

## 大会優秀賞

## 第54回全国大会（平成9年前期 於：千葉工大）

### 大会優秀賞受賞論文（11件）

大会優秀賞は、全国大会で発表された全論文の中から大会優秀賞・大会奨励賞選定委員会で厳選する審査の上理事会に推薦し承認をえて受賞論文を決定いたしました。

全国大会には、大会優秀賞（年齢制限なし）・大会奨励賞（大学学部卒業後10年以内）の表彰制度があり、大会優秀賞受賞論文は学会誌に掲載し広く会員の皆さまへ紹介します。

現在、第56回全国大会（平成10年前期 於：中大理工学部）の講演申込を受付中です。奮ってご応募ください。

タイトル

著者（○印受賞者（登壇発表者））

1. 情報検索のための高速日本語形態素解析システム「すもも」  
○鷲坂光一，山崎憲一，廣津登志夫，尾内理紀夫（NTT）
2. インターネットにおけるトラフィック分析 ○小松原重之，鈴木亮一，三上博英（NTT）
3. ソフトウェア性能評価のためのランダムデータ高速生成法  
○二村良彦，大谷啓記，青木健一（早大）
4. データフロー解析に基づくプログラム保守支援 ○四野見秀明，藤井邦和，高橋真由美（IBM）
5. トランザクション処理によるディスクアクセスのトレースを用いた Hot mirroring の性能評価  
○茂木和彦，喜連川優（東大）
6. モバイルグループウェアシステム「なかよし」の構想  
○倉島顕尚，市村重博，田頭 繁，前野和俊（NEC）
7. 商用機械翻訳ユーザ辞書の共通フォーマット設定に向けて  
○亀井真一郎（NEC），平井德行（シャープ），斎藤由香梨（富士通研），伊藤悦雄（東芝），  
赤羽美樹子（ノヴァ），松山 努（NEC 情報システムズ），村木一至（NEC）
8. 画像データのカプセル化による著作権管理手法  
○木俣 豊（通信・放送機構），田中克己，上原邦昭（神戸大）
9. 小型情報機器のための Scroll Display ○椎尾一郎（IBM）
10. 時系列情報を考慮したアクセスログ解析 ○畑島 隆，元田敏浩（NTT）
11. 拡張有限状態機械とベトリネットを表示編集できる GUI ツールの作成と応用例  
○岡野浩三，山本 亮，東野輝夫，谷口健一（阪大）

# 情報検索のための高速日本語形態素解析システム「すもも」

鷺坂光一 山崎憲一 廣津登志夫 尾内理紀夫

NTT 基礎研究所

## 1 はじめに

これまで日本語形態素解析は、機械翻訳などを行うための最初のステップとして用いられることが多かった。ここでの解析間違いは当然のことながら後のステップの解析精度に影響を及ぼす。このため、これまでは形態素解析の精度の向上を主眼として研究が進められてきており、計算量についての研究なども幾つか見られるもの<sup>1)</sup>、実システムの速度はそれほど重視されていなかった。しかし、形態素解析を情報検索のような大量の文書処理に適用する場合、処理の高速性が極めて重要となる。本論文では、応用の立場から形態素解析の速度向上の重要性を指摘するとともに、我々が実装したシステム「すもも」における高速化技法を述べ、最後に評価結果を示す。

## 2 大量文書処理のための形態素解析

インターネット技術の普及にとともに、電子的にアクセス可能なデータは増え続けており、例えば World Wide Web(WWW)によりアクセス可能なページは6000万ページ以上あると言われている。仮に1ページ当たりのテキストサイズが1Kバイトだとすると、これは60Gバイトに相当する。もちろん日本語で書かれたページは一部にしか過ぎないが、今後ますます増大していくことは明らかである。WWWのページで特徴的なことはデータが日々更新されるという点である。このためWWWの空間から目的のページを探す情報検索サービスは、日々、更新されたページを処理して検索インデックスに追加する、あるいは検索インデックスを再構築する必要がある。

日本語で記述されたページから検索インデックスを作る方法は幾つかあるが、形態素解析をすれば、より高度で精度の高い検索ができる可能性がある。ここで問題となるのが、ギガバイト単位のデータ処理に要する形態素解析の時間である。上で述べたような情報検索サービスでは、形態素解析と検索インデックスの作成をデータ流入量を上回る速度で行う必要がある。後者については十分高速なシステムが存在するものの、形態素解析については少なくとも我々の知るところでは存在しない。

我々は、ギガバイト単位のデータを実用的な時間内に処理できることを目標に形態素解析システム「すもも」

を設計、実装した。以下では「すもも」において用いられた高速化技法とその評価について述べる。

## 3 形態素解析の高速化

形態素解析のアルゴリズムには、コスト最小法<sup>2)</sup>を用いた。コスト最小法は形態素解析の方法としてよく知られた手法であり、多くのシステムがこれを用いているが、実装方法によって速度は大きく異なる。

### 3.1 解の候補の決定方法

一般にコスト最小法を用いる場合、接続しうる形態素間にコスト付きリンクを張ったグラフを作り、その中でコスト最小のパスを捜し出す。この方法は次に小さいコストを持つパスを知る必要がある場合には意味がある。例えば、仮名漢字変換のための形態素解析では、ユーザがそのパスは正しくないかと判断したら次のパスを出力しなければならない。このように形態素解析の結果を判断するには何らかの意味的な解析が必要となる。一方、情報検索においては、何ギガバイトもの解析結果を意味的に判断することは不可能である。このため、情報検索用の形態素解析システムではただ一つの解析結果を出力すれば十分だと考えられる。

「すもも」では、ある文字列が辞書にあった時、その形態素の前に接続しうる形態素のうち、最もコストが小さいものだけにリンクを張り、それ以外を捨てている。文末まで解析が到達すると同時にコスト最小のパスが得られるため、グラフを調べ直す必要はない。

### 3.2 未定義語の扱い

未定義語の扱いについても、情報検索においては、正しく未定義語が取り出せたことを判断する主体がない。また、質問の中の未定義語も同じように誤解析をすれば検索には特に支障がない、と考えられる。

このため未定義語を完全に扱うことは、本応用ではそれほど重要ではない。「すもも」では、連続するカタカナ、英字、数字などを一つの形態素として扱う。これにより外来語や製品番号などの未定義語はほぼ回避可能である。また、接続しうる形態素がなくなって文末まで解析が進められないという条件が成立した場合のみ未定義語探索を行うこととした。この条件のチェックは少ない手間で行うことができるため、速度に影響を与えない。現在の未定義語探索アルゴリズムは、接続不能となった形態素を未定義語とするという非常に単純なものである。

Sumomo - a fast morphological analyzer for information retrieval  
Mitsukazu WASHISAKA (wasisaka@rouge.br1.ntt.co.jp),  
Ken-ichi YAMAZAKI (yamazaki@nuesun.br1.ntt.co.jp),  
Toshio HIROTSU (hirotsu@square.br1.ntt.co.jp),  
Rikio ONAI (onai@square.br1.ntt.co.jp),  
NTT Basic Research Laboratories.

### 3.3 辞書の構造と読み込み

辞書の構造としては、辞書の中を1回探索するだけですべての形態素候補を見つけることができるようにトライ構造<sup>3)</sup>を用いた。しかし、トライ構造をどの様に構成するかによって、辞書引きの速度は大きく異なる。「すもも」の実装にあたっては、次の方法を採用した。

- 長さ1の形態素は、パスのすべての分岐で探索されるため、テーブルで直接アクセスする。
- 長さ2以上の形態素は、最初の2文字をもとにハッシュ表に入れ、3文字目以降をトライ構造とする。

辞書の読み込みに関しても、従来のシステムでは辞書全部を読み込まずに、必要な部分だけをキャッシュするようにしていた。これは、メモリの少ない計算機では使用メモリ量を抑えるという点で意味があったが、大容量メモリが計算機に搭載されるようになった現在では、解析速度を低下させる原因になっている。辞書のサイズが計算機のメモリサイズに比べて比較的小さいのであれば、辞書全部をプロセスのメモリ空間にマッピングし、読み込みはページフォールトに任せた方が、キャッシュを管理する手間もなくアクセス速度も速い。このため「すもも」では、mmap関数を用いて辞書全部をマッピングする方法を採用した。

## 4 性能評価

### 4.1 解析速度

すもも、茶筌 1.0b5, Juman 2.0の解析速度を比較した。使用した計算機はSun SPARCstation20(SunOS 4.1.4, 主記憶 224MB, クロック 75MHz), プログラムはgcc 2.7.2で-Oオプションを付けてコンパイルした。測定にはtimeコマンドを用い、10回測定した後、最高値と最悪値を除外した残りの平均値をCPU時間(ユーザ時間+システム時間)とした。

	解析方法	辞書語彙数	辞書サイズ
すもも	コスト最小法	約 35 万語	約 7.5 MB
茶筌 1.0b5	コスト最小法	約 11 万語	約 7.5 MB
Juman 2.0	コスト最小法	約 11 万語	約 51 MB

表 1: 比較対象プログラム

	CPU 時間 (秒)	解析速度 (バイト / 秒)
すもも	13.3	152746.2
茶筌 1.0b5	159.9	12744.8
Juman 2.0	1309.1	1556.2

表 2: 実行速度 (テキストサイズ 2037253 バイト)

使用している日本語辞書の違いにより、形態素解析の出力結果は若干の異なるものの、すももでは茶筌 1.05bの12倍、Juman 2.0の98倍の解析速度をあげることができた。なお、参考までにSUN Ultra-2(200MHz)では290Kから300Kバイト/秒を得ている。

### 4.2 解析精度

現時点ではコストの調整を行っている段階であり、以下に述べる解析精度は一時的なものである。

日本語辞書にはEDR日本語単語辞書<sup>4)</sup>を使用しており、この辞書では各形態素に左属性と右属性が与えられている。ここでは次の2つの方法で精度を測定した。

1. 属性精度……左属性と右属性が完全に一致した割合 (ただし、サ変名詞と普通名詞は区別していない)
2. 句切り精度……形態素の区切りが一致した割合

2を測定したのは、EDRの属性分類が非常に細かく、また一部に意味を意識した属性があるため、属性が完全に一致しなくても、情報検索のための形態素解析として十分な場合があるためである。

精度の測定のためEDRの日本語コーパスから任意に1000文を抽出し解析した結果、属性精度で93.5%(形態素単位)、区切り精度で84.7%(文単位)を得た。これらの精度はコスト調整により現在も向上中であり、また属性分類の変更まで含めた改善策も検討中である。

## 5 おわりに

情報検索サービスに用いるための形態素解析とその高速化方法について述べた。現時点での解析速度は、高速ワークステーション上で300Kバイト/秒前後である。これは約1ギガバイト/時間に相当し、当初の目標であるギガバイト単位のデータを実用時間内で処理するという目標は達成した。

さらなる速度の向上のためには、属性分類や活用の扱いといった部分が最も重要である。例えば、EDR日本語単語辞書では、他の辞書では一つの形態素として扱っているものが、意味を考慮して複数に分類されていたり、活用語尾を一つの形態素として扱っていたりしている。このため形態素候補の数が増加し、コスト比較の手間が速度低下をもたらしている。これら、辞書に関する部分は時間と労力を要する仕事であり改良の難しい点もある。しかし、形態素解析を大量文書に適用する重要性は今後ますます高まると思われ、解析速度も考慮して形態素を分類した辞書の構築も求められる。

日頃から貴重なアドバイスを頂いているNTT基礎研究所対話研究グループの皆様、石井健一郎情報科学部長に深く感謝します。

### 参考文献

- [1] 久保, 新田: 接続コスト最小法による形態素解析の提案と計算量の評価について, 信学技法, NLC90-8, pp.17-24, 1990.
- [2] 田中: 自然言語解析の基礎, 第3章形態素解析, pp.133-153, 産業図書, 1989.
- [3] 青江: トライとその応用, 情報処理, Vol.34, No.2, pp.224-251, 1993.
- [4] EDR 電子化辞書 1.5 版仕様説明書, 日本電子化辞書研究所, 1996.

