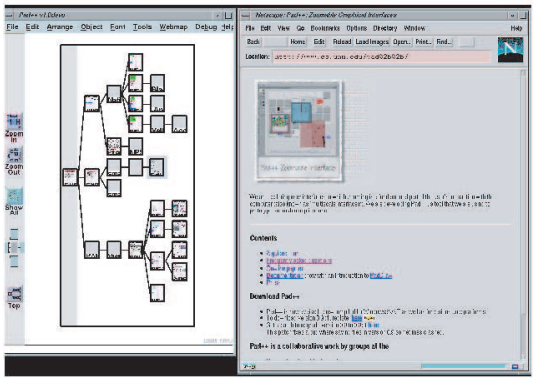

図 2.5: Pad Prints

2.4　本章のまとめ

本章ではまず，一般的なデータ検索手法と本研究との関連性を述べた．次にデータ内容を基にしたデータ間の関連性算出手法について述べ，本研究の手法と比較した．最後に，本研究の対象である履歴データ検索の効率化を行う既存手法とその問題点について述べた．次章では，本研究の目的とユーザ動作を基にしたデータ検索用メタ情報生成機構について詳しく述べる．

第3章　ユーザ動作を基にしたデータ間関連性とデータ重要度算出機構

本章ではまず，本研究の目的を実現するためのアプローチについて述べる．次に，そのアプローチの具体的な実現手法であるユーザ動作を基にしたデータ間関連性とデータ重要度算出機構について述べ，最後に本機構関の関連研究について述べる．

3.1 アプローチ

2.3 節で述べたように，既存の履歴検索を効率化する手法の問題点はユーザが手動で操作をしなければならない点である．本研究の目的は，普段のユーザに手動操作を要求しない手法で，履歴データ検索を効率化することである．本研究ではユーザ動作を基にデータ間関連度とデータ重要度を算出し，関連度を用いた関連検索機能とデータ重要度によるソート機能を既存の検索機能に付加する手法を提案する．本節では，それぞれの機能概要と機能を付加する理由について述べる．

3.1.1 関連検索機能

関連検索とは，あるデータに関連したデータを提供する手法である．具体的には，Google の関連ページ機能表示機能が挙げられる．既存システムの多くはデータの内容から関連度を算出しているが，本研究ではユーザ動作を基に関連度を算出する手法を用いる．ユーザ動作を基にした関連度を用いることで，検索システムはユーザに対して個々人の参照記憶を基にした関連検索を提供できる．例えば，ユーザは検索システムに対して「Word ファイル A を作成していたときに参照していた Web ページ」といった要求ができる．人間の記憶は連想からなるため，履歴検索においてはユーザの過去動作を基にした手法は有効である．

3.1.2 重要度によるソート機能

データ重要度によるソート機能とは，検索結果のデータを重要度順にソートしたり，重要度の低いデータに対して，フィルタをかける機能である．検索結果が膨大な場合，この機能は必要不可欠である．既存のソート手法としては，Google は Page Rank 機能を用いた検索結果のソート機能が挙げられる．既存システムの多くはデータの内容から重要度を算出しているが，本研究ではユーザ動作を基に重要度を算出する手法を用いる．ユーザ動作を基に重要度を算出する手法は，文章内容ではなくユーザの主観に基づいた重要度算出ができる．文章内容から算出される重要度とユーザが感じた重要度は異なるため，履歴データの重要度算出においてこの手法は有効である．

3.2 DMemFinder

本節では，前節で述べたアプローチの具体的な実現手法であるユーザ動作を基にしたデータ間関連性とデータ重要度算出機構について述べる．初めに本機構の概要を述べ，次にその機構の機能群について述べる．

3.2.1 概要

本研究ではユーザ動作としてアプリケーションイベントを取得し，データ間関連度とデータ重要度算出を動的に行うミドルウェア **DMemFinder(Deep Memory Finder)** を構築する．DMemFinder の概要を図 3.1 に示す.

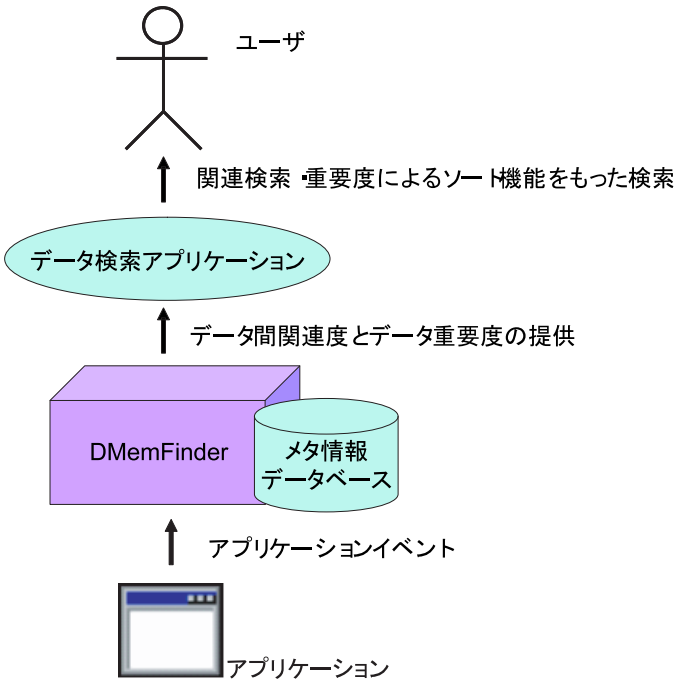

図 3.1: ユーザ動作を基にしたデータ間関連性とデータ重要度算出機構

まず，DMemFinder はアプリケーションイベントを取得し，データのメタ情報として保存する．そして，アプリケーションからの要求に応じて，メタ情報データベースからデータ間関連度とデータ重要度を算出し，アプリケーションに提供する．データ間関連度とデータ重要度を用いることでデータ検索アプリケーションは，関連検索と重要度によるソート機能をもった検索をユーザに提供できる.

