

オブジェクト指向を用いた計算機使用支援のための知識ベースシステム

1B-5

山本 誠 西原 清一
筑波大学 電子・情報工学系

1. はじめに

計算機システムのもつ機能を理解し、自由に使いこなせるようになるためには、ユーザーマニュアル等による自学自習のみでは、かなりの時間と労力が必要となる。このような負担は、その計算機の専門家の援助があれば、軽減することが可能である。そこで計算機自身が専門家(エキスパート)として、利用者に対話することにより、その理解を促進するシステムが有用であると考えられる。本研究では、このようなコンサルテーションを行うシステムUNIX Tutor (UT) について、開発中である。現時点では、UTの基本処理である質問応答処理に焦点を当てて、その一部にオブジェクト指向⁽¹⁾の考え方を導入した開発を行っており、以下では、その手法について述べる。

2. UNIX Tutor (UT)

UTはUNIXを対象とした計算機の使い方を利用者に示す知識ベースシステムであり、その構成は図1のようになる。知識ベースにおける知識の表現法としては、フレームモデル⁽²⁾を採用した。これによりUNIXコマンドの使い方に関する知識をまとめた単位(フレーム)で表すことができ、UTの処理に適した知識の構造化が図られる。利用者はUTに対して質問応答を行うが、質問は基本的に、1個以上のキーワードの並びで構成される。質問応答処理における探索オペレーションの流れの概要を図2に示す。

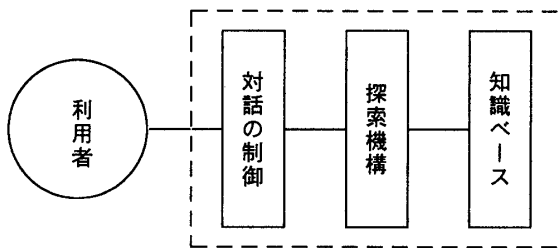


図1. 知識ベースシステム UT の構成

3. オブジェクト指向の導入

UTは、初期の機能に固定せず、使用が進むにつれて、質問応答処理の強化(UTの回答に対する追加質問など)や、その他の機能追加(学習機能など)が起こると想定される。したがって機能拡張が行い易い成長性の高いシステムである必要がある。それらの種々の機能は、質問応答処理に深く

関係してくるので、質問応答処理に対して予め柔軟性をもたせなければならない。

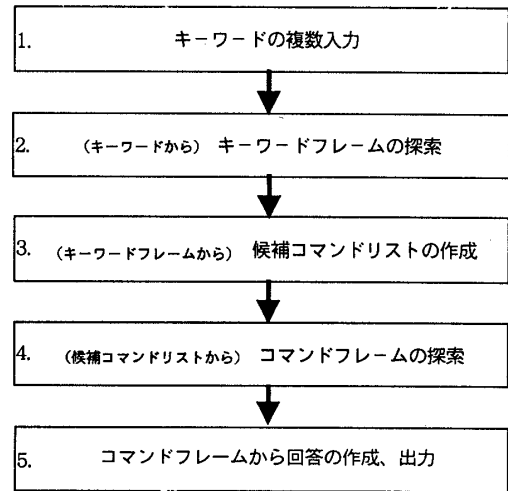


図2. UTの質問応答処理の流れ

利用者はキーワードとして、UNIXコマンドそのものや、キーワードフレームが知識ベースに存在しないものといった様々な入力を行うことが想定される。ここで質問応答処理の目的は、コマンドフレームを求め、回答を提示することにある。そのため質問応答処理は、図2に示した一般的処理過程(1-2-3-4-5)以外の処理も行う必要がある。そのような例外的処理は、ある段階における処理がうまくいかなかった場合に起こる。例外的処理は、一般的処理とは処理過程(各段階の組合せ)を変えて対応する。例えば利用者が、文字列'cp'をキーワードとして入力した場合を考える。cpはUNIXコマンドの1つである。システムは第1段階ではcpをキーワードとして捉える。第2段階でキーワードcpに対する処理において、キーワードフレームの抽出に失敗する。第2段階は失敗したときの処理として、キーワードcpをコマンド候補リストに置き換える。その後処理は第4段階へとぶ。この例の場合、処理過程は1-2-4-5となる。質問応答処理は、例外的処理も含めて5つの段階の組合せによってなされる。

このような処理を従来の手続き型プログラムによって記述したならば、上述した機能付加を行うたびに、処理を全体

に渡って修正しなければならず、UTの場合適したものとは言えない。これを避けるため、つまり高いモジュラリティを得るため、以下の手法を取り入れる。

各段階に共通して言えることは、各段階における知識を処理した結果、次の段階への新たな知識が得られるということである。そこで知識そのものを主体とし、知識にそれを操作するオペレーションを付加するのが妥当であると考え、オブジェクト指向を取り入れる。これにより、各段階の処理は知識、オペレーション共に独立してカプセル化がなされる。このようにUTの探索機構にオブジェクト指向を導入することは、保守、成長性からみて非常に有効な手段であると考えられる。なお、ここでいうオブジェクト、メッセージ、クラス、継承等のオブジェクト指向の概念は、Smalltalk - 80の定義を基にしている。