# インターネットにおけるトラフィック分析

小松原 重之 鈴木 亮一 三上 博英

comacie@slab.ntt.co.jp, ryoichi@slab.ntt.co.jp, mikami@slab.ntt.co.jp

日本電信電話株式会社 ソフトウェア研究所 広域コンピューティング研究部

## 1. はじめに

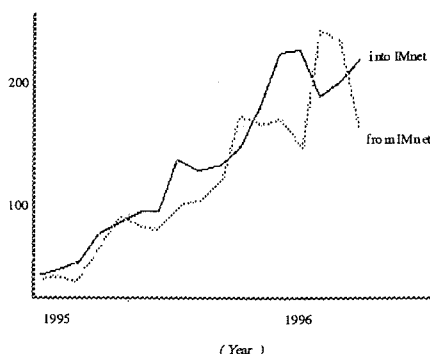
インターネットのトラフィックは、多様なユーザによる利用と、その形態の急激な変化のために複雑な振舞いをする。本研究では、その挙動をモデル化し、インターネットにおけるトラフィック予測技術を確立することを目指している。なお実装と評価に当たっては省際研究情報ネットワーク (Inter Ministry research information Network; 以下 IMnet) のトラフィックデータを用いた。

## 2. 評価の期間と変動

同じトラフィックデータであっても評価期間によって、振舞いは異なって見える。以下では年間変動を長期的、月間変動を中期的、それより短い変動を短期的なものとして定義し、それぞれの評価期間における変動の特徴について述べる。

### 2.1 長期的な変動の特徴

図 1 は IMnet と、他の NSP (Network Service Provider) の間の、トラフィック推移である。



【図 1; IMnet における長期的変動】

出方向入方向ともに、1 年間で 10 倍以上に増加したことが分かる(この間に IMnet に接続する組織の

The Analysis of the Internet Traffic

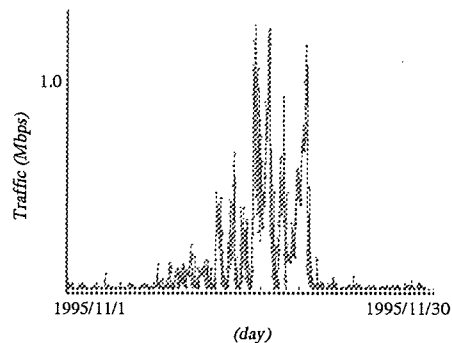
Shigeyuki KOMATSUBARA, Ryoichi SUZUKI and Hirohide MIKAMI

NTT Software Laboratories

数は約 2 倍に増加)。年間のトラフィック変動は、ユーザ数の増加よりも、各ユーザの利用形態の変化による影響の方が大きいことが、推察できる。

### 2.2 中期的な変動の特徴

図 2 は IMnet の東京大阪間リンクでの、APEC 大阪国際会議(11 月 14~21 日開催)の影響である。

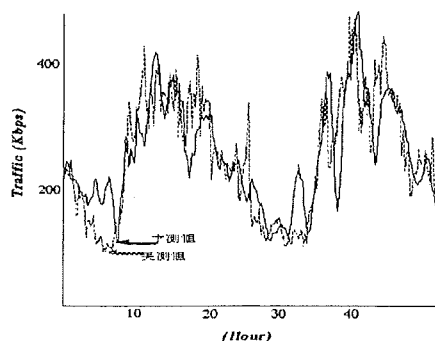


【図 2; 中期的な変動の例】

このようなイベントに伴うトラフィックの変動は、人気ブラウザの配信や、ネットワークを介した研究実験などにおいても見ることができる。ただしこれらの変動は、再現性が低い。

### 2.3 短期的な変動の特徴

図 3 中の実測値は、IMnet のバックボーンのある 2 日間のトラフィックの推移である。

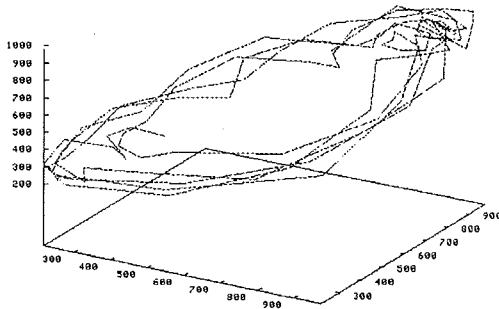


【図 3; 短期的な変動の例と予測値】

サンプリング・レートは10分間隔であるが、微視的な激しい振動と、低周期の増減について再現性が見られる。この性質に着目したトラフィックの解析を、次の手法により行った。

### 3. カオス的手法による解析<sup>[1]</sup>

図4は、Takensの理論<sup>[2]</sup>に基づきIMnetのトラフィックデータから作成した、多次元埋込みアトラクタである(解析次元は3次元)。週日の5日分の短期的なデータによって表現されている。



【図4:IMnetトラフィックのアトラクタ】

滑らかな円弧となっている部分では、トラフィックが推移的に振舞うことを示し、右上部の渦状の部分には、昼間の混雑時におけるカオス的な振舞いを示している。これらの性質を利用し、単位時間後のトラフィックを予測するシステムを作成した<sup>[3]</sup>。図3中の予測値データは、本システムにより得た予測結果である。10分間隔の予測試行で、平均誤差率12.5%の予測精度を得ている。

## 4. トラフィック予測システムの実装

実行環境(Sun Sparc5, メインメモリ128MB)において、単位時間以内に、単位時間後のトラフィックを計算するプログラムを作成した。

### 4.1 リアルタイム性について

本システムは計算に際し、10分間隔のトラフィックデータを格納したデータベースを探索し、中間ファイルをメモリ上に作成する。ここで、データベースアクセスは1エントリあたり10.9msec程度であった。このため探索範囲を8,000時間以上とすると、中間ファイル生成が10分間で行えず、リアルタイム処理が成立しないことが分かった。また、中間ファイルを使う最も大きな処理である解析次元の変更を行っても、処理時間の増加は見られなかった。これらのことから、中間ファイルの生成が処理を律速していることが分かる。現在、その部分の高速化を検討している。

## 4.2 アルゴリズム

本システムの予測アルゴリズムは、軌跡上の各点について次の計算を行う。

1. 設定された探索範囲の中から、現在の最近傍値を求める
  2. 最近傍値の単位時間後までの推移を求める
  3. 現在値と、上項の推移から予測値を求める
- この処理を繰り返し、逐次的に予測を行う。また、実装にあたっては次の機能を加えている。

1. 週日のデータと週末のデータを分離、独立して計算する機能
  2. データベースの各エントリに対して、あらかじめ時間帯によるグルーピングを行う機能
- 前者は、振舞いの異なるデータを分離し、別々に計算を行うことが目的である。これにより計算量の増加はあるが、3%程度の予測精度向上を実現している。

後者は最近傍値の算出の際に、データベースの探索量を減らすことが目的である。タイムスタンプなどの情報に基づいてインデックスされた集合から探索を行うことで、計算量の削減を実現する。この機能により、データベース処理の簡略化(最大アクセス時の1/144回へ削減)を実現し、予測精度に大きな影響がないことを確認した(予測誤差上昇1%未満)。

## 5. まとめ

インターネットトラフィックの、長期的、中期的、短期的な変動の特徴について述べた。また、カオス的手法によりアトラクタを作成、その妥当性についてIMnetのトラフィックデータを用いて、予測可能性を示すことで検証した。予測にあたり、データベース処理が律速プロセスとなること、アダプティブな実装によってそれに対応出来ることを示した。

今後は中間ファイル生成ルーチン的高速化を行い、実行環境の強化を図る。また、計算コストと性能(誤差率)の2つの観点から、最適な実装について考察を進める予定である。

## 【参考文献】

1. 中川善富,塚本健一,山田雅之,世木博久,伊藤英則,"カオス的手法を用いた交通量解析",第52回全国大会1Z-2,May1996.
2. F.Takens,"Detecting Strange Attractors in Turbulence, In Dynamical Systems and Turbulence"(eds.D.A.Rand and L.S. Young),Springer-VerlagBerlin,1981.
3. 小松原重之,鈴木亮一,三上博英,"IMnetにおけるトラフィックオペレーションについて",情報研報Vol.96,No.111(ISSN0919-6072)96-DSM-4,1996.

## 3J-01 ソフトウェア性能評価のためのランダムデータ 高速生成法

二村良彦 大谷啓記 青木健一  
早稲田大学 理工学部

### 1. はじめに

アルゴリズムの性能の精密な評価を行うためには、対象となるアルゴリズムの性能に影響を及ぼす特性を制御して生成されたランダムデータを用いる必要がある。また、性能評価に要する時間を短くするためには、ランダムデータを高速に生成する必要がある。特にアルゴリズムの教育において、学生に作成させたアルゴリズムを正当に評価するためには、合理的な評価用データが不可欠である。しかし、所望の特性を有するランダムデータの高速生成は、アルゴリズムの専門家にとっても容易ではない。例えば、長さ  $n$  の順列を等確率、即ち  $1/n!$  で生成する  $O(n)$  アルゴリズムは知られている[7]。しかし、整列法等の精密な評価のためには、このような一様乱順列による評価では不十分である。何故ならば、アルゴリズムの性能に影響を与える要因はデータのサイズだけではなく、データが有する構造もアルゴリズムの性能を左右するからである。例えば、一様乱順列および一様乱数列は、極めて偏った構造を持つ。我々は先に各種の性質を有する乱順列を計算機オーバーフロー(またはアンダーフロー)無しで高速に生成する方法を提案した[3]。そこでは順列の含む葉数(数列において自分より小さい隣人を持たない要素の個数)を制御して順列を  $O(n)$  で生成する実用的近似方式についても報告した。そこで述べた精度の高いランダムデータ生成法は、高性能確率的アルゴリズム[5]の設計にも利用可能であった。そこでの成果に基づき、我々は、アルゴリズムの研究および教育に関る者が、各種のランダムデータをインターネット(<http://www.futamura.info.waseda.ac.jp/index-j.html>)を通じて容易に入手可能にするために、ランダムデータ

Fast Random Data Generation for Software Evaluation,  
Yoshihiko Futamura, Hirofusa Ohtani and Kenichi Aoki,  
School of Science and Engineering, Waseda University

サーバーを開発した。本稿は、ランダムデータ生成法、ランダムデータサーバー(RDS)およびランダムデータサーバージェネレータ(RDGG)の概要について報告する。

### 2. 一様乱数列の問題点

アルゴリズムの性能評価の際には一様乱数列(又は乱順列)が利用される場合が多い。しかし、それ等は非常に偏った性質を有する。例えばその葉数は殆どの場合約  $(n+1)/3$  である[3]。またランズ(数列における上昇列の個数)の平均はその性質より明らかに約  $(n+1)/2$  である。従って例えば一様乱数列を用いて整列法の評価を行うと QUICKSORT[6]のように葉数が多い場合に速く、少ない場合に遅い方法に有利になる。一方、MERGESORT[7]や LOAS[4]のように葉数が少ない場合に速く、多い場合に遅い方法にとって一様乱数列による評価は不利である[4]。実際に整列が行われる問題領域に適合した葉数を持った数列を用いて評価をしないと、実用的評価とは言い難い。実際に整列の対象となる大規模データの葉数を我々は計測していないが、それは[7]にある通り、整列済みに近いと予想される。整列済みに近いデータの葉数は 1 に近く、QUICKSORT が 1 番得意とする領域である。従って、アルゴリズムの性能を精密に評価するためには、アルゴリズムが実際に扱うデータ領域を持つ特性を制御しながら乱数列を生成する必要がある。

特殊な特性を持つ乱順列を生成する試みはいくつか報告されているが[1, 2]、それ等は一様性の保証をしていない。例えば[2]では、最長上昇部分列(LUSSと略記する)の長さを指定して乱順列を生成している。しかしそこでの方法は、複数のLUSSを持つ順列の方が、LUSSを1つしか持たない順列よりも出現確率が高い。例えば長さ6かつLUSS長を3とした場合、順列

4, 1, 5, 2, 6, 3は, 順列6, 5, 4, 1, 2, 3の4倍以上の確率で生成される. また[1]では葉数を指定し乱順列を生成しているが,  $\chi$  自乗検定での結果が思わしくない. また2分木を等確率で生成する方法も報告されているが, これは特性を制御した一般の順列の生成には直接応用できない[3].

### 3. 各種乱順列の生成法

我々は[3]において, 長さ $n$ , 単純指標 $m$ を持つ乱順列を $O(nm)$ で生成する計算機オーバーフロー(またはアンダーフロー)無しの方法および, 単純指標が葉数である場合には順列を $O(n)$ で生成する実用的近似方式について報告した. 単純指標とは, 数列の葉数, ランズ, 上昇部分数等に対応する特性のクラスである. そして形式的には順列がその生成規則に従って1つ短い順列から生成される際に, 次の2性質を有する順列の特性のことである: (1) 特性指標(順列から非負の整数上への関数及び関数値)が高々1しか増加しない. (2) 新たな要素の挿入個所を順列の長さ $n$ と特性指標に基づいて決められる.