3.2.2 アプリケーションイベントの取得

DMemFinder で監視対象とするアプリケーションイベントはデータアクセスイベントとアプリケーション固有イベントからなる．本節ではそれぞれのイベントについて詳しく述べる.

データアクセスイベント

データアクセスイベントとはユーザがデータを参照した動作を基にしたイベントである．具体的にはデータへのロード・アンロード，クリップボード利用，選択文字列反転,

テキスト内検索などが該当する．ユーザが席を離れていたるかどうかは，マウスとキーのイベントを監視し一定期間以上入力がなかった場合，席を離れているとし，参照時間から除外する．これらのメタ情報を用いるによりユーザは過去のデータアクセスイベントを基にした検索ができる．例えば「文書 A を作成中に，クリップボードのコピー先として利用していた Web サイト」といった検索が可能になる．

アプリケーション固有イベント

アプリケーション固有のイベントとは「メッセンジャーで同僚 A に話かけられた」や「音楽 B を再生していた」などが該当する．これらのイベントを用いることで,「同僚 A と話していたときに参照していたファイル」といった検索が可能になる．

また，特に近年情報科学の進歩によりユビキタスコンピューティング環境 [23] が実現されつつある．ユビキタスコンピューティング環境とは，様々なセンサや小型の計算機が人々の生活環境に埋め込まれ，ユーザに対してサービスを提供する環境である．本機構はそのようなアプリケーションにも対応可能である．具体的には，センサを付け部屋の入退出管理アプリケーションが挙げられる．本機構を用いて人の入退出イベントを監視すると「3 日前，友人 A と一緒に見ていた Web サイト」といった検索ができる．

3.2.3 データ間関連度の算出

本節では, 関連検索を実現するためのデータ間関連度算出手法について述べる．DMemFinder は以下の項目を基に関連性を算出する．

参照時刻

近い時刻に参照していたデータは関連が深いとする．しかし

テキスト内検索

同語句をデータ内で検索した場合，それらのデータ同士は関連が深いとする．

クリップボード

データ A でクリップボードにコピーし，データ B にペーストした場合，データ A とデータ B は関連が深いとする．

3.2.4 データ重要度の算出

データ重要度は参照時間，参照回数，クリップボードの利用，選択文字列反転回数から算出する．しかし，これらの値が常にユーザにとっての重要度と関連があるとは限らない．例えば，あるユーザは重要な資料に対して選択文字列反転をする癖があったとする．しかし別のあるユーザも選択文字列反転をするとは限らない．この問題を解決するため

にシステムに学習期間を設け，ユーザの癖を算出する手法を用いる．まず，学習期間中にユーザはデータに対して手動でも重要度を打ち込む．学習期間終了後，手動で打ち込まれた重要度とメタ情報の相関を求め，ユーザの癖を解析する．例えば，ユーザが重要度を高く設定したサイトの多くで選択文字列反転を多く行っていた場合，重要度と選択文字列反転に相関があるといえる．入力したデータ重要度とデータを読んでいるときに行った選択文字列反転回数の分布がユーザ A と B で図 3.2，図 3.3 のようになったとする．

