

特集 「ユーザモデリングと適応的インタラクション」

# 知的インタフェースにおけるユーザモデリングと 適応機能

## User Modeling Methods for Intelligent Interface Systems

吉田 健一\*  
Kenichi Yoshida

\* (株)日立製作所システム開発研究所  
Systems Development Laboratory, Hitachi, Ltd., Yokohama 224-0817, Japan.

1998年11月2日 受理

**Keywords:** intelligent/adaptive interface, user modeling, machine learning, knowledge acquisition.

### 1. はじめに

計算機と利用者の対話過程に関する情報に基づき、利用者の挙動や作業のモデルを構築し、モデルを利用することで、より使いやすい計算機インタフェースを実現しようという研究の歴史は古い。

例えば、利用者の計算機操作手順に繰り返し構造がないか観測し、度々繰り返される操作手順を自動化する HyperText システムとして、EAGER [Cypher 91] が有名である。Schlimmer らは繰り返し現れる言葉を検知することでメモ作成を支援するシステムを提案している [Schlimmer 93]。Masui [Masui 94] らは繰り返し構造を利用する emacs エディタを報告している。

統計や機械学習の技術を用いて、利用者の計算機操作手順に内在する規則性を抽出しようという研究例も多い。文献 [Dent 92, Hermens 93, Mitchell 94] 等では分類木学習方法の適用が、文献 [Maes 93] では K-nearest neighbor 法の適用などが試みられ、会議スケジュール・システムや電子メール分類システム等が開発されている。

これらは「繰り返し起こる操作手順の自動化」が計算機の使いやすさ向上に有効であることを示した研究例と見ることができる。また、利用者の計算機操作手順に内在する繰り返し構造自体の研究の歴史も古く、例えば、Greenberg らはコマンド履歴中に現れる繰り返し構造を分析・報告している [Greenberg 88]。

コマンド予測など古くから行なわれていた利用者の

挙動を予測しようとした研究は、利用者の「気まぐれな挙動」を扱うことの困難さから必ずしも実用的な「予測システム」にはつながっていない。しかし、モデル化技術自体は、WWW ブラウザに組み込んだ閲覧支援システム [Joachims 97, Sakagami 97, Sakagami 98] においてユーザの URL 選択を支援したり、Prefetching により WWW アクセスの高速化を計る技術 [Banatre 97] として、近年も利用が試みられている。これら最近の研究事例は、モデルによる予測の困難さを認識した上で、その「限界のある技術」の利用方法を考案した点に特徴がある。しかし、利用者の挙動をモデル化する技術としての統計や機械学習技術の重要性を否定するものではない。

本解説では、著者らが開発したユーザ適応型インタフェース [Yoshida 96] を例に、ユーザ支援機能実現におけるモデル構築の重要性と、機械学習など人工知能技術の応用、及びこのようなアプローチの限界に関して述べる。また、モデル構築過程をなるべく機械化した上記インタフェースの実装例と、うまく人手を使うことを指向したヘルプデスクシステムの実装例 [Kang 97] を対比することで、幾つかの考察を行う。