単純指標を持つ順列の総数は下記の特性方程式により表すことが出来る[3]:

$$S(n, m) = x(n, m)S(n-1, m) + y(n, m)S(n-1, m-1).$$

また, 次の確率関数 $P(n, m)$ を分母と分子のオーバーフローを起こさずに計算することにより長さ $n$ 単純指標 $m$ の乱順列を $O(nm)$ 時間およびスペースで生成することが出来る:  $P(n, m) = x(n, m)S(n-1, m) / S(n, m)$ . しかも特性指標が葉数の場合には確率関数 $P$ の簡単な近似式を見付けることができた[3]. 近似方式と従来の方法との比較を通じて葉数を指定して長さ100000以上もの大きな順列を生成する場合には, 近似方式を利用せざるを得ないという結論を我々は得ている[3].

### 4. ランダムデータサーバー

たとえ文献[3]を読んでも, その内容を理解し所望の特性を有するランダムデータを生成するプログラムを開発することは容易ではない. 特にデータを高速に生成するためには, プログラミング上のテクニカルノウハウが必要である. また, ランダムデータ

の生成方式は特許出願されている(特願平8-250030, 1996年9月)ので, そのプログラムを自由に配布することには不都合がある. そこで我々は, ジェネレータ自身は配布しないことにした. その代わりランダムデータの必要な研究者や学生に対し, 所望のデータを供給するランダムデータサーバーを開発した. それはこの後の講演3J-02と3J-03で報告する通り, 指定された特性指標と長さを持つ乱順列を, 指定された個数インターネットを通して供給することが可能である.

### 5. おわりに

アルゴリズムの研究と教育のために利用されることを期待して, ランダムデータの生成法と, それに基づくランダムデータサーバーを開発した. そしてそれをインターネット上で誰にでも利用できるようにした. RDSのようにテクニカルノウハウと知的所有権(特許)の凝縮されたソフトウェアを, それ自身を配布せず, その機能だけを社会にサービスする機会は今後ますます増大すると思われる.

### 6. 参考文献

- [1] 浅野: 各種ソーティングアルゴリズムの実際的评价, 情報処理学会アルゴリズム研究会30-7, 92年11月.
- [2] Cook, C.R. and Kim, D. J.: Best sorting algorithm for nearly sorted lists, CACM, Vol. 23, No. 11, 1980, pp. 620-624.
- [3] 二村, 大谷, 青木, 二村: 単純指標を持つ乱順列の高速生成法, 情報処理学会アルゴリズム研究会, 97年1月.
- [4] 二村, 二村, 遠藤, 平井: 葉数最適整列法LOASとその実現法, 情報処理学会アルゴリズム研究会44-2, 95年3月.
- [5] R. Gupta et al: On Randomization in Sequential and Distributed Algorithms, ACM Computing Surveys, Vol. 26, No. 1, March 1994
- [6] Hoare: Quicksort *Comput. J.*, 1, 1, 1962, 10-15.
- [7] Knuth, D.: *The Art of Computer Programming*, Vol. 1-3, Addison-Wesley, 1973.

## 3K-02 データフロー解析に基づくプログラム保守支援

四野見 秀明 藤井 邦和 高橋 真由美

日本アイ・ビー・エム株式会社 アジア・パシフィック・プロダクツ

### 1 はじめに

筆者らは、PL/I、COBOL対象の保守支援ツールRE/Cycleの研究開発を行って来た[1][2]。以前の報告後、対象解析範囲を1コンパイル単位内から、プログラム群へと拡張を行い、外部ルーチン呼び出しを経由する他プログラムへの影響波及解析をも可能にし、名前も'悟' Satoru (Source code Analysis Tool for pRogram Understanding) とした。本稿では、'悟'の機能について、保守支援におけるデータフロー解析結果の有用性という観点で報告する。技術的には、プログラム・スライシング[3]の実用レベルでの応用である。これらの機能は、西暦2000年対応のプログラム修正時にも強力な支援となる。

### 2 保守時のプログラム理解

プログラム、特に第三者の書いたプログラムを保守する場合、まずその内容を正確に理解しなければならない。ところが、現実の保守現場では、対象となるプログラムに関する仕様書が無くなっていたり、元々存在していなかったり、あってもプログラムに対する長年の修正の繰り返しが仕様書に反映することを怠ったため、仕様書に書いてある内容と実際のプログラムの実装が異なっていることも多い。その時、担当者が唯一頼りになるプログラムを詳細に読んで、その内容を理解しなければならない。

我々はRE/Cycle[1]をプロトタイプとして、社内外の保守現場との共同研究を行ってきた。それを通して、既存のPL/I、COBOLのプログラムの保守を困難にしているのは、GOTOの多用やルーチンの呼び出し関係の複雑さ等の制御の追跡の難しさもあるが、データがどのように利用され、流れていくかを追跡することの難しさが大きいことが明らかになった。

保守時の多くの部分を占めている作業は、ある部分を修正する場合の影響箇所を発見し、作業の見積りをしたり、実際の修正作業をすることである。プ

ログラム中の修正による影響が、データの流れていくに沿って代入文やサブルーチン呼び出しにより、プログラム内の他の箇所もしくは、他のプログラムへと波及していくのを見落とさず正確に把握する必要がある。また、プログラムを理解しようとする場合、注目する入力データが、プログラムの中でどのように受け渡されているかを追跡して行く必要がある。データの流れていくを追跡することが、正確なプログラム理解へと繋がるのである。

しかし、PL/I、COBOLの保守現場でのプログラムは、データの受渡しが明示的な代入であることは少ない。以下のようなデータの階層関係やメモリー上での配置を考慮しながら把握する必要がある：

- 同じメモリー領域を別のデータ構造として共用する複雑な再定義(COBOLのREDEFINES、PL/IのDEFINED、BASEDの利用)
- データ構造単位での代入や、COBOLのCORRESPONDING、PL/IのBY NAMEによる代入
- コンパイラが行うデータの境界合わせ

'悟'では、以上の点をツールが考慮し見落しの無い正確な影響波及分析、プログラム理解を実現する。

### 3 データ依存関係解析

'悟'の解析は、プログラム・スライシング技術[3]の静的スライスにおけるデータ依存関係(Data Dependence)の利用に相当する。プログラム・スライスには、プログラムを静的に解析して得られる静的スライス(Static Slice)と、ある入力に基づいてプログラムを実行したパスに関して得られる動的スライス(Dynamic Slice)がある。動的スライスとは実行時に動的に決定されることに関する解析も可能な反面、プログラム中に実行したパス上のことしか解析できない。静的スライスは、(入力データにより、ポインタが指す位置が移動するなど)実行時にしか決定されないことに関しては解析できないが、実行されなかったパスをも含む全てのパスに関してプログラムを解析できるという利点がある。プログラムのある箇所を修正した場合の影響波及分析という観点では、静的スライスが適している。ちなみに、COBOLではポインタは使用されることは無く、PL/I

Program Maintenance Support Based on Data Flow Analysis

Hideaki Shinomi, Kunikazu Fujii,

Mayumi Takahashi

Yamato Laboratory, IBM Japan, Ltd.



でも殆どの場合、ポインタを動的に移動することは無く、主に同じメモリー領域を異なる名前でも共有するために(DEFINEDの代わりに)使用される。

'悟'では、プログラムの構文解析後に、ステートメント間の制御を表現するコントロール・フロー・グラフを内部的に生成し、その上でステートメント上の変数間のデータ依存関係を生成する。その依存関係を利用して次節で述べる機能を実現する。

#### 4 データフロー解析による保守支援

ここでは、'悟'の機能の中で、データ依存関係を辿ることにより実現している機能について述べる。

##### 4.1 プロセス間のデータフロー表示

'悟'では、プログラム間の呼び出し関係や、プログラム内での内部ルーチンの呼び出し関係を表す図を自動生成する(図1)。その呼び出し関係を表現する矢印の横にのみの印としてプロセス間を流れるデータを表現する(データカップル)。呼び出し関係の矢印を選択することにより、呼び出しにより渡される/返される変数名がリストされる。データ構造中の下位構造のみが渡される時は、その修飾された下位構造名がリストされる。呼び出しによる入出力インターフェースが明確になる。リストされるのは、呼び出しパラメータにより渡されるものだけではなく、(EXTERNAL属性を含む)グローバル変数を経由して渡される/返されるデータも含む。

##### 4.2 影響波及分析

データ依存関係を辿ることで、ステートメントや、データ構造中のフィールドの意味を変えた時の影響箇所を把握できる。'悟'では、ステートメント中の変数を選択し、その値が代入/参照する箇所(代入/参照検索)、また、データの流れの追跡による、その値に影響を与える/が影響を与える全ての箇所(代入/参照連鎖検索)を画面上で表示できる。前者はデータの流を追跡しながらのプログラム理解に有効であり、後者はプログラム修正時の影響波及の把握に有効である。既に述べたデータの再定義やデータ階層を考慮した正確で見落としのない解析を実現する。図1はファイルから読み込むレコードの1フィールドに関して意味的変更を行う場合の参照連鎖検索の結果である。影響波及するステートメント、内部ルーチン、プログラムが明示されている。

#### 5 西暦2000年問題への適用

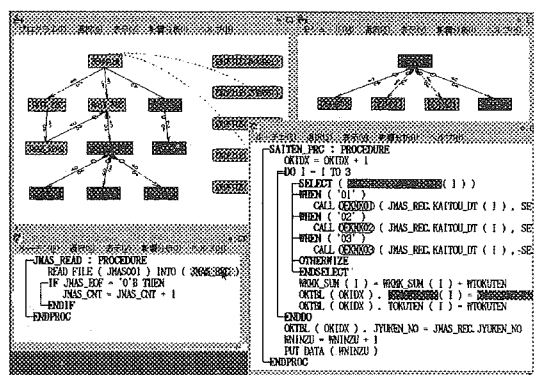


図1 影響波及分析結果表示(参照連鎖検索)

年数を表現する際に西暦の下2桁のみを使用しているプログラムが、西暦2000年を迎えて修正を余儀無くされている。'悟'は、その修正を見落としなく、効率良く行う手段を提供する。データカップルを参照することで、プログラム、ルーチンの呼び出しインターフェースの理解をすることが出来、仕様書のないプログラムに関しても、影響波及分析を起動すべき変数の決定を支援する。修正が必要とされる年数の変数から影響波及分析を起動することにより、影響が及ぶ箇所を正確かつ見落としなく発見することが可能である。

#### 6 おわりに

'悟'におけるデータフロー解析の結果として表示される機能と保守におけるその意義を述べた。その他、ソースコード表示のアクション・ダイアグラムや、データ構造のメモリー上での配置や、階層関係を視覚化したデータ構造図等の機能については別稿で述べる。また、'悟'の静的スライス解析結果としてのデータ依存関係と制御依存関係の抽出機能は、ある会社で、システム再構築のための既存プログラム理解に実際に使用されている。

#### 参考文献

- [1] 四野見秀明, 藤井邦和, 牧野正士, 津田和幸, "構造化分析/設計の方法論に基づいたプログラム理解支援ツール", ソフトウェア工学研究会資料 SE-87-1, 情報処理学会, 1992.
- [2] 四野見秀明, 藤井邦和, "プログラムの自動的な再構造化における限界とその解決法", 情報処理学会情報システム研究会 42-1, (1993).
- [3] 下村 隆夫, "Program Slicing技術とテスト, デバッグ, 保守への応用", 情報処理, Vol. 33, No. 9, pp. 1078-1086.