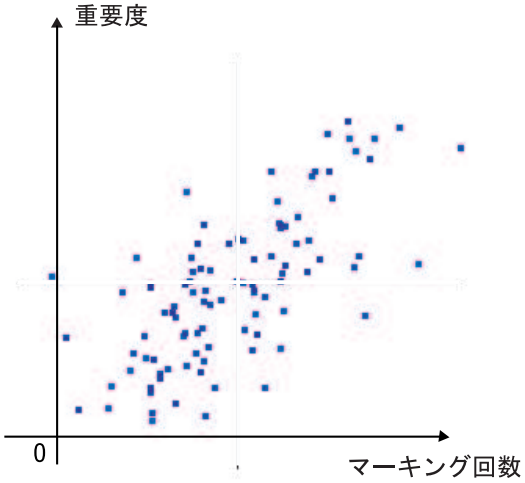

図 3.2: ユーザ A が付けた重要度と選択文字列反転回数

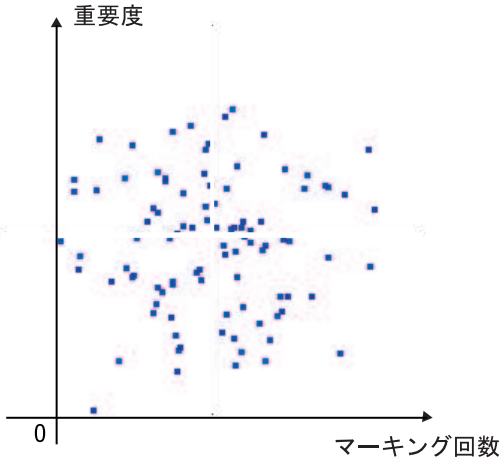

図 3.3: ユーザ B が付けた重要度と選択文字列反転回数

この 2 つの図で重要度と選択文字列反転回数の相関係数を算出すると

ユーザ A の相関係数　＞　ユーザ B の相関係数

となり，ユーザ A の方が選択文字列反転回数と重要度の相関が強いとなる．この相関係数の値を各メタ情報項目の重みとし，データに重要度を付ける．

3.3　関連研究

本節ではデータ間関連性算出とデータの重要度算出に関する先行研究を紹介し，それぞれと本研究の差異について述べる．

データ間の関連性算出を自動で行う先行研究として，大阪市立大学の前田氏による興味空間ブラウザ [24] が挙げられる．興味空間ブラウザの図を 3.4 に示す．興味空間ブラウザ上のアイコンがユーザが閲覧した Web ページを示し，ブラウザ上で近くにある Web ページ同士が関連が強い．興味空間ブラウザはユーザが閲覧した Web ページ中の語句を数量化　類を用いて解析し，Web ページ間の関連性を算出する．本研究では，ユーザ動作を基に関連性を算出しデータ中の語句解析を行っていないため，語句解析手法と並列拡張できる．

図 3.4: 興味空間ブラウザ

データの重要度算出を自動で行う先行研究として，関西大学の原田氏による Web 検索履歴を用いたブラウジング支援ツールが挙げられる [25]．このツールは，ユーザが過去に閲覧した Web ページの重要度を閲覧回数，閲覧日時，参照時間から算出する．しかし，全てのユーザに対して必ずしもこの要素が重要度と相関があるとは限らない．本研究では，学習期間を設け要素と重要度の相関係数を求めており，その点ではよりユーザ個々人の癖に対応している．

3.4 本章のまとめ

本章ではまず，本研究の目的を実現するためのアプローチについて述べた．次に，そのアプローチの具体的な実現手法であるユーザ動作を基にしたデータ間関連性とデータ重要度算出機構について述べ，最後に本機構関の関連研究について述べた．次章では，DMemFinder の設計について詳しく述べる．

第4章　DMemFinderの設計

本章ではDMemFinderの設計について詳細に述べる．まず，システム全体の概要について述べる．次にDMemFinderを構成するそれぞれのモジュールについて述べる．

4.1　設計概要

本節ではまず，DMemFinder のシステム構成と各モジュールの動作概要を述べる．

4.1.1　システム構成

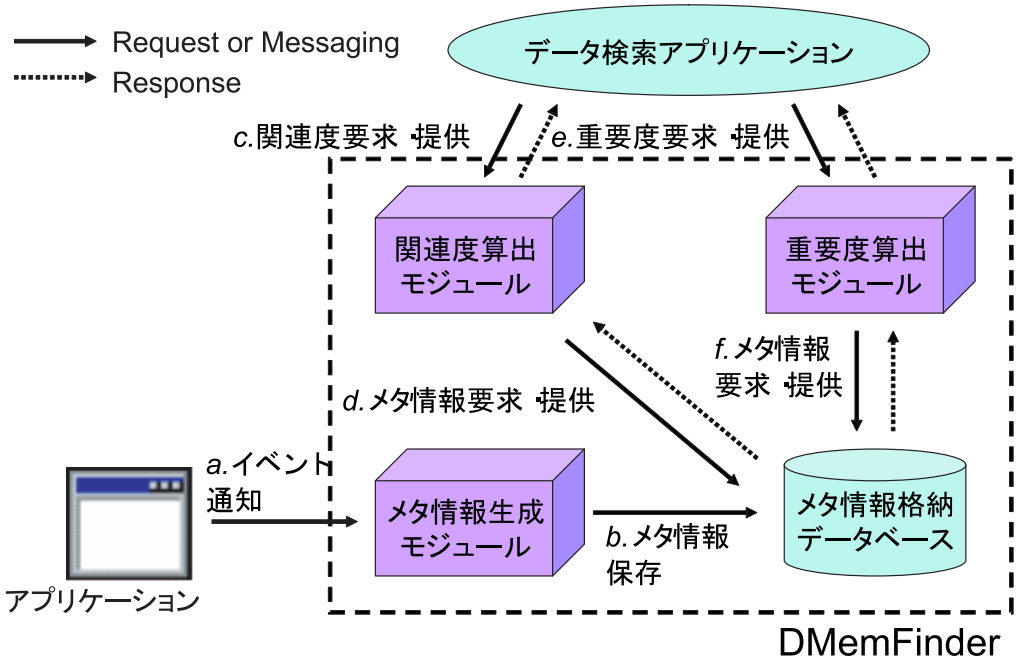

図 4.1: システム構成と基本動作

DMemFinder のシステム構成を図 4.1 に示す．DMemFinder はメタ情報生成モジュール，関連度算出モジュール，重要度算出モジュール，メタ情報格納データベースからなる．メタ情報生成モジュールはアプリケーションイベントを基にデータのメタ情報を生成する．メタ情報格納データベースはデータのメタ情報を保存し，他モジュールからのリクエストに応じてメタ情報を提供する．関連度算出モジュールと重要度算出モジュールはデータ検索アプリケーションからの要求に応じて，それぞれデータ間関連度，データ重要度を提供する．

4.1.2　動作概要

次に DMemFinder の動作概要を述べる．DMemFinder の基本動作は大きく分けて，メタ情報格納，データ間関連度算出，データ重要度算出からなる．以下にそれぞれの概要と各モジュールの関連を述べる．

メタ情報格納

DMemFinder は様々なイベントを取得し，その情報を基にデータのメタ情報を生成する．メタ情報生成モジュールはアプリケーションを監視しイベントを取得する (図 4.1 a)．イベントはユーザがデータへアクセスしたイベントとアプリケーション固有イベントからなる．前者は，データのロード，アンロード，ウィンドウフォーカスの移動，データ文字列の反転やクリップボードの利用などデータに関連するイベントが該当する後者は，人の入退出イベントや再生音楽の切り替えなど直接ユーザが閲覧しているデータに関連しないイベントからなる．そして，メタ情報生成モジュールはそれらのイベント情報をユーザが現在閲覧しているデータのメタ情報としてメタ情報格納データベースに保存する (図 4.1 b).