### 2. モデルに基づくユーザ支援

#### 2.1 ユーザ適応型インタフェース

著者らは「計算機利用者の挙動モデルを構築する上で、操作手順中の繰り返し構造の分析に加え、利用者が遂行している作業間の依存関係を分析することが役

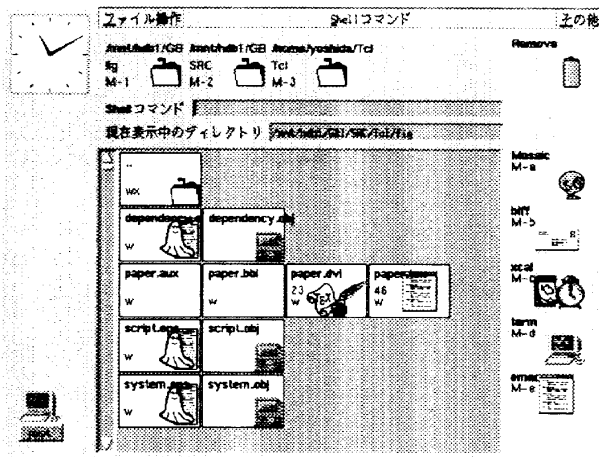


図1 ClipBoard のスクリーンイメージ

立つ」と考え、ClipBoard と呼ぶインタフェースシステムを試作した [Yoshida 96]。

図1にClipBoardの表示画面例を示す。ClipBoardはウィンドウシステム上のファイル操作支援インタフェースで、利用者が入力デバイスを用いて操作対象となるファイルを指定（一般にはマウスを用いてファイルをクリック）した場合に、ファイルの処理に使用するアプリケーションプログラムを選択する。従来の類似システムは常にあらかじめ指定しておいた単一のアプリケーションが選択されていたが、このシステムでは作業の進捗状況や利用者の癖に応じて起動するアプリケーションの選択を切り替える。

図1中心部分に並んだ小さな四角は、それぞれ計算機上のファイルであり、四角の中に示されたアイコンは自動選択・起動されるアプリケーションプログラムを示している。ClipBoardは、利用者の指示で作業を実行するコンピュータプログラムの入出力を観察し、複数のプログラム間の入出力関係から利用者の遂行している作業の間の依存関係を分析し、アプリケーションプログラムの選択を切り替える。アプリケーションの選択が切り替わった場合、画面上のアイコンを対応するものに変更する。

ClipBoardもユーザの挙動を予測しようとしたシステムであるが、次にどのファイルを使うかというヒントと、作業間の依存関係という追加情報を利用することで、操作手順のみからの予測より精度を高めようと試みた研究である。図2にClipBoardが利用者挙動のモデル化に利用している作業間の依存関係（実際にはアプリケーションプログラム間の入出力関係）の例を示す。

図2では文書処理システムとしてlatex（原稿であるtexファイルを処理する主プログラム）を、文献デー

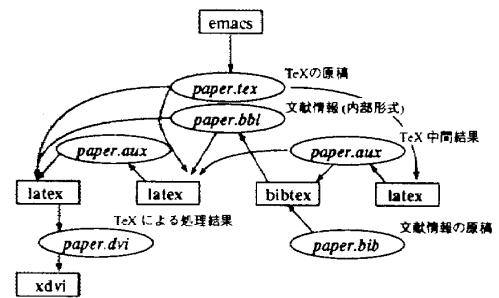


図2 ClipBoard が利用者挙動のモデル化に使った I/O 情報

タ処理システムとして bibtex (latex の中間出力と文献データベースである bib ファイルから関連する文献データを抽出する補助プログラム) を、及び、清書結果表示プログラムとして xdvi を利用している。また、latex と bibtex の仕様から、文献データを処理するために複数回 latex プログラムが起動されている。

ClipBoard は図2のようなグラフ形式で表現されたデータを分析し、図2でハッチされたようなパターンを利用者挙動のモデル（「利用者は emacs, latex, xdvi プログラムを順に用いて文書処理をする」というモデルに相当）として抽出するため、Graph Based Induction 法 (以下 GBI 法と呼ぶ) [吉田 95] という帰納学習方法を用いている。

ClipBoard の内部処理として GBI 法が分析するデータは、図2に例示したグラフが多数含まれたもの（または1つの大きなグラフと見なすことも可能）で、この為に改造された専用 OS の出力である。GBI 法は入力したグラフを分析し、繰り返し現れる部分グラフを抽出する。[吉田 95] にはこの処理が分類木学習の拡張であることが述べられているが、ここでは単純に、何度も繰り返し現れる部分グラフが利用者挙動のモデルとして抽出されると考えればよい。

ClipBoard はこのような挙動モデルを獲得した後、emacs, latex と処理が進んだ後の dvi ファイルの処理用アプリケーションとして xdvi を選択する。