## トランザクション処理によるディスクアクセスの トレースを用いた Hot mirroring の性能評価

茂木 和彦 喜連川 優  
東京大学 生産技術研究所

### 1 はじめに

2次記憶装置の高性能化・高信頼化を目的とした冗長情報を記録するディスクアレイ (RAID)[1] の開発が進められている。その中で、サイズは小さいが多数のアクセス要求があるような負荷では、ミラーや RAID5 が良いと考えられている。RAID5 ではパリティを用いた冗長化を行っており、データ書き込み時のパリティ更新のためのオーバーヘッドやディスク故障時のデータ復旧作業の影響による性能の低下が問題となっている。この点に関して優れているミラーでは、データのコピーを保持することによる冗長化を行っており、データ容量が少ないという問題点が存在する。これらの問題を解決するために「Hot mirroring」と名付けた記憶管理法を提案した [2]。本方式は、参照局所性を利用したミラーと RAID5 の階層構成により高性能性と高記憶効率性の両立を目指すものである。本手法について、より現実的な負荷での性能評価を行うため、TPC-C ベンチマーク [3] を基にしたトランザクション処理を実行した時のディスクアクセスのトレースを採取した。本稿では、このアクセス負荷に対する Hot mirroring を用いたディスクアレイの性能を評価した結果について述べる。

### 2 トランザクション処理によるディスクアクセスの特徴

RAID の用途として重要なものの一つにトランザクション処理システムを挙げることができる。そこで、TPC-C ベンチマークを基に商用ミドルウェアを用いて SPARCstation 20/502 (OS: Solaris2.3) 上にトランザクション処理環境を構築し、データ領域に対するディスクアクセスのトレースを、修正を加えた sd ドライバを用いて収集した。データベースのテーブルの構成を表 1 に示す。これら 9 つのテーブルと 2 つのノンクラスターインデックスをそれぞれ分離・独立した領域に記録する。データベースの規模は 14 ウェアハウスとし、テーブルとインデックスの記憶に総計 11,420MB を割り当てた。システム上のデータ用バッファキャッシュは 56MB とした。チェックポイントは実行されない。

100 万回のアクセス (読み出し 685,296 回、書き込み 314,704 回) におけるアクセス位置の分布の図 1 に示す。アクセス位置に関しては、アクセス頻度が高い部分と

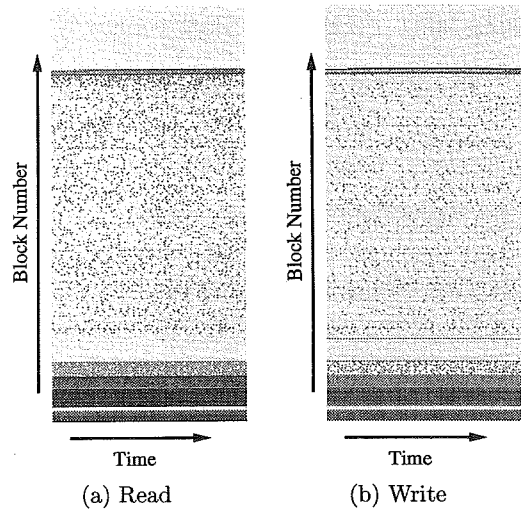


図 1: アクセス位置の分布

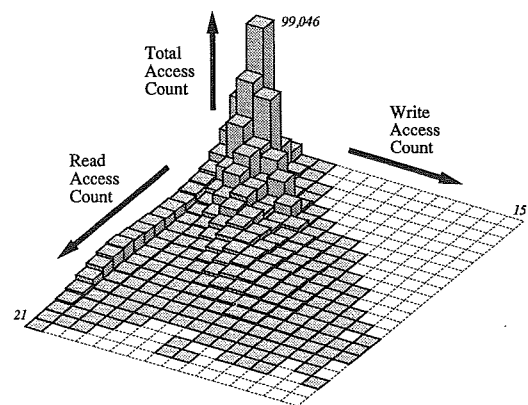


図 2: ブロックのアクセス頻度の分布

アクセス頻度が低い部分へとかなり明確に分離されることがわかる。また、このときの読み出し / 書き込み回数によりブロックを分類した時のアクセス頻度の分布を図 2 に示す。読み書き双方 1 回のみブロックに対するアクセスが一番多いものの、数回程度の読み書きが行われているブロックに対するアクセスも多数存在する。リード・モディファイ・ライトが行われているものが多い。問い合わせには最新のデータに関する (時間的局所性をもつ) ものがあり、図には明示されないものの、書き込み直後に読み出しが実行されるブロックが存在する。

Table name	Clustered index	Non clustered index
Warehouse	W_ID	—
District	D_ID, D_W_ID	—
Customer	C_W_ID, C_D_ID, C_LAST	C_W_ID, C_D_ID, C_ID
History	—	—
New-Order	NO_W_ID, NO_D_ID, NO_O_ID	—
Order	O_W_ID, O_D_ID, O_ID	O_C_ID, O_D_ID, O_W_ID, O_ID
Order-Line	OL_W_ID, OL_D_ID, OL_O_ID, OL_NUMBER	—
Item	I_ID	—
Stock	S_I_ID, S_W_ID	—

表 1: テーブルの構成

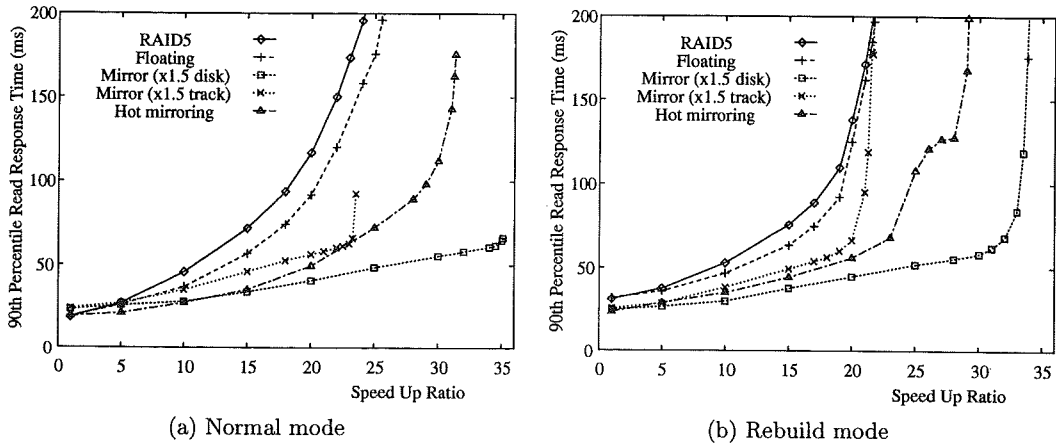


図 3: アクセストレースを用いた性能評価

### 3 アクセストレースを用いた性能評価

アクセストレースを用いた Hot mirroring の性能評価を行った。5 台のデータディスクに対して 1 台のパーティションを持つグループが 3 つある構成を仮定する。ディスク容量は 850MB とし、各ディスクの 10% をミラー ホット領域に割り当て、RAID5 コールド領域のキャッシュ的な動作をさせる。この構成においては、全ディスク容量の 75% のデータが記録可能であり、ホット領域に 6.6% のデータブロックを記録可能である。この構成におけるトレースデータを用いて評価した性能を図 3 に示す。図の横軸はトレースデータに基づく到着シーケンスの加速率を、つまり、加速率  $x$  は到着時間間隔を  $1/x$  にすることを意味する。縦軸は、(a) 50 万アクセス中、(b) 復旧開始から終了までの復旧動作中の 90% 読み出しレスポンスタイムを示す。比較のため RAID5、フローティング、ミラーの性能も示した。データ容量を等しくするため、ミラーではディスク台数を 1.5 倍にしたものと、シリンダ内のトラック数を 1.5 倍にしたものを載せた。RAID5 では 0.15%、その他では 0.1% の容量の不揮発性書き込みバッファが存在すると仮定した。通常動作時で  $\times 1.5$  トラックのミラーの性能が低めなのは、本評価でのアクセスパターンはある狭い領域に集中しているために平均シーク時間が短い、ミラーでは書き込み時にロングシークを必要とするためである。復旧動作時に Hot mirroring の性能が単調に悪化しないのは、書き込みバッファと空き領域作成動作により書き込み時の動作状態が幾つかの状態に分け

られるためである。Hot mirroring は RAID5 とフローティングより高い性能を示す。また、ディスク台数を増やしたミラーよりは性能が低いものの、そのコストを考えると Hot mirroring は十分に良い性能を出しているといえることができる。

### 4 まとめ

TPC-C ベンチマークを基にしたトランザクション処理の実行時のテーブル記憶領域へのディスクアクセスのトレースを採取し、そのアクセスの特徴を調べた。このトレースを用いて Hot mirroring の性能評価を行った。Hot mirroring は同様な構成の RAID5 やフローティングよりも高い性能を示す。ディスク台数を増やした同容量のミラーよりは性能は低いものの、システムのコストも考えると Hot mirroring が最も良い構成であると言えるであろう。

### 参考文献

- [1] D. A. Patterson, G. A. Gibson, and R. H. Katz. "A Case for Redundant Arrays of Inexpensive Disks (RAID)." In *Proc. of ACM SIGMOD Conf.*, pp. 109-116, June 1988.
- [2] K. Mogi and M. Kitsuregawa. "Hot mirroring: A method of hiding parity update penalty and degradation during rebuilds for RAID5." In *Proc. of ACM SIGMOD Conf.*, pp. 183-194, June 1996.
- [3] Transaction Processing Performance Council (TPC). TPC BENCHMARK<sup>(TM)</sup> C Standard Specification, Revision 3.1, June 1996.

### 3T-3 モバイルグループウェアシステム「なかよし」の構想

倉島 顕尚 市村 重博 田頭 繁 前野 和俊

NEC C&C 研究所

#### 1 はじめに

我々は、携帯端末のモビリティを活かした協同作業システムの新しい応用を開拓する目的でモバイルグループウェアシステムの研究を推進している [1]。携帯端末を協同作業システムの端末として利用できる場面は、いくつも考えられるが、モバイルグループウェアシステム「なかよし」では、複数の端末が互いに電波が届くような近い範囲に集まったときに、一時的にネットワークを構成して協同作業環境を提供するものである。本発表では、この「なかよし」システムのコンセプトを説明し、PHS子機間パケット通信を利用して開発したシステムについて述べる。

#### 2 システムの利用形態

「なかよし」システムは、互いの端末からの電波が直接届く範囲内の端末間での協同作業を対象としている。例えば、次のような利用形態が考えられる。

- 教育の場。先生と生徒が教室内で各自の端末を使用し、教材の配布や、それに沿った学習の実施。
- 講演会場。会場に設置された端末を用いて講演者がプレゼンテーションを行う。参加者が持ってきた端末上にも資料が配布され、プレゼンテーションの進行にしたがって、表示が切り替わる。
- ビジネスの場。会議に参加して互いに資料を配布したり、端末間で名刺交換を行ったりする。また、この履歴を蓄積して、再利用する。

本システムは、以上の例に示すように、人と人が出会う場での協同作業の支援を行うものである。人の出会いは、家庭、職場、学校、外出先などさまざまな場所で発生するので、利用の柔軟性の面からは、特定の環境に依存しないシステム構成であることが望まれる。

#### 3 要求仕様

利用イメージから、以下の前提条件が導き出せる。

- ユーザが利用する端末以外、サーバマシンや基地局など特別な設備を必要とせずに動作すること。

- ユーザの利便性を第一とし、簡単な操作インタフェースで利用できること。

第一条件は、図1のように移動端末が集めた場でシステムが動作することを保証するためのものである。したがって、図2のように固定端末を含めた構成にしても良い。有線系のバックボーンに接続された固定端末を含めれば、その端末を介してインターネット/イントラネット上のコンテンツへのアクセスも可能となり、システムの利用価値が向上する。

これらの前提条件に加え、協同作業を支援するためのアプリケーションはTCP,UDP/IPを利用することとし、システムを構成する端末に要求される機能を検討した結果得られた要求仕様を以下に列挙する。

1. 挟帯域の通信路を利用した場合でも動作すること。
2. 電波到達範囲内の端末間でIPが利用できること。
3. 第2に関連し、端末がある作業グループから別のグループに移動する毎の、利用者によるIPアドレスやネットワークマスクなどのネットワーク設定の変更が不要なこと。
4. 作業グループの情報は端末が取得し、利用者がグループを選択するだけで参加できること。

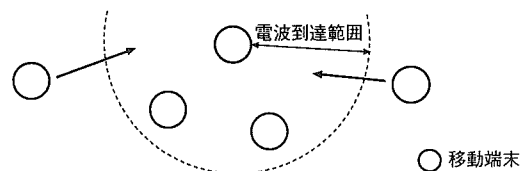


図1: 移動端末だけによる構成

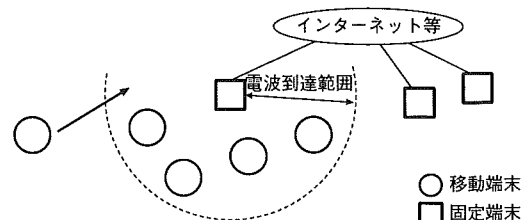


図2: 固定端末を含む構成

Concept and Design of a Mobile Groupware System "NAKAYOSHI"