データ間関連度算出

DMemFinder はアプリケーションからの要求に応じて，データ間関連度を提供する．データ間関連度とは，2 つのデータの関連の強さを表した値である．関連度算出要求を受け取った関連度算出モジュールは，関連度を算出したいデータそれぞれのメタ情報をメタ情報格納部から取得する (図 4.1 c)．関連度算出モジュールは，取得したメタ情報の内，データ間の参照時刻や反転文字などを比較する．そして，それらの値からデータ間関連度を算出し，データ検索アプリケーションに送る (図 4.1 d).

データ重要度算出

DMemFinder はアプリケーションからの要求に応じて，データ重要度を提供する．データ重要度とは，ユーザの過去動作を基にデータにレーティングをした値である．データ重要度算出要求を受け取ったデータ重要度算出モジュールは，重要度を算出したいデータのメタ情報をメタ情報格納部から取得する (図 4.1 e)．データ重要度算出モジュールは，取得したメタ情報の内，データの参照時間，参照回数，反転文字数，クリップボード利用数などを比較する．そして，それらの値からデータ重要度を算出し，データ検索アプリケーションに送る (図 4.1 f).

4.2 各モジュールの詳細

本節では DMemFinder 各モジュールの詳細について述べる．

4.2.1 メタ情報生成モジュール

メタ情報生成モジュールはアプリケーションイベントを取得し，その情報を基にデータのメタ情報を生成する．メタ情報生成モジュールは，まずアプリケーションイベントの内，データのロード，アンロード，ウィンドウフォーカスの移動からユーザが現在閲覧してい

るデータを判断する．そして，データ文字列の反転，クリップボードの利用や人の入退出イベントを受け取った際に，それらのイベント情報をユーザが現在閲覧しているデータのメタ情報としてデータベースに保存する．アプリケーションからイベントを取得する手法には，アプリケーションに改良を加える手法とアプリケーションイベントをフックする手法がある．前者の例としては，各種プラグインをインストールする手法が挙げられる．後者の例としては，キーボードやフォーカスの移動といったOSからの命令をフックする手法が挙げられる．また，キーボードとマウスのイベントから現在ユーザがPCから離れていないかどうかの判別を行う．ユーザがPCから離れていた場合，その時間をデータ参照時間から除外する．メタ情報の格納手法にはファイルとして保存する手法やリレーショナルデータベースに保存する手法がある．

4.2.2 関連度算出モジュール

関連度算出モジュールはアプリケーションからの要求に応じて，データ間関連度を提供する．関連度算出要求を受け取った関連度算出モジュールは，関連度を算出したいデータそれぞれのメタ情報をメタ情報格納データベースから取得する．関連度算出モジュールは，取得したメタ情報の内，データ間の参照時刻や反転文字などを比較する．そして，それらの値からデータ間関連度を算出し，データ検索アプリケーションに送る．関連度は参照している時間が近いデータ同士の関連が近いとする．かつ，普段から見ているサイトはレートを下げる．

ユーザがデータAにアクセスした時刻が

$$(a_1, a_2, \cdots, a_n) \quad (n \text{ は正の整数})$$

であり，データBにアクセスした時刻が

$$(b_1, b_2, \cdots, b_m) \quad (m \text{ は正の整数})$$

であるとき，データA，B間の関連度R_{ab}は

$$R_{ab} = \frac{\sum_{k=1}^{n} \min(|a_k - b_1|, |a_k - b_2|, \cdots, |a_k - b_m|)}{n}$$

で求める．

4.2.3 重要度算出モジュール

DMemFinderはアプリケーションからの要求に応じて，データ重要度を提供する．データ重要度は，ユーザの過去動作を基に算出される．あるデータの重要度算出要求を受け取ったデータ重要度算出モジュールは，そのデータのメタ情報をメタ情報格納部から取得する．そして取得したメタ情報の内，クリップボード利用回数，選択文字列反転回数，ア

クセス回数，総合アクセス時間からデータ重要度を算出しデータ検索アプリケーションに送る．

データ X のクリップボード利用回数が x_c，選択文字列反転回数が x_m，アクセス回数が x_a，総合アクセス時間が x_t であり，それぞれの項目の重みが r_c，r_m，r_a，r_t のときデータ X の重要度 R_x は，

$$R_x = r_c x_c + r_m x_m + r_a x_a + r_t x_t$$

で求める．

重みは学習期間を設け，その期間内でのユーザの振るまいによって算出する．学習期間中，ユーザはアクセスしたデータに重要度の値を付けていく．項目 Y の値 y とユーザが付けていった重要度 r が

$$(y_1, r_1), (y_2, r_2), \cdots, (y_n, r_n) \quad (n \text{ は正の整数})$$

のとき，項目 Y の重み r_y は

$$r_y = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(r_i - \bar{r})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \bar{r})^2}}$$