ClipBoard のように自動的な利用者挙動のモデル作成を試みたシステム実用化の難しさは、利用者の「気まぐれな挙動」への対応である。著者らは文献 [Yoshida 96] において、従来の作業手順の分析に加え作業間の依存関係を分析することがモデル作成に役立つことを報告した。作業間の依存関係はモデル作成において重要な情報を提供するが、モデルの学習には作業手順だけからの学習と同様に複数の挙動例を必要とし、急に作業内容を変えたときの追従には問題も多い。また、急な臨時作業への追従性を重視しすぎると、1 回限りの

臨時作業を敏感にモデルに反映しすぎるという問題も生じる。

このような利用者の「気まぐれな挙動」への対応は、学習時に利用する事例数の調整などによる学習速度の制御だけでは本質的な解決を与えられない。学習したモデルの利用形態の改良のようなモデル化技法以外の工夫も重要である。

## 2・2 ヘルプデスクシステム

利用者の「気まぐれな挙動」へ対応する1つの方法は、モデル化を全て利用者にまかせることである。利用者挙動のモデル化を自動化せずに、知識のマニュアル収集による知識ベースアプローチをとる研究も多い。スクリプトプログラムの自動作成システム APU [Bhansali 93] や、ビジネスチャートエディタ Gold [Myers 94] 等がこのアプローチをとっている。

このような、知識のマニュアル収集に頼った知識ベースアプローチを変化の速い分野に応用するとき、いかに知識ベースの構築・修正をサポートするかというのは大きな問題である。著者らはこの問題を検討するため、計算機の利用方法に関するヘルプデスクシステムを開発した [Kang 97]。現在、このシステムは事例文書中の単語出現頻度を解析 [Salton 94] して、類似した事例を事例データベースから検索するシステムの後処理システムとして実際の運用に向け調整が進められている。

図3は試作したシステムの表示画面例である。このシステムは計算機利用者が問題を解決するため情報を検索するのに用いるシステムで、問題内容を記述した検索用短文書と、使用しているオペレーティングシステム名称、アプリケーションプログラム名称、問題を表すキーワード等をシステムの入力要求に合わせて入力していくことで、過去の類似事例の情報が検索できる。

例えば図3は「小型パソコン用OS」を使って「ワープロ」を動かしているときに、ファイルの「保存」で問題が生じたことまで利用者が指定したときの画面例で<sup>\*1</sup>、問題をさらに絞るために「上書き/ファイル/ディスク」等が関係するか否かについて、システムが入力要求を行っている。

このシステムでは、図3画面上半分に示された質問に答えるうちに、画面下半分に示された類似事例が、より関係の深そうな事例(画面下半分に表示される2つの表のうち1つ目。図3では何も選択されていない)とそれ以外の事例(画面下半分に表示される2つの表の

\*1 画面例は実際のデータを用いて作成されたが、論文化にあたり具体的な商品名を普通名詞に置き換えた。

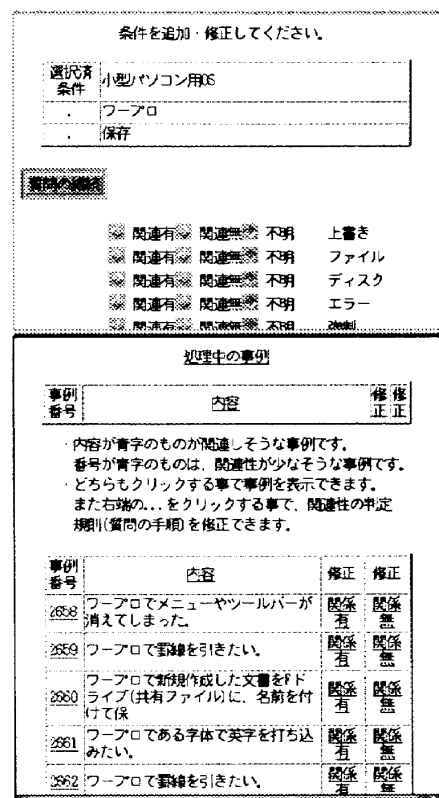


図3 ヘルプデスクシステムのスクリーンイメージ

うち2つ目。図3では事例番号2658, 2659等)に分類されていく。

画面下半分に示された類似事例は、検索用の短文書と事例文書中の単語出現頻度を解析し、類似した事例文書を検索した結果である。文書中の単語出現頻度の解析は、文書の意味を反映した解析が可能となることへの期待から近年注目を集めている。このシステムではさらに、初めの検索用文書が短く正確に意味解析できず、検索結果にノイズが多くなる問題に対応を試みている。具体的には、マニュアル収集による知識ベースを使ったシステムで、検索結果を関係のありそうなものとなさそうなものに分類している。