4. 探索機構へのオブジェクトの導入

オブジェクトを導入するためにまず、知識の分類を行う。図2を見ると、各処理段階によって扱う知識が異なることがわかる。これに沿って知識は次のように分類できる：1. キーワード、2. キーワードフレーム、3. 候補コマンドリスト、4. コマンドフレーム。探索オペレーションをその操作対象となるこれらの知識ごとに分割し、得られた（知識、分割オペレーション）の各組をオブジェクトとして構成する。オブジェクトは、知識の種類に応じてクラスと呼ばれる型を持ち、オブジェクトを操作するオペレーションは、そのオブジェクトの属するクラスに定義する。基本的にクラスは、上述した各知識に対応させて設定する。

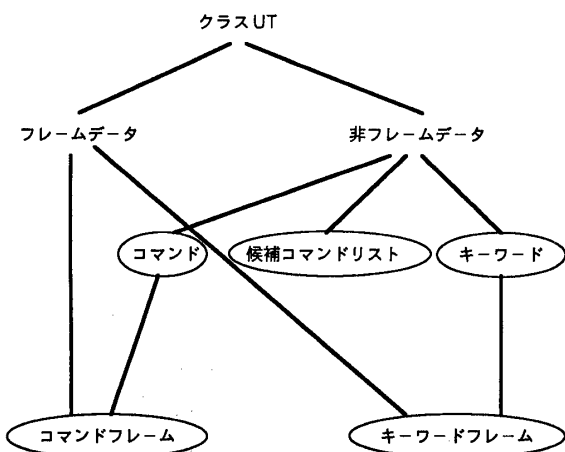


図3. UTにおけるクラスとその階層構造

次にクラスの階層と性質の継承（インヘリタンス）を導入する。UTにおけるクラスとその階層構造は、継承を考慮して、図3に示すように決める。さきに知識の種類に応じて設定したクラスの他に、各知識の共通項を表す3つの抽象クラス（クラスUT、フレームデータ、非フレームデータ）を設定する。抽象クラスには、下位クラスに共通する性質（オペレーション）を定義することにより、各下位クラスに同じ性

質を再度、定義する必要はない。これは継承によるものであり、結果として探索処理の記述の手間を減らすことが可能となる。なお図3に示すように、継承については、多重継承を含むものとした。

5. 探索の制御

オブジェクト指向は探索過程にのみ取り入れた。つまり、知識ベースにははじめから各知識がオブジェクトとして入っていない。探索過程において知識ベースから知識を抽出する時点で、オブジェクトとして生成される。

探索の制御はオブジェクトを導入することにより、メッセージパッシングを基本とする。作業領域（WM）を設定し、そこには知識に対応するオブジェクトが入る。WMにメッセージを送ることによりそのオブジェクトを操作するオペレーションが起動され、次の段階の知識が得られWMに入る。メッセージはつねにWMに送ることにより、探索処理を行うものとし、探索機構が簡潔な機構となるようにした。

キーワードとして文字列'look file'を入力したときの例を図4に示す

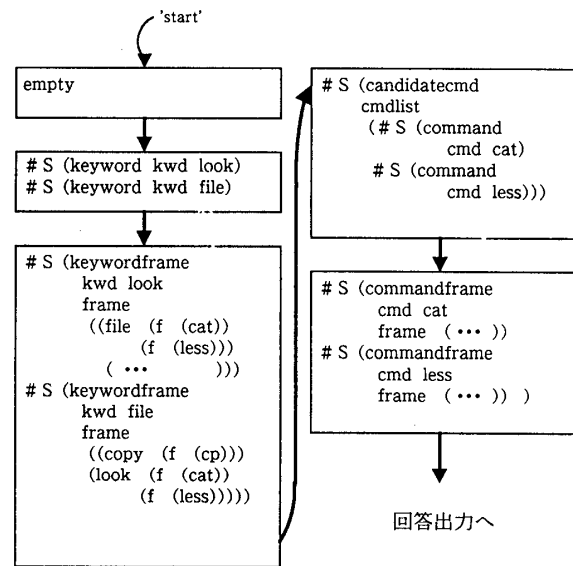


図4. 作業領域（WM）の変移による探索処理例

6. まとめ

UNIX Tutorが機能拡張が容易になるように、その核となる質問応答処理にオブジェクト指向の考え方を導入した。現在、Common Lispにオブジェクト指向を取り入れたPortable CommonLoopsを用いて、その探索機構の開発にあたっている。

参考文献

- (1) A. Goldberg, D. Robson : Smalltalk - 80 *The Language and its Implementation*, Addison - Wesley, 1983.
- (2) M. Minsky : A Framework for Representing Knowledge, in *The Psychology of Computer Vision* (ed : P. H. Winston), McGraw Hill, 1975.