で求める．

4.3 本章のまとめ

本章では DMemFinder の設計について詳細に述べた．そして，システム全体の概要について述べ，次に DMemFinder を構成するそれぞれのモジュールについて述べた．次章では，DMemFinder の実装について詳しく述べる．

第5章　DMemFinderの実装

本章ではDMemFinderの実装について述べる．まず，実装の概要について述べ，次に各モジュールの実装について詳しく述べる．

5.1 概要

本節では DMemFinder の実装概要を述べる．DMemFinder が監視するアプリケーションとして，Mozzila Firefox[26] と OpenOffice.org[27] を用いた．Firefox と OpenOffice を用いた理由は両者とも OS に依存せず動作し，DMemFinder をクロスプラットフォームに改良するとき，便利だと判断したからである．Firefox からイベントを取得するために Firefox エクステンションを作成し，OpenOffice からイベントを取得するために OpenOffice ServiceManager を利用した．また，データ検索アプリケーションとして DMemSearch を実装した．DMemSearch はユーザに対して関連検索や重要度順検索を提供する．DMemFinder とそれぞれのアプリケーションの関係を図 5.1 に示す．

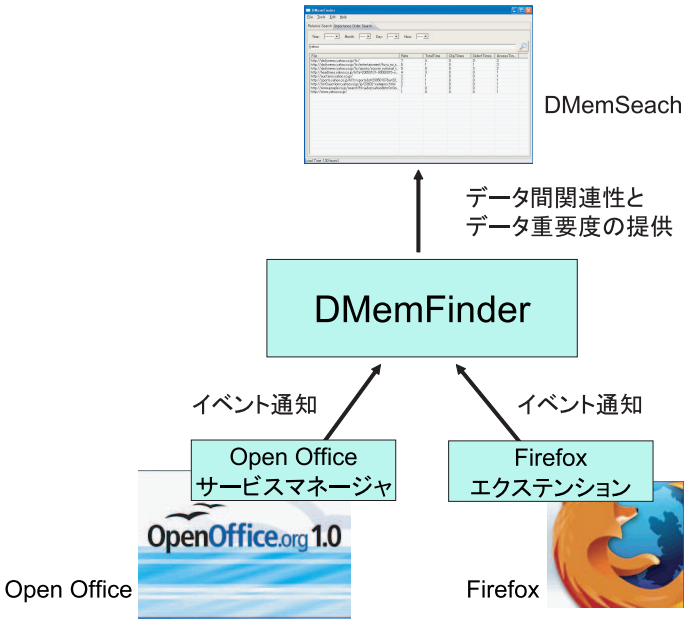

図 5.1: DMemFinder とアプリケーションの関係

5.2 実装環境

本節では DMemFinder の実装環境について述べる．まず，利用した Windows および Firefox と OpenOffice のバージョンを図 5.1 に示す．

開発言語を表 5.2 に示す．開発は主に Java 2, Standard Edition Development Kit で行った．一部 Windows 特有の機能を利用するために VisualC++を利用した．具体的にはキーイベントフックとウィンドウフォーカスフックを VC++で実装した．また，Firefox エクステンションを作成するために Java Script を用いた．さらに，DMemSearch の UI は軽量化のため Eclipse.org[28] の SWT(Standard Widget ToolKit)[29] を用いた．

表 5.1: 実装環境

名称	バージョン
Microsoft Windows	XP Professional SP2
Mozilla Firefox	1.0
OpenOffice.org	1.1.3

表 5.2: 開発環境

名称	バージョン
Java 2, Standard Edition Development Kit	5.0
VisualC++	6.0
JavaScript	1.5
Eclipse.org Standard Widget ToolKit	3.1 M4

5.3 DMemSearch の実装

本節では作成したデータ検索アプリケーション DMemSearch のユーザインターフェースについて述べる.

5.3.1 関連度検索

5.3.2 重要度検索

5.4 本章のまとめ

本章では DMemFinder の実装について述べた．そして，実装の概要について述べ，次に各モジュールの実装について詳しく述べた.