## 2・3 Ripple Down Rule 法

計算機は日々進化・変化しており、ヘルプデスクシステムがカバーすべき内容は急速に変化している。上記システムは Ripple Down Rule 法 [Compton 92] (以降 RDR 法と呼ぶ) という知識収集方法を利用して、知識ベースの内容が陳腐化したときに、即座に内容を修正する仕組みを持っている。具体的には、図3の画面による問診の結果が不適切なとき(関係ない事例を1つ目の表中に表示した場合)に、不適切な解答を与えた欄右端の関係無の部分をクリックすることで、知識ベースの修正を即座に行える。また、問診終了時にも(ま

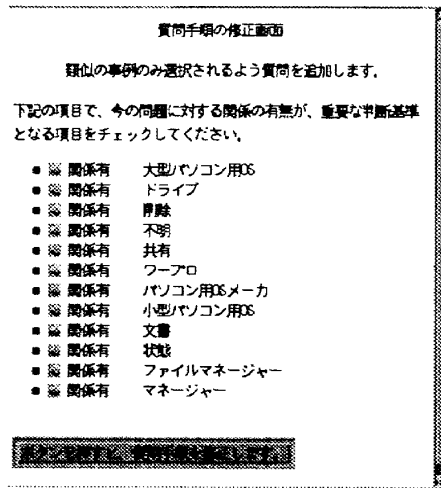


図4 ヘルプデスクシステムのインタビュー過程

たは途中においても) 関係のある事例が2つ目の表に表示されている場合、欄右端の関係有の部分をクリックすることで同じことができる。前者の場合、類似事例が検索されないようなフィルタリングルールが知識ベースに追加され、後者の場合、類似事例を関係ありとするようなルールが追加される。

図4に知識ベース修正画面の例を示す。この画面は問診終了時にも関係のある事例が2つ目の表に表示されていたときに、欄右端の関係有の部分をクリックして表示させた画面で、表示されているキーワードのいくつかを選択すると、選択されたキーワードを条件部に、該当事例が関係あることを結論に持つルールが知識ベースの適当な場所に追加され、以降の検索で同様なケースでは類似事例が関係ありとして、図中1つ目の表に表示されるようになる。

RDR法は例外知識の塊として知識ベースを構築する。知識ベースの大元(全てのルールの親ルール)には常に成り立つと考えるルール(条件部が恒真、図5でrule 0)として結論部に標準的な解答(例えば「解析不能」など)を持つルールを記憶しておく。全てのルールには、子供のルールとして、その例外事例を記録するルールを記憶できる。例えば図5でrule 1はrule 0の例外を記録したルールであり、rule 1の条件部にはrule 0の例外が成り立つ条件を記録しておく。

知識ベースをこのように構築しておく、大元のルールから始めて条件が成り立つ限り子供のルールを調べていき、最後に条件が成り立ったルールの結論部を使うことにより、状況に最も適合した結論を得ることができる。また、知識ベースの不備により結論が適切でない場合、不適切な結論をもたらしたルールに例外があることを記録(子供となる例外ルールを記録)する