Akihisa Kurashima (kura@mmp.ci.nec.co.jp), Shigehiro Ichimura, Shigeru Tagashira, Kazutoshi Maeno  
C&C Research Laboratories, NEC Corporation

## 4 システムの設計指針

要求条件に従い、システムの設計指針を定めた。

**IP マルチキャスト通信の利用** 要求条件1に対し、IP マルチキャスト通信を積極的に利用した構成とした。協同作業システムでの IP マルチキャスト通信の利用に関する検討は [2] で行っているが、通信の信頼性に欠けるため、参加退席や操作権制御など、グループの管理に関係し、トラフィックは少ないが、信頼性の要求される部分では TCP/IP のユニキャスト通信を使用し、アプリケーション内でのテレポインタの表示など、トラフィックが多く、信頼性が要求されない部分では、UDP/IP のマルチキャスト通信を使用するとした。

**アドホックネットワークの構築** 要求条件2,3に対しては、協同作業に際して、端末が一時的なネットワークを構成することとした。この一時的なネットワークをアドホックネットワークと呼ぶ [3]。「なかよし」システムでは、アプリケーションは TCP,UDP/IP を利用することを前提としているので、アドホックネットワーク上で IP が動作することが必須である。なお、アドホックネットワークの構築に際してのユーザからの操作は、最小限に抑えるよう工夫する。

**グループ情報の分散管理** 移動端末のみでシステムを構成することがあるため、特定のマシンで集中してグループ情報を管理できない。そのため、開催されているグループの情報は、各端末間で互いに交換し、各端末が全ての開催されているグループの情報を保持する [4]。この機構は、要求条件4を満たす。

## 5 システムの実装

本システムは、無線通信の利用を想定している。無線通信の方式は複数あるが、PHS 子機間パケット通信 [5] を利用したシステムを開発したので報告する。

### 5.1 端末構成

サブノートサイズのノート PC に試作 PHS カードを挿入した端末を複数セット用意し、実験環境とした。PC の OS には Windows95 を採用した。

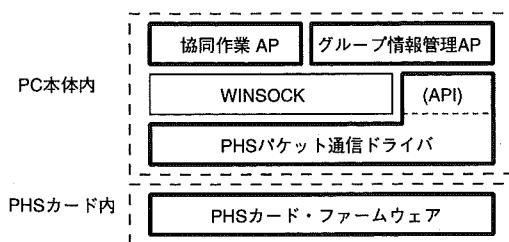


図3: 端末内のソフトウェア構成

### 5.2 ソフトウェア構成

端末内のソフトウェアの構成は図3に示す通りであるが、このうち、太線で示した部分を開発した。

**協同作業 AP** グループ内で動作する協同作業用のアプリケーション。通信インフラに依存しないで動作するように、TCP,UDP/IP を基盤とした。具体的には、OS 提供の WINSOCK ライブラリを使用し、プレゼンテーション AP などを作成した。

**グループ情報管理 AP** 開催されているグループの情報を交換するためのアプリケーション [4]。グループで利用される IP マルチキャストアドレスなどの情報を保持している。ユーザがグループを選択すると、この情報をパラメータとして協同作業 AP を起動する。

**PHS パケット通信ドライバ** PHS 子機間パケット通信を行うためのドライバ [3]。WINSOCK のランタイムルーチンから呼び出せるように、パケット通信部分は NDIS3.1 に対応している。また、PHS に関する設定など、WINSOCK から呼び出されない PHS 固有の機能に付いては、別途 API を用意し、アプリケーションに対し開放している。

**PHS カード・ファームウェア** PHS 子機間パケット通信を行う PHS カード内のファームウェア [3]。

### 5.3 実験と評価

以上のように製作したソフトウェアを実験用の端末上で実行し、動作を確認した。グループ内の通信に利用したのは PHS のチャンネル一つである。PHS は一チャンネルあたり 32kbps の伝送速度を持つが、この速度でもデータの同報に IP マルチキャスト通信を使用すれば、テレポインタや手書き程度のデータの転送・表示はストレスなく行えることが確認された。

### 参考文献

- [1] 倉島, 前野, “グループウェア API (GAPI) の提案とその PenMERMAID への応用 ~インプリメンテーション~,” 情処 49 全大 5E-5 (1994).
- [2] 田頭他, “マルチメディア分散在席会議システムにおける IP マルチキャストの適用,” 情処 52 全大 2X-1 (1996).
- [3] 市村他, “モバイルグループウェアシステム「なかよし」におけるアドホックネットワークの構築方法とその実装,” 情処 54 全大 3T-4 (1997).
- [4] 田頭他, “モバイルグループウェアシステム「なかよし」における IP マルチキャスト通信の利用,” 情処 54 全大 3T-5 (1997).
- [5] 武次, “PHS packet ad hoc network における multicast 伝送方式,” 信学技報 PCS96-110 (1996).

## 4 B—4 商用機械翻訳ユーザ辞書の共通フォーマット設定に向けて

亀井 真一郎 \*1 平井 徳行 \*2 斎藤 由香梨 \*3

伊藤 悦雄 \*4 赤羽 美樹子 \*5 檜山 努 \*6 村木 一至 \*1

\*1 NEC \*2 シャープ \*3 富士通研 \*4 東芝 \*5 ノヴァ \*6 NEC 情報システムズ

kamei@hum.cl.nec.co.jp, nnd6@isl.nara.sharp.co.jp, yukari@ling.flab.fujitsu.co.jp,

etsuo@sp.tokyo-sc.toshiba.co.jp, aka@nova.co.jp, {hiyama, muraki}@hum.cl.nec.co.jp

### 1 はじめに

翻訳に使用される辞書データの流通・相互利用を促進するため、アジア太平洋機械翻訳協会 (AAMT) に加盟する機械翻訳 (MT) メーカーが中心となり、各社の MT システムのユーザ辞書と相互変換可能なユーザ辞書共通フォーマット (Universal PlatForm; UPF) の設計を開始した。本稿では活動の目的と方針概要を述べる。

商用の MT システムが製品化されて 10 年余が経過した。当初、MT システムは翻訳の専門家向けに設計・開発され、主に技術翻訳に用いられてきたが、近年のインターネットの発達とパソコンの普及により、MT システムは一般ユーザにも急速に浸透し始めている。MT システムを有効活用するためには、各ユーザ毎に、頻繁に使用する語彙を「ユーザ辞書」として蓄積し、システムの基本辞書と合わせて使う必要がある。しかし辞書作成は一般に時間と労力がかかる仕事であり、個人ユーザー一人が辞書を個々に充実させるには限界がある。

この問題の具体的解決方法として、個人が個別に蓄えている辞書データを流通させ、相互利用するための環境の整備が挙げられる。現在は国内の 20 数社が機械翻訳システムを商品化しているが、それらの機種の違いを越えて、共通にユーザ辞書を交換できる仕組みがあれば、各人がユーザ辞書を作成するコストが大幅に削減できる。このことにより MT の利用が促進され、ひいては日本人の外国語文書受発信が促進される。

このような環境整備の具体的活動として、AAMT では、今年度 (平成 8 年度) から来年度にかけ情報処理振

興事業協会 (IPA) の創造的ソフトウェア育成事業の予算補助を受け、各社の MT システムに共通のユーザ辞書記述フォーマット UPF の開発と、ホームページによる仕様公開の活動を開始した。仕様は検討段階においても適宜一般に公開する方針である。

### 2 UPF 設計の基本方針

異種システム間で辞書データを交換できるようにするため、以下のような環境を開発する方針である。

- (1) 共通フォーマットの設計
- (2) 共通フォーマットと各システムのユーザ辞書の間の双方向コンバータの提供
- (3) 共通フォーマットで記述された辞書を蓄え流通させるための、一般アクセス可能な電子環境の提供

上記 (1) のフォーマット開発は、現実に発売・利用されている複数のシステム間でそのユーザ辞書情報を比較することにより行なうこととした。この方針をとることで、共通フォーマットが現実のシステムから遊離してしまう危険を回避できると考えている。上記 (2) の双方向コンバータは、各 MT メーカーがそれぞれ独自に開発するものである。共通フォーマット設計の際には、共通フォーマットから各システムのユーザ辞書への変換 (ダウンロード) と、各システムのユーザ辞書から共通フォーマットへの変換 (アップロード) との両方が可能となるよう考慮する必要がある。上記 (3) の辞書共有環境としては、AAMT のホームページを想定している。また直接 UPF 形式で辞書を記述するための辞書エディタも提供する。各ユーザは共有環境に置かれている UPF 形式の辞書データを自分の使っている MT システムのユーザ辞書のフォーマットに変換して使用することができる。また逆に各ユーザが自分の使用している MT

Common Formats for User Dictionaries of MT Systems  
Shin-ichiro KAMEI, tokuyuki HIRAI, Yukari SAITOH,  
Etsuo ITOH, Mikiko AKABANE, Tsutomu HIYAMA, Kazunori  
MURAKI

システムで作成したユーザ辞書は、UPF形式に変換してこの辞書共有環境に置き、他ユーザ（異システムユーザも含む）と共有できる。

UPFは、さしあたり日本語と英語の2カ国語を分析対象として具体的設計をすすめるが、その形式は多言語に対応できるものを目指している。また一つの言語の生成と解析の辞書はできる限り統一した形式で記述できるように仕様設計することを目標としている。具体的記述形式はSGMLに準拠したタグを用いる。

### 3 UPF設計の具体的活動

#### 3.1 基本変換標準と拡張変換標準

上述したようにUPFは、UPFから各システムのユーザ辞書への変換（ダウンロード）と各システムのユーザ辞書からUPFへの変換（アップロード）との双方向が可能となるように設計する必要がある。現実利用されている複数のシステムのユーザ辞書で扱える語彙（品詞）には相違があるから、狭い意味で上記の双方向条件を満たすためには、各システムに共通して記述できる語彙（記述可能な語彙の「AND」）だけを対象範囲とする必要がある。一方、そのような「AND」仕様だけでは、詳細・広範な語彙情報の記述を許すシステムが有効活用されないという問題が生じる。そこで上記の両方の要求を満たすため、UPFでは以下の2種の対象範囲を設定することとした。

##### (a) 基本言語変換標準

全MTシステムのユーザ辞書で取り扱うことができ、UPFとの間で相互変換（アップロード・ダウンロード）可能であることを推奨する語彙の情報を記述する形式

##### (b) 拡張言語変換標準

各MTシステムで記述する可能性のあるすべての語彙の情報を記述する形式

すでに商品化されユーザに使用されている5つの異なるシステムのユーザ辞書の比較検討を元にして、現在上記2種の仕様設計作業を進めている。基本的には第1年度に基本言語標準を、第2年度に拡張言語変換標準を設計するが、基本言語標準の設計に際しても拡張言語標準を考慮する必要があるのは言うまでもない。また拡張言語標準の設計段階で再度基本言語標準に戻って改良する必要が生じるかも知れない。

#### 3.2 基本言語変換標準の概要

基本言語標準の設計に際しては、まず各システム共通の「記述用語」の設定作業を行なう必要があった。つまり、品詞のセット、品詞の呼称など用語と定義の統一から作業を開始した。特に日本語の場合、基本となる品詞設定についても学校文法では機械翻訳にとって不十分であり、準拠すべき標準が存在しない。具体例としては「形容動詞」という品詞を独立の品詞として立てているシステムと「形容詞」の下位として扱っているシステムが存在した。またその登録単位も語幹登録、終止形登録の二通りがあった。このような用語・形式の統一を行ない、基本言語変換標準としては、現在のところ以下を対象として原案作りをすすめている。

日本語： 名詞、固有名詞、動詞、サ変動詞、  
形容詞、形容動詞、副詞

英語： 名詞、固有名詞、動詞、形容詞、副詞  
上記の品詞設定は、実際に現在までの10年間にMTユーザが登録した辞書の約9割が名詞、固有名詞であるという各社に共通した経験的データに基づいている。名詞の下位分類とすることも可能な固有名詞を独立の品詞として立てたのは、ユーザが高頻度で登録する可能性のある語群であることがその理由である。固有名詞と同様に、サ変動詞も登録率を考慮して動詞の下位でなく独立品詞とした。このように、基本言語標準の設計に際しては、言語学的に厳密な現象記述よりもデータの流通性に重点を置いた。