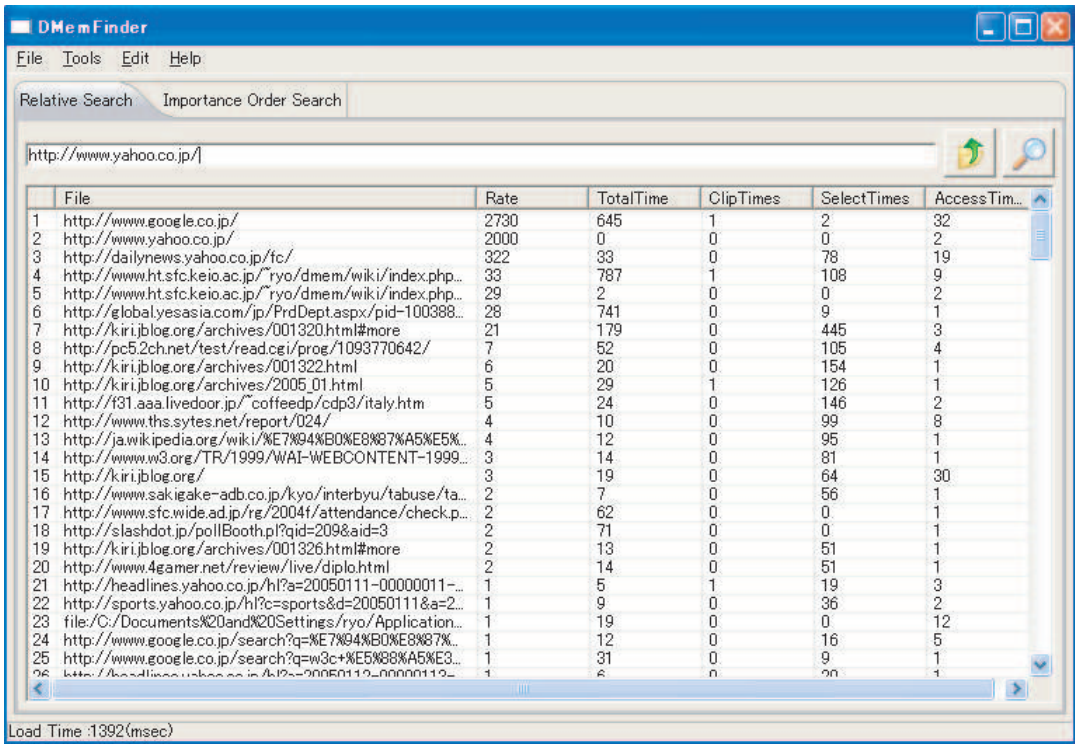

図 5.2: 関連度検索

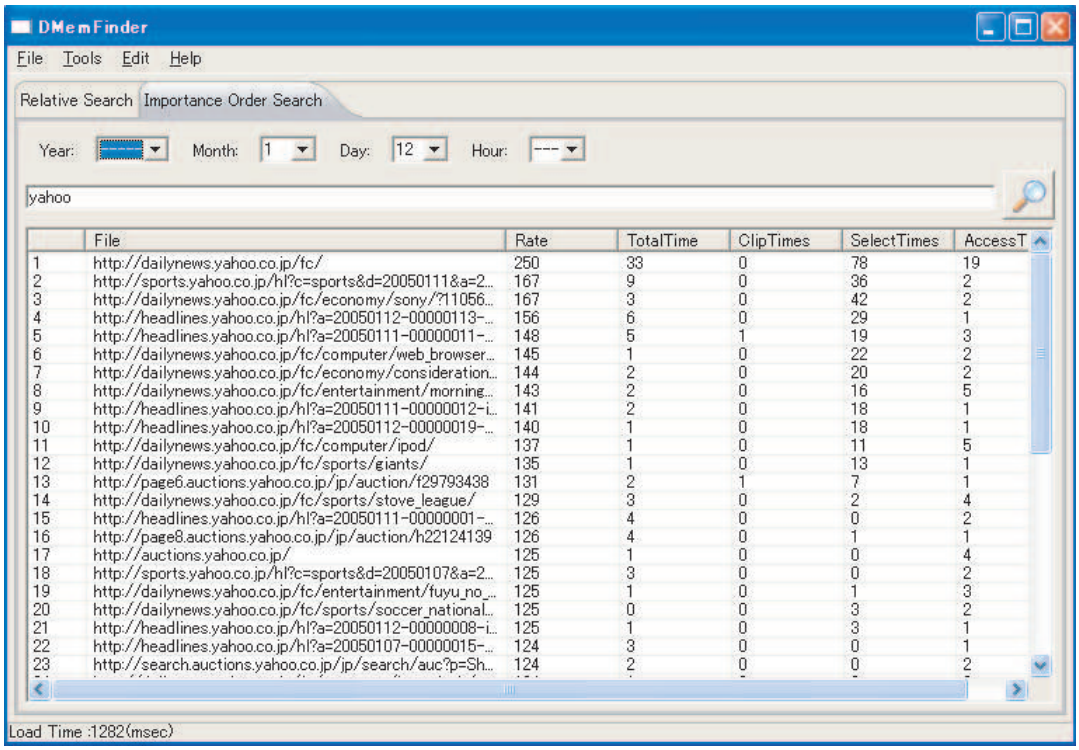

図 5.3: 重要度検索

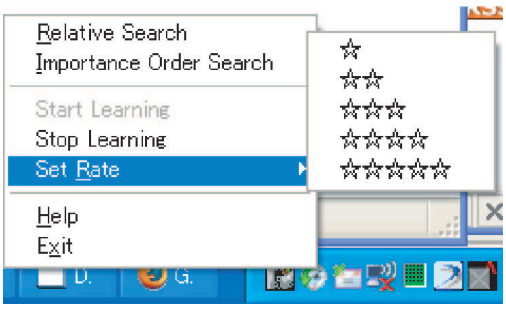
Relative Search
Importance Order Search
Start Learning
Stop Learning
Set Rate
Help
Exit
☆
☆☆
☆☆☆
☆☆☆☆
☆☆☆☆☆

図 5.4: レーティング

第6章　評価

本章では，DMemFinderについて拡張性評価と有用性評価を行う．

6.1 拡張性評価

DMemFinder はユーザの過去履歴検索の支援が目的であり，過去履歴に関するメタ情報を保存しなければならない．そこで本節では，保存したデータの量と検索応答速度の関係を調べ，拡張性を評価する．

6.1.1 手法

まず，保存データ量が増えたときに関連検索と重要度検索それぞれにどの程度時間がかかるかを計測する．そして，次にプログラムを故意的に遅らせ，検索応答にどの程度遅延が生まれるとストレスになるかをユーザアンケートを行いて調べる．最後にユーザにストレスを与えずに検索機能を提供する実装の最適化手法について考察する．実験環境を表 6.1 に示す.