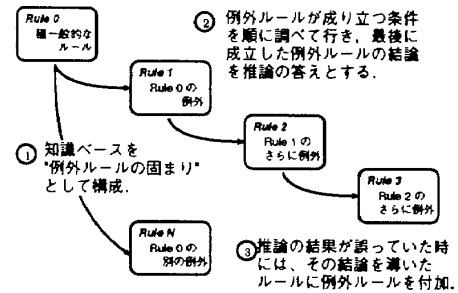


図5 RDR法の考え方

ことで、知識ベースを修正できる。図4の知識ベース修正画面の処理では、このような考えにより作成したルールの知識ベース上の記憶場所を決めている。

RDR法は、比較的経験の浅い利用者のための情報検索システムを専門家の援助のもとに構築する方法として、医療分野などで有効性が報告されている[Compton 92]。経験の浅い利用者がシステムを使って正解にたどり着けなかった事例を使い、専門家がシステムに新しい知識を追加していく過程から、場あたりの信用のおけない方法にも思われがちであるが、専門家がシステムに「簡便に」新しい知識を追加できるのが最大の利点である。

### 3. モデル構築用手法の比較

RDR法が使用する知識ベースの構造と、CART[Breiman 84]やID3[Quinlan 86]など機械学習システムが使用する分類木の構造との類似性には注意を要する。RDR法の知識収集の処理過程は、CARTなどが多数のデータの統計的な傾向を使い分類木に新規のテスト条件を追加する処理を、データの統計的な傾向の代わりに専門家の経験的な知識を使う処理に置き換えたものと見ることができる。作成した知識ベースや分類木の使用過程にも類似性がある。

図6は、上記の考えを使って[吉田 95]で著者らが報告した種々の分類木学習方法の関係を見直したものである。[吉田 95]では、伝統的分類木学習法[Breiman 84, Quinlan 86]やInductive Logic Programming法[Muggleton 92, Pazzani 92, Quinlan 90, Shapiro 83]並びにGBI法が、データの表現形式を変えた類似技術と見なせることを述べた。すなわち、種々の分類木学習方法は、

- (1) 入力したケースデータを作成中の分類木で分類するステップ
- (2) 分類後のケースデータ毎に分類用属性表を再作成するステップ

```

Algorithm 分類木学習
variable
   $G_{in}$  : ケース・データ
   $T^{in}$  : 分類木
begin
   $T \leftarrow \emptyset$ 
  repeat
     $T$  による  $G_{in}$  の分類
    Proc. 1, 2, 3 による分類用属性表の作成
    Proc. A, B による新規テスト条件の選択と
       $T$  への追加
  end
Proc. 1 伝統的分類木学習法
  常に同じ属性表を使用
Proc. 2 Graph Based Induction 法
  新しい属性をグラフの中から追加
Proc. 3 Inductive Logic Programming 法
  新しい属性を論理式の集合の中から追加

Proc. A 機械学習法
  情報量など機械的指標を利用
Proc. B RDR 流知識獲得法
  専門家の経験的な知識を利用

```

図6 モデル構築手法の比較

(3) 分類木に新規テスト条件を付加するステップからなっていると見なせ、上記3者の違いは2番目のステップにあることを述べた。

図6では、さらにマニュアルによる知識収集方法と機械的な分類木学習方法の類似点と相違点を明確にすることで、モデル構築の際の利用者の「気まぐれな挙動」への対応方法に関してヒントを与えてくれる。すなわち、知識収集方法と機械的な分類木学習方法が図6に示した関係にあるとして整理できるなら、作業開始直後の学習データが少ない時期はマニュアルによる知識収集を行い、データ数が増えてきた段階で機械的な学習方法に切り替えるシステムや、普段は機械的な学習方法にモデル作成をまかせるが、臨時作業を開始したときだけマニュアル操作で臨時モデルを作成するようなシステムの設計が可能に思われる。

知識収集の方法はRDR法だけではない。また、[吉田95]で整理した分類木学習方法も機械学習の方法を全ては網羅しておらず、機械的なモデル構築の方法全てではない。その意味で図6は不十分であるが、図6に示されたような対応関係が機械的な分類木学習手段とマニュアルによる知識収集手段の間に成り立つことを意識した上で利用者モデルの構築手法を考えることは、必要に応じて両者を使い分け、半自動的にモデルを構築するシステム開発に重要な視点と思われる。また、このような複合的なアプローチではインタフェースの作り方(両手法の使い分け方)が重要になるが、これに関しても残された研究課題は多い。