### 4 おわりに

本稿では、異システム間でユーザ辞書データを交換・流通させるための共通フォーマット（UPF）開発活動の概要を述べた。UPFとしては、基本標準、拡張標準の2つを開発する方針である。ワーキンググループで原案を作成し、AAMT加盟メンバの承認を経て、MT業界の標準とし、電子ネットワークによってユーザ辞書データを流通させるのが目標である。この活動の第1年度末である現時点では、基本標準の仕様がほぼ固まっている。今後は各システムのユーザ辞書との間の双方向変換の確認を行なうのと並行して、拡張標準の設計を行ない、来年度末に基本・拡張の両仕様をFIXして一般公開する予定である。この活動が、個人のもつノウハウの交換、流通を活性化し、MT技術の普及に貢献し、日本人の外国語情報受発信を促進することを望んでいる。

## 画像データの 캡セル化による版權管理手法

木俣 豊† 田中克己†† 上原邦昭†††

†通信・放送機構 神戸リサーチセンター

††神戸大学大学院自然科学研究科知能科学専攻

†††神戸大学工学部情報知能工学科

### 1 はじめに

インターネットの普及に伴い、様々なマルチメディアデータがネットワーク上で取り扱われるようになってきた。このようなマルチメディアデータは、コンピュータ上のデジタル情報として提供されるために品質を損なうことなくコピーすることが容易で、著作権や版權を保護することが難しい。

画像データの著作権保護の技術としては、画像情報の中に人間には識別できないように著作権情報を付加する電子透かしの技術 [1][2] がある。この手法は著作権の存在をデータとして画像に埋め込むことによって、制作者の權利を保護するものである。その一方で制作者が使用を許諾したもの以外にはそのデータを使用させないといった、より強い管理の手法も求められている。本研究ではマルチメディアデータを 캡セル化して使用することによって得られる機能とそれを応用した版權管理手法の有効性を Java を用いて検証した。

### 2 マルチメディアデータ 캡セル化の概念

ユーザによる二次的な編集を禁止したマルチメディアデータの場合には、そのデータの編集を制限する必要がある。しかし、GIF や MPEG などの一般的なマルチメディアデータは、データ側からアプリケーションの制御が不可能である。そのためにアプリケーションで二次的な加工が行われ制作者の意図が損なわれたり、データの不正使用や不正加工が行われる問題があった。このような問題に対してオブジェクト指向技術は有効な解決手段となる。オブジェクトは 캡セル化されたデータへのアクセスを特定のメソッドだけに制限する事が可能である。従って、内包されたマルチメディアデータは不正なアクセスから保護され、不正な改変を防止する事ができる。また、メソッドに製作者が意図した機能を付加することにより、アプリケーションに依存することなくマルチメディアデータの

Copyright management method with the encapsulation of the image data

Yutaka Kidawara†, Katsumi Tanaka††, Kuniaki Uehara†††

†Kobe Research Center, TAO

††Division of Intelligence Science, Graduate School of Science and Technology, Kobe University

†††Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University

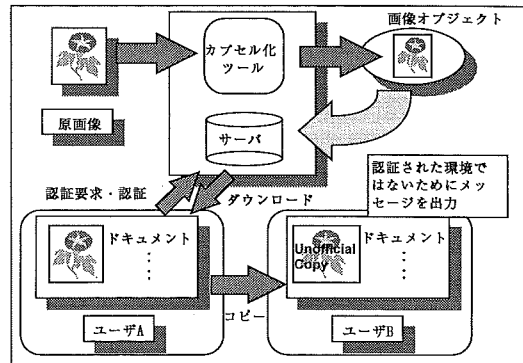


図 1: 版權管理手法の概念

新しい表現手法を組み込む事が可能である。

### 3 版權管理のための 캡セル化

ネットワーク上で決裁を行った後にそのマルチメディアデータをダウンロードする従来の方法では、決裁されたデータが完全な形でローカルコンピュータ上にダウンロードされるために、データの不正な変更やコピーが可能であった。我々が提案するマルチメディアデータオブジェクトでは、以下の機能を付加した 캡セル化を行う事によって版權管理を実現する。

- ユーザ認証機能
- マルチメディアデータ表示・再生機能
- ユーザ非認証時のデータ表示・再生機能

図 1 にマルチメディアデータオブジェクトを用いた版權管理手法の概念を示す。

### 4 Java を用いた画像データ 캡セル化

今回、我々は GIF や JPEG 等の画像データを対象として Java を用いた 캡セル化による版權管理のための画像オブジェクトを開発した。Java は OS に依存しないプログラムの作成が可能であり、Java で作成したオブジェクトは従来の画像データ同様に様々な環境で使用できる。

#### 4.1 画像オブジェクトとしてのアプレット

Java アプレットをネットワーク上で流通するオブジェクトとして考えるとその中に画像データを 캡セル化す



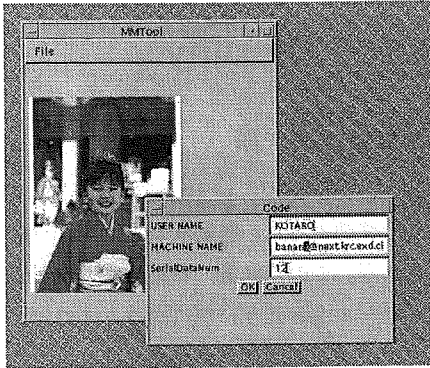


図 2: 画像データカプセル化ツール

る事で画像オブジェクトとして用いることができる。しかし通常のアプリレットでは実行時に外部の画像データを読み込むため画像がカプセル化された状態を永続化することが困難である。我々は画像データをアプリレットの初期値として組み込むことで画像データの隠蔽化と永続化を実現した。

#### 4.2 画像データカプセル化ツール

通常の画像データを Java でカプセル化するためには、画像データを解析し、その値を組み込んだ Java のソースコードを作成する必要がある。しかし画像データは多くのピクセル情報から構成されており、手作業でプログラミングすることは困難である。そこで我々は任意の画像データをカプセル化するための画像オブジェクト生成ツールを開発した。このツールは JPEG や GIF 等の画像データを読み込み、画像情報を記録した後に任意のデータと共に画像情報を組み込んだ Java のソースコードを自動生成する。作成されたソースコードはコンパイルを行った後に画像データがカプセル化されたアプリレット、つまり画像オブジェクトとなる。

これによって、Java の知識を持たないユーザでも必要な画像データを自由にカプセル化する事ができる。図 2 にカプセル化ツールのインターフェースを示す。

#### 4.3 著作権管理のための画像オブジェクト

カプセル化ツールを用いて作成した画像オブジェクトの使用例を図 3 に示す。この例ではユーザがワードプロセッサを用いて認証された 3 つの画像オブジェクトと 1 つの認証されていない画像オブジェクトを使用したドキュメントを製作している。認証されていない画像オブジェクトでは unofficial copy というメッセージが原画像上に表示されて使用できないが、認証された画像オブジェクトではそれらのメッセージは表示されず、完全な画像として



図 3: 画像オブジェクト使用例

表示されている。

このドキュメントを他の認証を受けていないユーザが使用した場合には、すべて unofficial copy メッセージが出力されるために不正なコピーの使用は不可能となる。これらの処理は画像オブジェクト内の表示メソッドが行っている。従ってワードプロセッサ側では画像オブジェクトに対して表示メッセージを送信しているだけであり、認証機能について特別な機能は必要としない。またこの機能はワードプロセッサ、WWW ブラウザ等の様々な Java が動作するアプリケーション上で有効である。

#### 5 おわりに

本研究では、マルチメディアデータをカプセル化する事によって得られる機能について検証を行った。そして Java を用いて画像データをカプセル化する事によって様々な環境下で使用することが可能となる著作権管理のための画像オブジェクトが実現できた。今後は、認証機能の強化、画像オブジェクトの配布、管理及び編集機能について研究を行う予定である。

#### 謝辞

本研究は、一部、文部省科学研究費重点領域研究 (課題番号 08244103) による。

#### 参考文献

- [1] Watermarking Technology:  
<http://www.digimarc.com>
- [2] Data Hiding:  
<http://www.trl.ibm.com/projects/s7730/Hiding/index.htm>
- [3] 木俣 豊, 田中 克己, 上原 邦昭: 著作権管理のための Java による画像データカプセル化, 情報処理学会研究会報告, 97-DBS-111 (1997)

# 小型情報機器のための Scroll Display

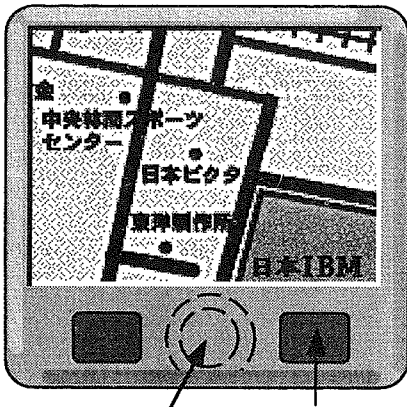
椎尾 一郎

日本アイ・ビー・エム (株) 東京基礎研究所

## 1. はじめに

手のひらに収まる程度の大きさの超小型携帯コンピュータが、個人用情報機器(PDA)などの用途で活発に利用されるようになった。しかし、表示装置が小型であるため、大きな面積を持つオブジェクト(広いデスクトップ、WWWページ、新聞・地図・書類など)を閲覧・操作することが困難である。筆者は、超小型コンピュータに、指示装置の機能を組み込むことで、この課題を解決しようとした[1]。本研究では、この試作機を用いて評価実験を行った。

## 2. システムの概要



mouse ball (back) mouse button

図1. 背面にマウス装置を内蔵した超小型コンピュータ

図1に本研究で提案する超小型携帯コンピュータの概要図を示す。コンピュータの背面にメカニカルマウスのボール機構を組み込んでいる。コンピュータ

を、机上で移動すると、移動量と方向がマウス部分で検出される。この移動量/移動方向だけ、表示内容をスクロールするようにプログラムすれば、使用者は、あたかも、大面積の書類の一部をのぞき見るウィンドウ枠を手にとり、見たい方向に移動する感覚を得ることになる(図2)。この結果、小さな表示画面であっても、直観的でわかりやすい閲覧操作を提供できる。

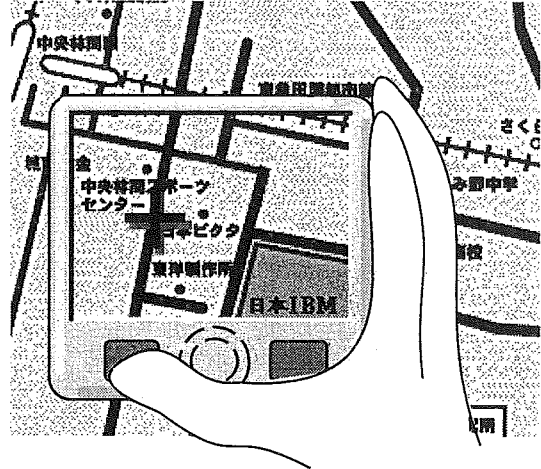


図2. コンピュータの移動に従って内容をスクロールすることで、大きなオブジェクトの部分を表示する仮想的な窓枠を掴んで閲覧するユーザモデルを提供する。十字のポインターは、表示部分の位置を反映する位置に表示する。

## 3. オブジェクトの操作

### 3.1 ポインター

表示されるオブジェクトを指示して、直接操作を行うために、ポインターを表示する。ポインターは表示画面にはほぼ固定されていて、ユーザは指示したいオブジェクト(たとえばWWWページのボタン)の真上にポインターが重なるまでコンピュータを移動し、ボタン操作により直接操作を行う。

### 3.2 メニュー

ボタン操作により表示するポップアップメニューに対して、他のオブジェクトと同様に、ポインターを表示装置に固定し、コンピュータを上下に動かすことで、メニュー項目を選択する。表示装置に収まらない長いメニューの場合は、その一部を覗き見る操作になる。(図3)