表 6.1: 実験環境

項　　目	環　　境
CPU	Pentium 4 2.8EGHz
FSB	800MHz
L1 キャッシュ	16KB
L2 キャッシュ	1MB
メモリ	512MB
ハードディスク回転速度	7200rpm (SATA)
OS	WindowsXP Professional SP2

6.1.2 結果と考察

関連検索と検索時間，重要度検索と検索時間のそれぞれの関係を図，図に示す.

また，検索遅延時間とユーザのストレスとの関係を調べたアンケート結果を表 6.2 に示す.

結果より履歴がほげページを超えたときに全体の半数以上のユーザがストレスを感じるといえる．もし，過去の検索を対象範囲から外すとした場合ページ程度内を検索するのが望ましい．Nielsen//NetRatings 社の調査結果 [**?**] によると，平均的なユーザの 1 月あたりの Web ページ閲覧数は 911 ページ (2004 年 10 月) という報告がなされており，約週間使用する頃に遅延を感じてくると想定できる．もし，過去の検索を対象範囲から外すとした場合，週間分程度が望ましくなる．コンピュータスペックは個々人によって違う点，今回はプロトタイプ実装であり，保存方法を最適化していく必要がある．どの程度のデータ量

表 6.2: 検索遅延時間とストレス

遅延時間 (msec)	許容できる範囲である	やや遅く感じる	遅く感じる
500	%	%	%
1000	%	%	%
1500	%	%	%
2000	%	%	%
2500	%	%	%
3000	%	%	%
3500	%	%	%
4000	%	%	%

で 1 つのファイルにまとめるとよいとの考察をまとめる．(今は 1 履歴につき 1 ファイル作っている)

6.2 有用性評価

本節では DMemFinder が有効にユーザの履歴検索を支援するのかについて，検証を行う．まず，今回の実装で利用したメタ情報の項目が正当かどうか検証する．つぎにそれらのメタ情報を利用した重要付けがユーザの意図にそうかの検証を行う．

6.2.1 利用したメタ情報の有用性評価

まず，今回の実装で利用したメタ情報の項目が正当かどうか検証する．実験対象ユーザ情報を表 6.3 に示す．

表 6.3: 対象ユーザ情報

利用人数	人
男女比	:
年齢平均	歳
1 日の PC 利用時間平均	時間
PC 利用期間平均	年

表 6.4 に示す．
分散が大きい場合，個々人の癖が大きい癖学習の必要性がある．
分散が小さい場合，個々人の癖がないできあいのフィルタでもいいかもしれない

表 6.4: メタ情報と重要性の相関係数

項目	平均値	中央値	分散
データアクセス回数			
データアクセス時間			
クリップボード利用回数			
文字反転回数			

相関係数の平均が 0．1 未満ほかつ分散が低い場合，その項目はほとんど意味がない

6.2.2　検索の有用性評価

目的

データを指定すると関連データが上位にソートされるのかを調べる．

手法

1. Amazon を用い，商品を 15 分間探してもらう．この際，DMemFinder を学習モードにし，全ページにレーティングをしていく．
2. 学習モードを中止し，重み計算を行う．
3. DMemFinder を普段利用しているユーザにて，Web を参考にしながら OpenOffice Writer でレポートを作ってもらう．
4. 「曙の k － 1 戦績を詳細に A4 一枚程度にまとめよ」というのをテーマとしてユーザに与える．
5. 実際レポートを作りに役にたったページはその場でブックマークに入れていく
6. レポートのタイトルを用いて関連検索を行い，ブックマークしたページの順位の平均を求める．
7. レポートを作成していた時間を基準に重要度検索を行い，ブックマークしたページの順位の平均を求める．

6.3　本章のまとめ

本章では，DMemFinder について定性的評価と定量的評価を行った．次章では，今後の課題を述べ，最後に本論文をまとめる．

第7章　結論

本論文では，グラフ表現を用いたロケーションモデル G-LoM を動的に生成し，システム管理者によるロケーションモデル作成を支援する NiSMo を設計し実装した．本章では今後の展望を挙げ，最後に本論文をまとめる．

7.1 今後の展望

本節では，本研究のこれからの展望について述べる．

7.1.1 ユーザ動作となるメタ情報項目の追加

今回はユーザのデータアクセス時間，アクセス回数，文字列選択回数，クリップボード使用回数でデータ間関連度とデータ重要度を算出した．しかし，さらに多くの要素を検証していく必要がある．例えば，ユーザの視線を利用し，データの重要度を算出する先行研究もある [30]．データの重要度ユーザがある領域を長時間見ていた場合や何度も見ていた場合，領域注目度は徐々に増加していく．

7.1.2 データの動的分類

履歴データを分類する上で大変なのが分類作業である．この分類作業を自動化する様々な手法が現在までに考案されてきた [31]．本機構を用いることユーザ動作を利用した自動分類ができる．例えば，頻繁に訪れるが一回あたりの滞在時間が短い Web ページを「確認用ページ」と定義すれば,「確認用ページ」に分類されたページは常に監視し，変更があったらユーザに知らせるといったアプリケーションが作成できる．

7.1.3 メタ情報共有化

他人との情報の共有化を計りデータ検索をグループ内で最適化するアプリケーションも考えられる．共有化を行うことで友人にとってが重要度だと判断した Web サイト一覧を検索できる．しかし，他人と共有する場合は，ユーザプライバシを保護するために適切なアクセス権限機構が必要になる．先行研究として，友人同士でブックマークを共有し合う機構がある [32].