#### 4. モデルに関する考察

インタフェースの利用者適応機能実現には、3章で述べたモデル構築技法のみならず、獲得したモデルの利用方法など種々考察しておくべき点が多い。

どのタスクをモデル化し、どのようにモデルを用いるかという点は重要である。コマンド予測など、利用者の挙動を予測しようという場合、ターゲットを定型書類の作成等の業務に絞らないと、自動的なモデル化にしろ手作業による知識収集にしろ「気まぐれな挙動」の扱いに障害を生じる。[Hermens 93, Schlimmer 93]は定型性の強い作業に着眼してシステムを開発している。また、[Banatre 97, Joachims 97, Sakagami 97, Sakagami 98]等は動作が利用者には直接見えない場所での支援に機能を絞っている。これらはタスクの選択が正しく行われた研究事例ととらえることも可能である。

また、自動的な適応機能が必ずしも使用感の向上に結びつかない場合もありうる。[Masui 94]は自動的な適応を避け、利用者の明示的な指示により繰り返し構造を抽出する仕組みを実現している。自動的な適応により利用者が計算機の動作として予測する以外の動作をすることは必ずしも使用感の向上に結びつかないが、[Masui 94]はモデル作成を明示的に指示することで、この問題を避けていると見なせる。

また、モデル構築に用いるデータの選択も重要である。著者らは[Yoshida 96]において作業手順の分析に加え作業間の依存関係を分析することがモデル構築に役立つことを報告した。また、[Sakagami 97]は操作時間の利用を提案している。2.2節で述べたシステムでは、事例文書中から切り出した単語(具体的には図4中にリストアップされた単語)を利用してルールの特徴判定部を作成している。利用者の意図を反映できるように用いるデータを選択することは、作成されるモデルの精度に直接影響する重要な考慮点である。

#### 5. おわりに

本解説では、著者らが開発したユーザ適応型インタフェース[Yoshida 96]と計算機の利用方法に関するヘルプデスクシステム[Kang 97]を例に、ユーザ支援機能実現におけるモデル構築の重要性と、機械学習・知識収集手法など人工知能技術の応用に関して述べた。さらに、機械学習法と知識収集手法の関連を述べると共にモデルに関する話題を概観した。

## 謝 辞

本論文中紹介した幾つかの仕事は、著者と大阪大学産業科学研究所元田浩教授の共同研究の成果である。日頃の御指導に感謝します。また、日立電子サービス殿からはヘルプデスクシステムに関して実際のユーザサポートの例を提供していただいた。貴重なデータの提供に感謝します。