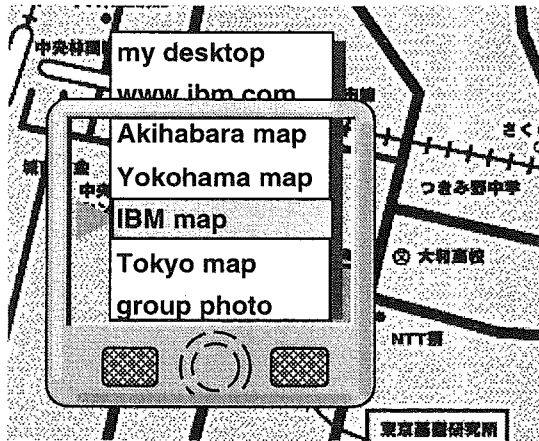


図3.ポップアップメニューを表示し、コンピュータを移動すると、表示中央の項目が選択される。

### 4. 試作と評価実験

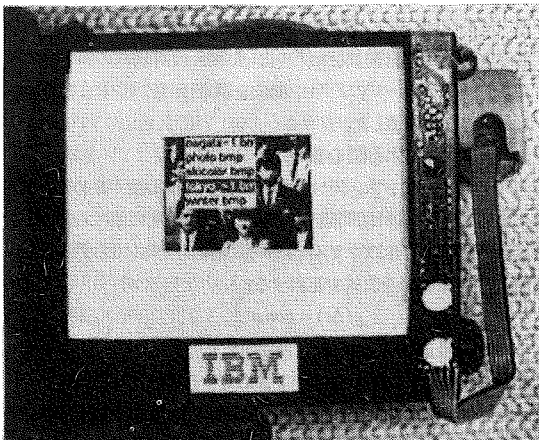


図4.試作した超小型コンピュータ。マウスを内蔵し、本体を移動することでオブジェクトの閲覧・操作を行う。

本方式による指示機構の有用性を確認する目的で、図4に示す試作機を作製した。ここでは液晶表示装

置の右部分に小型シリアルマウス機構を付加している。最終的には単体で稼働すべきであるが、試作機ではノートブック PC に有線で接続し、オブジェクト操作プログラムを PC 側で動かす、操作結果を試作機のディスプレイに表示した。

この試作機を 10 人以上のコンピュータ経験者に使用してもらった。その結果、小さな表示装置であっても、大きな地図や集合写真を直接的にスクロールできることが使いやすいと評価された。

また、3 人の被験者を対象に Fitts の法則[2]に基づく評価実験を行った。一人の被験者が、距離(D)が 5mm から 320mm 先の、幅(S) 5mm から 40mm の目標を 5~10 回選択するのに要した平均時間を図5に示す。距離が 40mm 以上の目標のほとんどは、操作開始時に試作機のディスプレイ(44.5x37.0mm)に表示されない。しかし、短距離の目標と同じ直線上にあり、Fitts の法則に基づいた結果が得られた。他の二名の被験者の結果も同様であった。仮想的な窓枠を移動して閲覧するユーザモデルが有効に働いていると考えられる。

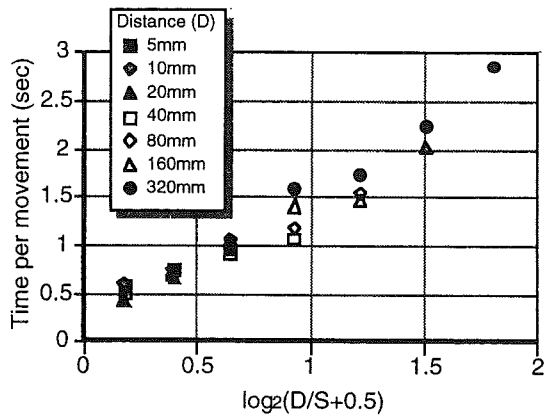


図5.Fittsの法則に基づいて解析した目標選択の所要時間

#### [参考文献]

[1] 椎尾一郎: “超小型コンピュータのための直接操作”, 情報処理学会モバイルコンピューティング研究グループ研究報告, Vol.96, No.MBL-3, pp.59-64,1996.11.21-22  
 [2] Stuart K. Card, Thomas P. Moran, Allen Newell, “The Psychology of Human-Computer Interaction”, Lawrence Erlbaum Associates Publishers, 1983.

# 時系列情報を考慮したアクセスログ解析\*

畑島 隆† 元田敏浩‡

日本電信電話株式会社 ソフトウェア研究所

takashi@canary.sl.cae.ntt.co.jp motoda@canary.sl.cae.ntt.co.jp

## 1. はじめに

WWWコンテンツへのアクセス数の計測手法では、サーバのアクセスログを解析する手法がある。また、アクセス履歴からユーザモデルを構築[1]し、インターネット上での one to one marketing 等に適用する試みがある。しかしこれらの手法では、いずれも一定期間のアクセスの総量を算出し、期間中のアクセスを平等に扱うため、アクセス履歴情報が損なわれている。

本稿ではアクセス履歴の時系列を考慮した評価指標「関心度」を提案し、適用領域を明らかにする。

## 2. 関心度の定義

本稿で提案する関心度は、WWWアクセスログの時系列情報を考慮[2]し、コンテンツへのアクセス履歴を時間関数により得点化したものであり、以下の項目で定義される。

- ①コンテンツへのアクセス間隔
- ②コンテンツに対する直前のアクセス時点の関心度

アクセスが集中すると関心度は高くなり  
以下に関心度の評価式の定義を述べる。

## 2.1 評価式の定義

関心度ではアクセスによる増加と時間の経過による減衰が同時に作用する。関心度の評価式の要求条件を以下の時間関数により定義する。

直前のアクセスからの時間間隔  $\Delta t$  について、アクセスによる関心度の増加あわす関数を  $g(\Delta t)$  とする。また、時間経過による関心度の減衰を、直前のアクセス時点の関心度との比率によってあわす関数を  $h(\Delta t)$  とする。このときアクセスの発生していない任意の時刻  $t(t_n < t < t_{n+1})$  の関心度  $F(t)$  を次式で定義する。

$$F(t) = h(\Delta t)F_n \quad \text{ただし、} \Delta t = t - t_{n-1} \quad (1)$$

$$F_n = g(\Delta t) + h(\Delta t)F_{n-1} \quad \text{ただし、} \Delta t = t_n - t_{n-1} \quad (2)$$

ただし  $F_n$  はコンテンツに対する  $n$  回目のアクセス(時刻  $t_n$ ) 時点での関心度である。

また、各関数の要求条件として以下の式を定義する。

・関心度刺激関数  $g(t)$

$$g(t) > 0, \quad \text{ただし } g(0) = \max(g(t)) \text{ (定数)}$$

$$g'(t) \leq 0 \quad (3)$$

・関心度減衰関数  $h(t)$

$$0 \leq h(t) \leq 1, \quad \text{ただし } h(0) = 1$$

$$h'(t) < 0, h''(t) \geq 0 \quad (4)$$

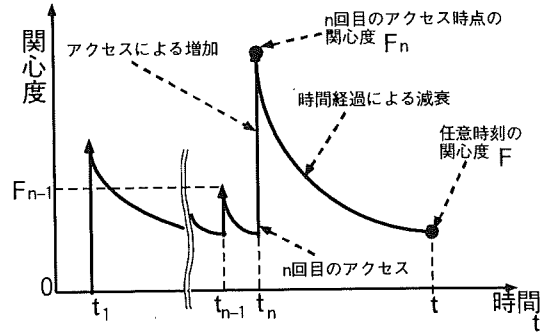


図1: 関心度の変化

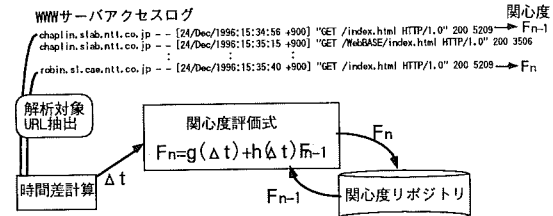


図2: 関心度評価システムの構成図

式1で提案した関心度の変化の様子を図1に示す。また、解析対象をコンテンツのURLで指定する場合の関心度評価システムの構成は、図2のようになる。以下に本節の定義にしたがって、関心度刺激関数と関心度減衰関数をモデル化した関心度評価式の例を述べる。実際のWWWユーザのアクセス動向を調査し、アクセスによる関心度の変化の様子を調査した。

## 2.2 アクセスによる関心度増加

ユーザによるコンテンツへのアクセスにより、関心度が高められる。変化量はアクセスの時間間隔や、それまでの関心度の値によって決定される。しかし、今回はアクセスによる刺激を時間間隔によらず一定とし、関心度刺激関数を次式で定義する。

$$g(\Delta t) = const. \quad (5)$$

## 2.3 時間経過による減衰

前述の定義により、あるコンテンツに対する関心度は、そのコンテンツに対してアクセスが存在しない限り減衰する。関心度減衰関数をモデル化するために、WWWサーバのアクセスログから実際のWWWユーザのアクセス動向の調査を行った。調査対象としてNTT DIRECTORY<sup>1</sup>のサーチエンジンInfoBee<sup>2</sup>への検索キーワード投入件数を用いた。キーワードが現わすイベント終了後同じイベントが発生しない期

\* An Analyzing of the Access\_logs Regarding its Time Sequence Data.

† Takashi Hatashima, NTT Software Laboratories.

‡ Toshihiro Motoda, NTT Software Laboratories.

<sup>1</sup> NTT DIRECTORY, <http://navi.ntt.co.jp/>

<sup>2</sup> InfoBee, <http://navi.ntt.co.jp/j/info/infobee.html>

間において、時間経過による検索数の減少が顕著に現れるキーワード「選挙」<sup>3</sup>に着目した。選挙終了後の検索数の時間経過による減衰を、関心度の減衰の一例として関心度減衰関数に用いる。

また、昼間・夜間といった時間帯による検索数の変動を正規化するために以下の式であらわされるアクセスシェアを各時間帯で求めた。

$$\text{アクセスシェア} = \frac{\text{当該時間帯の対象語検索数}}{\text{当該時間帯の総検索数}} \quad (6)$$

関心度の減衰を①情報の利用度の低下[3]②情報の老化[4]③記憶の忘却[5]の観点から考察した結果、知見において減衰モデルは指数関数により定義されているため、関心度減衰関数に指数関数を用いた。

時間経過による自然な減衰傾向を示すと考えられる期間のアクセスシェアの減衰を近似して求められた時間関数を関心度減衰関数とした。投票終了後、開票情報に対する検索要求が発生し、20日19時にピークとなった。翌日起床後に情報検索するユーザを除外するため、アクセスシェアが増加する直前の21日午前4時までを選択した。図3にこの期間のアクセスシェアの実数と関心度減衰関数を示すように、ピークとなった時刻 ( $t = 0$ ) 以降の時間経過を  $\Delta t$  [hour]として、前述した定義により  $h(0) = 1$  を切片とすると、関心度減衰関数は次式のように導かれる。

$$h(\Delta t) = \exp(-0.2765\Delta t) \quad (7)$$

今回は関心度刺激関数  $g(\Delta t)$  を定数とし、1回のアクセスにつき1点の関心度値が加算されるとしたため  $g(\Delta t) = 1$  となるので、 $n$  回目のアクセス以降の任意の時刻 ( $t_n \leq t < t_{n+1}$ ) における関心度  $F(t)$  は次式で定義される。

$$F(t) = F_n \exp(-0.2765\Delta t) \quad \text{ただし } \Delta t = t - t_{n-1} \quad (8)$$

$$F_n = 1 + F_{n-1} \exp(-0.2765\Delta t) \quad \text{ただし、} \Delta t = t_n - t_{n-1} \quad (9)$$

なお、 $F_n$  はあるコンテンツに対する  $n$  回目のアクセス (時刻  $t_n$ ) 時点における関心度であり、 $\Delta t$  の単位は [hour] である。

### 3. 関心度の適用領域

関心度の以下の事例へ適用を検討している。

#### ・ キャッシュ管理

NTT DIRECTORY<sup>1</sup> のような WWW と DB の連携により検索サービスを提供するシステムでは、検索結果が逐次キャッシングされる。キャッシュ管理に関心度を適用し、関心度の高い検索結果だけをキャッシングする。これにより最近多く検索された内容だけがキャッシングされるため、キャッシュ利用効率の向上が見込まれ、より多くの検索要求に対して高速応答が可能になる。

#### ・ コンテンツランキングの提供

関心度の高いコンテンツをランキング表示することにより、人気コンテンツを表示するサービスを提供する。

#### ・ コンテンツ鮮度管理

関心度の低いコンテンツに対してしきい値を設定することにより更新や廃棄の基準とする。

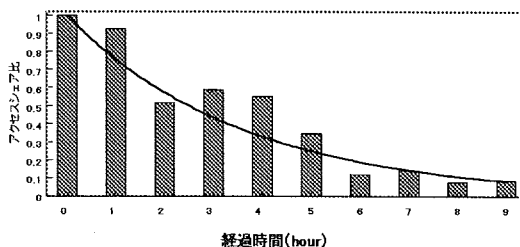


図3: アクセスシェアのピーク以降の時間変化と指数近似のグラフ

なお、関心度評価値の算出にはアクセス履歴の時間情報のみを用いているため、WWWサーバに限らずアクセスに対してタイムスタンプを発行する機能を持つサーバのコンテンツ管理に適用可能である。