7.1.4 作業状況分析

企業の端末に本システムを導入し，重要度の上位に仕事と関係ないサイトが来ていないかを監視し，個々人の仕事作業評価を行える．既存のシステムではキー入力数を監視したり，トラフィック制限をかけたりする手法があるがこのシステムならば，より管理対象者の仕事状況がわかる．

7.2 まとめ

本論文では，システム管理者によるロケーションモデル作成を支援する NiSMo を設計，実装し，評価した．

謝辞

本研究の機会を与えてくださり，ご指導を賜りました慶応義塾大学環境情報学部教授徳田英幸博士に深く感謝いたします.

慶応義塾大学徳田・村井・楠本・中村・南合同研究会の先輩方には折りにふれ貴重な指導と助言を頂きました．特に，徳田研究室の先生方や先輩方，ACE(Active Computing Environmets) 研究グループの方々に深く感謝いたします．また，桐原幸彦氏，青木崇行氏，中西健一氏，岩谷晶子氏，村上朝一氏，出内将夫氏には絶えざる励ましや丁寧なご指導をを賜りました．須之内雄司氏，松倉友樹氏，駒木亮伯氏の多大な協力に感謝します.

最後に，本研究を通じて様々経験や刺激を受ける機会を頂きましたことに，深く謝意を表します.

平成 17 年 1 月 22 日

大澤 亮

参考文献

[1] 総務省統計局. http://www.stat.go.jp/.

[2] Penn State's School of Information Sciences and Technology. Cite seer. http://citeseer.ist.psu.edu/.

[3] Linda Tauscher and Saul Greenberg. How people revisit web pages: Empirical findings and implications for the design of history systems. International Journal of Human Computer Studies, Special issue on World Wide Web Usability, 47(1), p97-138, 1997.

[4] Marvin Minsky. *The society of mind*. Simon & Schuster, Inc., 1986.

[5] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, Vol. 30, No. 1-7, pp. 107–117, 1998.

[6] Google. Google personalized search. http://labs.google.com/personalized/.

[7] Google. Google scholar. http://scholar.google.com/.

[8] W3C. Resource description framework(rdf). http://www.w3.org/RDF/.

[9] Deborah L. McGuinness and Frank van Harmelen. Web ontology language overview. http://www.w3.org/TR/owl-features/, 2 2004.

[10] Google. Google desktop search. http://desktop.google.com/?promo=app-gds-en-us.

[11] Copernic Technologies Inc. Copernic desktop search. http://www.copernic.com/en/products/desktop-search/.

[12] Ask Jeeves Inc. Ask jeeves desktop search. http://sp.ask.com/docs/desktop/.

[13] Microsoft Corporation. Msn desktop search. http://beta.toolbar.msn.com/.

[14] X1 Technologies Inc. X1 desktop search. http://www.x1.com/.

[15] Basis Technology Corp. Rosette linguistics platform. http://www.basistech.com/.

[16] Jose M. Martinez. Mpeg-7 overview. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, 3 2003.

[17] NEC and Samsung. 階層的判別分析法, 顔部品特徴表現方式. http://www.nec.co.jp/press/ja/0312/1601.html, 12 2003.

[18] 野口悠紀雄. 「超」整理法—情報検索と発想の新システム. 中公新書, 11 1993.

[19] Ujiko. http://www.ujiko.com/.

[20] Yahoo! Inc. My yahoo! search. http://mysearch.yahoo.com/.

[21] 6Bytes Software. Visual marks. http://www.6bytes.com/bookmark_manager/about_visual_marks.html.

[22] Ron R. Hightower, Laura T. Ring, Jonathan I. Helfman, Benjamin B. Bederson, and James D. Hollan. Padprints: graphical multiscale web histories. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pp. 121–122. ACM Press, 1998.

[23] Mark Weiser. The computer for the 21st century. Scientific American 256(3), September 1991.

[24] MURAKAMI Harumi and HIRATA Takashi. A system for generating user's chronological interest space from web browsing history. International Journal of Knowledge-Based Intelligent Engineering Systems Vol.8 No.3, 2004.

[25] Takehiko Ohno. Impact: Eye mark reusing technique to support information browsing task. Proceedings of The 8th Workshop on Interactive Systems and Software (WISS'2000), 2000.

[26] The Mozilla Organization. Mozzila firefox. http://www.mozilla.org/products/firefox/.

[27] Inc Sun Microsystems. Openoffice.org. http://www.openoffice.org/.

[28] eclipse.org. Eclipse. http://www.eclipse.org/.

[29] eclipse.org. Swt: The standard widget toolkit. http://www.eclipse.org/.

[30] Takehiko Ohno. Eyeprint: support of document browsing with eye gaze trace. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pp. 16–23. ACM Press, 2004.

[31] Yoelle S. Maarek and Israel Z. Ben Shaul. Automatically organizing bookmarks per contents. In *Proceedings of the fifth international World Wide Web conference on Computer networks and ISDN systems*, pp. 1321–1333. Elsevier Science Publishers B. V., 1996.

[32] Alessandra Alaniz Macedo, Khai N. Truong, José Antonio Camacho-Guerrero, and Maria da GraÇa Pimentel. Automatically sharing web experiences through a hyperdocument recommender system. In *HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pp. 48–56. ACM Press, 2003.