## ◇ 参 考 文 献 ◇

- [Banatre 97] Banatre, M., Issarny, V., Leleu, F., and Charpiot, B.: Providing Quality of Service over the Web: A Newspaper-based Approach, in *6th International World Wide Web Conference*, pp. 301-309, 1997.
- [Bhansali 93] Bhansali, S. and Harandi, M. T.: Synthesis of UNIX Programs Using Derivational Analogy, *Machine Learning*, Vol. 10, pp. 7-55, 1993.
- [Breiman 84] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J.: *Classification and Regression Trees*, Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
- [Compton 92] Compton, P., Edwards, G., Srinivasan, A., Malor, R., Preston, P., Kang, B. H., and Lazarus, L.: Ripple Down Rules: Turning Knowledge Acquisition into Knowledge Maintenance, *Artificial Intelligence in Medicine*, Vol. 4, pp. 47-59, 1992.
- [Cypher 91] Cypher, A.: EAGER: Programming Repetitive Tasks by Example, in *CHI'91*, pp. 33-39, 1991.
- [Dent 92] Dent, L., Boticario, J., McDermott, J., Mitchell, T., and Zabowski, D.: A Personal Learning Apprentice, in *AAAI-92*, pp. 96-103, 1992.
- [Greenberg 88] Greenberg, S. and Witten, I. H.: How Users Repeat Their Actions on Computers: Principles for Design of HISTORY Mechanisms, in *CHI'88*, pp. 171-178, 1988.
- [Hermens 93] Hermens, L. A. and Schlimmer, J. C.: A Machine-learning Apprentice for the Completion of Repetitive Forms, in *Proc. of the Ninth Conf. on Artificial Intelligence for Applications*, pp. 164-170, 1993.
- [Joachims 97] Joachims, T., Freitag, D., and Mitchel, T.: Webwatcher: A tour guide for the world wide web, in *Proc. of IJCAI'97*, pp. 770-775, 1997.
- [Kang 97] Kang, B. H., Yoshida, K., Motoda, H., and Compton, P.: Help Desk System with Intelligent Interface, *J. of Applied Intelligence*, Vol. 4, pp. 297-328, 1997.
- [Maes 93] Maes, P. and Kozierok, R.: Learning Interface Agents, in *AAAI-93*, pp. 459-465, 1993.
- [Masui 94] Masui, T. and Nakayama, K.: Repeat and Predict - Two Keys to Efficient Text Editing, in *CHI'94*, pp. 118-123, 1994.
- [Mitchell 94] Mitchell, T., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D.: Experience with a Learning Personal Assistance, *CACM*, Vol. 35, No. 7, 1994.
- [Muggleton 92] Muggleton, S. and Feng, C.: Efficient Induction of Logic Programs, in Muggleton, S. ed., *Inductive Logic Programming*, pp. 281-298, Academic Press, 1992.
- [Myers 94] Myers, B., Goldstein, J., and Goldberg, M.: Creating Charts by Demonstration, in *CHI'94*, pp. 106-

111, 1994.

- [Pazzani 92] Pazzani, M. and Kibler, D.: The Utility of Knowledge in Inductive Learning, *Machine Learning*, Vol. 9, pp. 57-94, 1992.
- [Quinlan 86] Quinlan, J. R.: Induction of Decision Trees, *Machine Learning*, Vol. 1, pp. 81-106, 1986.
- [Quinlan 90] Quinlan, J. R.: Learning Logical Definitions from Relations, *Machine Learning*, Vol. 5, pp. 239-266, 1990.
- [Sakagami 97] Sakagami, H., Kamba, T., and Koseki, Y.: Learning Personal preferences on online newspaper articles from user behaviors, in *6th International World Wide Web Conference*, pp. 291-300, 1997.
- [Sakagami 98] Sakagami, H., Kamba, T., Sugiura, A., and Koseki, Y.: Effective Personalization of Push-type Systems—Visualizing Information Freshness, in *7th International World Wide Web Conference*, pp. 53-63, 1998.
- [Salton 94] Salton, G., Allan, J., Buckley, C., and Singhal, A.: Automatic analysis, theme generation, and summarization of machine-readable texts, *Science*, Vol. 264, pp. 1421-1426, 1994.
- [Schlimmer 93] Schlimmer, J. C. and Hermens, L. A.: Software Agents: Completing Patterns and Constructing User Interfaces, *Artificial Intelligence Research*, Vol. 1, pp. 61-89, 1993.
- [Shapiro 83] Shapiro, E.: *Algorithmic Program Debugging*, MIT Press, 1983.
- [吉田 95] Yoshida, K., Motoda, H., and Indurkha, N.: 類型パターンの抽出に基づく帰納的学習と演繹的学習の統合, *人工知能学会誌*, Vol. 10, No. 1, pp. 61-71, 1995.
- [Yoshida 96] Yoshida, K. and Motoda, H.: Automated User Modeling for Intelligent Interface, *International Journal of Human Computer Interaction*, Vol. 8, No. 3, pp. 237-258, 1996.

## —— 著 者 紹 介 ——



吉田 健一(正会員)

1980年東京工業大学卒業。同年、(株)日立製作所に入社。同社エネルギー研究所、1986年より基礎研究所、1998年よりシステム開発研究所。博士(工学)。1984年日本原子力学会論文賞、1990年電気学会論文賞、1992年度人工知能学会論文賞。