また、関心度は同一サーバ上のコンテンツ間の相対評価の指標であるため、同一条件でのアクセスログ解析が可能なWWWサーバ内のコンテンツ管理に有効な指標であるといえる。しかし、複数のWWWサーバについての解析では、I/PRO<sup>4</sup>のように他サーバのアクセスログを関心度解析サーバに転送し解析することにより、公正な評価手段として適用可能となる。

### 4. 課題

関心度減衰関数の関数のモデル化に用いた検索キーワード「選挙」によって、

① 短期間にアクセスが集中

② 単発のイベントであり他のイベントとの関連が薄い

という性質のはっきりした事象に対する情報要求をモデル化したのが、実際の関心度の減衰に近づけるためにはより多くの事象について調査する必要がある。

また関心度刺激関数を定数にしているため、アクセスの集中により関心度が無限大に発散してしまうことが予想される。しかし実際にはあまり多く同じ事象に触れると「飽き」がくるように、関心度についても増加の上限があると考えられるため、これを考慮した関心度の成長モデルを定義したうえで関心度刺激関数を定める必要がある。

### 5. まとめ

本稿ではWWWサーバへのアクセスの時系列を考慮したアクセス動向調査指標として関心度を提案した。実際のアクセスログにおけるアクセス件数の減少傾向から関心度減衰関数を導いた。また現在検討中の実システムへの適用例を示した。今後は多くの事例に検討モデルを進めるとともに、実システム上での評価を行う予定である。

### 参考文献

- [1] 三浦, 高橋, 島, "個人適用型WWWのためのユーザモデル構築", 情報処理学会, Interaction 97, Feb, 1997
- [2] 畑島, 元田, "WWWアクセスログの有効な解析法について", 第53回情処全大, 分冊 3, pp.217-218, 1996
- [3] Cole, P. F., "Journal usage versus age of journal", Journal of Documentation, 19, pp.1-11
- [4] Griffith, B. C. et al., "Aging of scientific literature: a citation analysis", Journal of Documentation, 35, 1980, pp.179-196.
- [5] 古川, "寿命の数理", 行動計量学シリーズ 13, 朝倉書店, 1996, pp.191-196

<sup>3</sup> 第41回総選挙, 告示 1996.10.8,

投票時間 1996.10.20, 7:00-18:00 (即日開票)

<sup>4</sup> I/PRO, <http://www.ipro.com/>

## 6P-07 拡張有限状態機械とペトリネットを表示編集できる GUI ツールの作成と応用例

山本 亮 岡野 浩三 東野 輝夫 谷口 健一

大阪大学 基礎工学部 情報科学科

## 1 まえがき

著者らの研究グループでは、通信プロトコル、ハードウェア・ソフトウェアの設計法、詳細化の正しさの証明法等の研究を行っている。この際、拡張有限状態機械 (EFSM) モデルやペトリネット (PN) モデルを用いることが多い。これらの設計ツールや証明支援ツールを作成する際、EFSM や PN を表示編集でき、また、これらの図上で、例えば、マウスのクリックで特定のコマンドを実行できるような汎用ツールが望まれる。そこで、以下の指針で EFSM や PN の図表示・編集ツールを Tcl & Tk ツールキットを用いて作成した。

1. 種々のアプリケーションから容易に利用できること。2. EFSM・PN が容易に記述、編集できること。3. EFSM・PN の図表示が美的であること。4. 利用者が各イベントで起動される関数を容易に定義できること。5. ボタンの追加など、システムの変更が容易にできること。

これらのうち 1. の一部と 5. は Tcl & Tk を用いることにより解決できる。2 章では主に 2.~3. に関してどのようにツールを作成したかについて述べる。3 章では、このツールの具体的な応用事例と各応用に要した手間について述べる。

## 2 表示ツール

## 2.1 表示ツールの機能

以下では、ノードは EFSM における状態あるいは PN におけるプレースまたはトランジションを表す。

1. ある定められた文法 (GDG と呼ぶ) に従って記述された一つの EFSM あるいは PN のファイルを入力として、キャンバス上に図表示できること。また、キャンバス上に図表示された EFSM あるいは PN を GDG 記述でファイルにセーブできること。
2. キャンバス上に図表示された EFSM あるいは PN の編集ができること。具体的には、
  - (a) ノードの追加、削除、移動ができること。
  - (b) ノードの削除、移動においては、関連するアークが削除あるいは移動できること。
  - (c) ラベルやトークンの変更ができること。
3. ファイルから読み込む際、各ノードを美的に自動配置できること。あるいは、別途セーブしていた各ノードの配置情報に基づいて配置できること。

4. 上記の諸機能とノードやアーク、ラベルのハイライト表示やトークンの移動等がすべて関数呼出しの形式で行えること。

機能の 1. は図表示システムとして当然の機能である。本ツールでは図形群のグループ化の機能をつけず、代わりに、2.(b)の機能を設けている。EFSM・PN に編集対象を限定しているため、指針の 2. を満たすにあたって 2.(b)の機能が有効であると考えられる。機能 3. は指針の 3. に対応する。本ツールでは機能の 4. の関数群を基本機能として提供する。これらの基本関数を組み合わせることにより、イベントで起動される関数を容易に使用者が定義できる (指針 4.)。

## 2.2 表示ツールの実現

機能の 1., 2. は Tk の Canvas Widget を用いて実現した。Canvas Widget では、円、長方形、線などをオブジェクトとして扱い、これらの生成、移動、消去、イベントとの結びつけ等を行うことができる。これらのオブジェクトの接続情報を連想配列等に保持することにより、機能の 2.(b)を実現できた。入力モードとして、EFSM, PN, Free の三つがあり、例えば、PN の入力の際にトランジション同士を接続できないようにしている。

機能 3. を実現するために文献 [1] のアルゴリズムを用いた。これは FSM の美的配置アルゴリズムであり、ノードとアークの接続関係、開始状態、終了状態を入力とし、各ノードの美的な配置情報 (すなわち各ノードの x-y 座標) を出力とする。EFSM に対してはそのまま利用した。PN に対してはプレース、トランジションをともにノードと見なしてこのアルゴリズムを適用し、ほぼ自然な配置が得られた。作業者が適宜、修正したい例もあったため、修正した x-y 座標を保存する機能やグリidding機能も追加した。

このツールは一つのウィンドウに一つの EFSM, あるいは一つの弱連結な PN を表示できる。アプリケーションによっては、複数の EFSM を表示したい場合も多い。この場合はこのツールを各 EFSM (または PN) ごと用いる。各ツール間の通信には、Tcl & Tk の send 機能を用いる。後述の応用例ではこの send 機能を用いている。

## 3 応用例

## 3.1 グループウェア実行支援系の GUI 部への本ツールの適用

文献 [3] におけるペトリネットモデルに基づくグループウェア実行支援システムはグループワークに

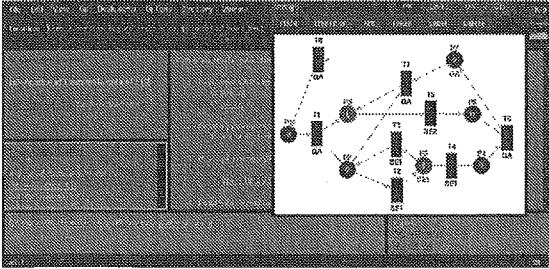


図 1: グループウェア実行支援システムの実行画面

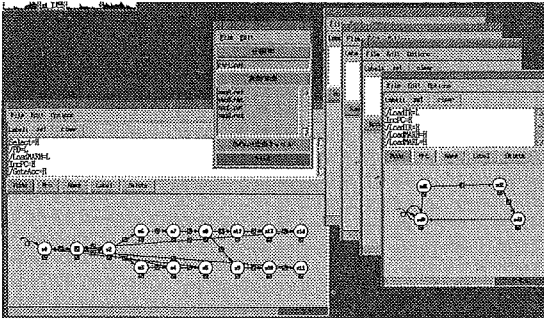


図 2: 証明支援システムの実行画面  
 おける各個人毎の作業内容(文書作成, コーディング, デバッグ等の)を各個人毎 WWW client を通じてガイダンスする. 各個人の作業順序はペトリネットとして与えられているため, 図示できることが望ましい. そこで, この表示ツールを用いて作業順序をペトリネットとして図示するようにした. 各作業はペトリネットのトランジションに対応し, 作業の進捗は, 図表示(トークンの移動, ハイライト表示の移動)で確認できる(図 1). 本ツールはペトリネットの接続情報とトークンの情報を入力とし, 上記のような表示を行う.

### 3.2 ハードウェア検証系の GUI 部への本ツールの適用

一つの EFSM として仕様が与えられたハードウェアの制御部を複数 EFSM 群に分割して実現する際に, 実現がもとの仕様に対して「論理的に正しいこと」を証明する自動証明アルゴリズムを考案し [2], 証明システムを作成した. この証明システムは文字端末上で使用するコマンドであり, 仕様 EFSM と実現である複数 EFSM 群の記述ファイルを入力とする. 実現が仕様に対して正しいと判断したときにはコマンドは「正しい」というメッセージを出力し, そうでない場合は間違いであると判断した遷移や実行動作を出力する. 本表示ツールと組み合わせることにより簡単に GUI ベースの証明システムにできた. 図 2 に実行画面を表す.

左のウィンドウは仕様の EFSM であり, 右にある複数のウィンドウは実現 EFSM 群である. 中央

のウィンドウにある proof ボタンを押すと証明コマンドが実行される.(誤りであるため)遷移や実行動作を証明コマンドが出力した場合は, 図表示された EFSM 上の当該遷移や実行動作がハイライト表示される. また, 仕様や実現 EFSM 群を Canvas 上で直接入力したり, 遷移の修正を行うこともできる. これらの機能を本ツールを用いて実現する際に新たに作成した機能は以下のとおりである. (1). あるハードウェア記述言語 (AHDL) で記述された各 EFSM のファイルを GDG に変換する(逆向きも含む). (2). 複数のツールや証明コマンド等の実行調整を行う. (3). 遷移のラベルのサイズが大きくなるのでラベルのアイコンを普段表示することにし, 内容を別途参照できるようにする. (1) は yacc を用いて文法変換コマンドを作成することにより実現した. (2) には, 前述の send 機能を用いた. (3) は, 指針の 5. に相当する作業であるが Tk の提供する基本 Widget 群を組み合わせるだけで実現できた.

### 3.3 作業工数

ハードウェア検証系を GUI 化する際に必要となったプログラムのプログラムサイズとコーディング時間について, ツール本体とともに以下に表す. なお, グループウェア実行支援システムについては新たに作成したプログラムはほとんどない.

項目	サイズ(言語)	コーディング
ツール本体	2700 行 (tcl)	1ヶ月
実行調整・ラベル表示部	450 行 (tcl)	1週間
文法変換部(両方向)	600 行 (yacc,c)	1週間

このように, 比較的短期間で GUI 化を行うことができた. GUI の制御のためのプログラムは Tcl & Tk を利用していることもあり非常に短期間で作成することができた.

## 4 あとがき

EFSM や PN を編集表示できる GUI ツールを作成し, 応用例を報告した. 今後の研究課題として, Tree 表示機能を実装することや, インタラクティブなコマンドに対応できるように双方向パイプを実装することなどを考えている.

謝辞 描画ルーチンを提供していただいた朝倉氏(大阪大学基礎工学研究科修士課程)に感謝します.

### 参考文献

- [1] 朝倉 義晴: “状態遷移図の自動描画アルゴリズム”, 大阪大学 基礎工学部 情報工学科 特別研究報告 (1996).
- [2] 森岡 澄夫 他: “一つの EFSM の複数 EFSM による実現の正しさの一証明法”, 第 52 回情報大全 1K-1, (1996).
- [3] 山口 弘純 他: “レジスタ付きペトリネットモデルで記述されたソフトウェアプロセスの実行支援”, 情報処理学会平成 7 年度サマワーショップ・イン・立山論文集, pp. 89-96, (1995).