

HyperCardを用いたPROLOGのマルチメディア インタフェース

2K-11

渋谷 正弘

(北海道工業大学 工学部)

田中 譲

(北海道大学 工学部)

1.はじめに

著者らは、以前にスプレッド・シートにPrologを埋めこみPrologを可視化したシステムを提案した(1)。更に、Prologの親和性を向上させ、画像や音声などのメディアも処理できるようにPrologにHyperCardインタフェースを与えたシステムを作成した(2)~(3)。

Prologは、言語仕様も単純で、宣言的なプログラミングができ、再帰プログラムも利用できるといった特徴を持っている。Prologプログラムを可視化し、コピー・ペーストなどの単純な操作でプログラムが作成できると、エンドユーザでも手軽にプログラムが作成できると考えた。そこで、Apple社のHyperCard(4)をPrologのインタフェースとして用いることにより、上記の事柄が可能になると考えた。

このシステムでは、Prologの項をHyperCard上のボタンなどのオブジェクトを用いて表現できるので、ユーザはオブジェクトによって示された項をカード上に配置することでPrologプログラムを作成できる。また、ファクト・データとして、テキストデータ以外に、カード上の絵や音のデータも扱うことができる。

Prologプログラムを実行して得られた結果は、カード上に可視的に表現できる。本システムでは、質問文中に出現する変数名と同じ名前を持つカード上のボタンに対して、質問文を評価して得られた変数の値をそのボタンに対して送ることができる。ユーザは、このボタンにHyperTalkを用いて送られてきた値をもとに任意の処理をするハンドラを記述することにより、質問文の実行結果を利用することができる。また、Prologプログラム中でHyperTalkを実行できるので、グラフィカルなプログラムが容易に作成できるようになった。

2.Prologプログラムの可視化

本システムでは、HyperCardのボタンなどのオブジェクトを利用して、Prologのルールを可視化できる(図1)。図1は、カード上にルール"append"

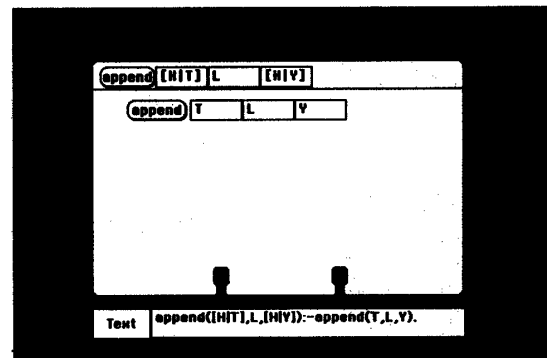


図1 ルールの様子

を示している。項は、"述語名"を示す丸みのある長方形と、"引き数部"を示す長方形より成る。このシステムでは、カードの大きさと述語を表わすボタンとの大きさの関係から、上から順に項を並べてルールを記述する。そのため、このシステムでは、一番上の項を頭部とみなし、それ以下の項を本体とみなす。

カード上に長方形で示された項は、マウス等により編集でき本体の部分にある項の上下の順序を入れ替えることで、ルールの実行順序も容易に変更できる。ユーザは、新規ルールを作成するには、新規ルール・カードを用意しダイアログを開き、述語名と引き数の数を入力し、ルールの頭部を決定する。本体の部分の値を決定するには、ダイアログを用いて新規に項を作成するか、既にデータベースに登録されている述語を参照する辞書から述語をコピーするか、あるいは、別のルール・カード上の項をコピーし、このカード上に貼ることにより決定できる。こうして作成されたルールは、辞書に登録され新たなルールを作成す

Multimedia Interface to The Prolog Programing through The HyperCard System

Masahiro SHIBUYA¹, Yuzuru TANAKA²

1:Hokkaido Institute of Technolgy 2:Hokkaido University

る場合簡単に参照できる。以上、ユーザはマウス等の簡単な操作より新規ルールが作成できる。

3.メディアの活用

本システムでは、Prologプログラム中で、カード上に描かれた絵、リソース中に蓄えられた音のデータが扱える。これらのデータは、ファクト・カードと呼ばれるカード上で保存、処理できる。絵や音のデータは、それぞれ'pict','snd'というデータタイプを持ち、テキストデータとは区別される。図2はファクト

"star(momoko,miyazawa,pict,snd)."

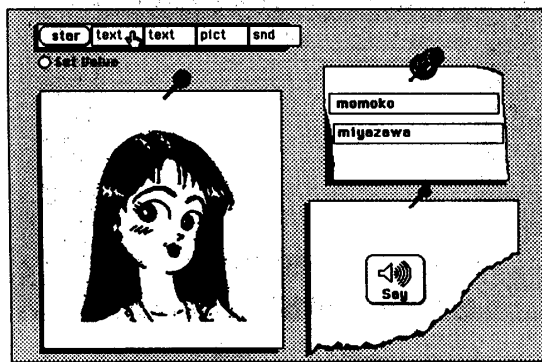


図2 ファクト・カードの様子

を示しており、第3の引き数'pict'は図2の顔写真を、第4の引き数'snd'は図2の音声ボタンを示している。これらの絵や音のデータを利用したいときはPrologプログラム中で、新たに用意したHyperTalkの関数を実行する。

4.質問結果の利用

本システムでは、Prologの質問結果を単にテキストデータとして表示するだけでなくテキストデータをHyperCardに取り込み利用することができる。例えば、ファクト"sound([c,d,e,f,g])"と"playNote(play)"がデータベースに登録してあるとする。この時、質問文

"?-sound(Set),playNote(Play)."

を実行すると答えとして、

"Set = [c,d,e,f,g],Play = play"

が得られる。これらの結果をカード上で利用するために、本システムではカード上のボタン"Set"に"[c,d,e,f,g]",Play"に"play"という内容を送ることができる。これらのボタンには、送られたメッセージの内容をもとにカード上である動作を行うプログラムがHyperTalkで記述してある。ボタン"Set"は、送られてきたリストデータを実譜で表示し、ボタン"Play"は譜面上の音譜を演奏する機能がある。この質問文を実行すると、譜面に音譜が並び演奏を開始する(図3)。

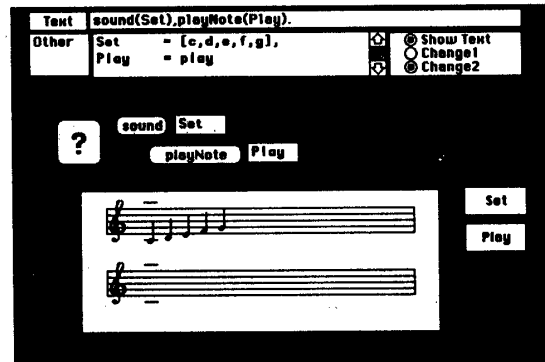


図3 質問結果の利用の様子

以上、質問文中の変数とHyperCardのボタンとの間に等しいという関係を持たすことにより、質問文中の実行結果や質問文中に含まれる情報をカードを利用し可視化できる。

5.まとめ

本論文では、HyperCardを利用して、Prologが可視化できることを示した。本システムを利用するとPrologプログラムはテキストエディタを使用せずに、ダイアログに対する値の代入とボタンのコピー・ペーストのみで作成できる。この方法は、プログラムをグラフィカルに作成・編集ができるようにするので、Prologは親和的になると考える。

また、質問文中に出現する変数とカード上のボタンとの間に等しいという関係を結ぶ事により、実行結果をグラフィカルに表示できることを示した。例えば、リストデータをグラフ表示する機能を持ったボタンや、表形式に表示する機能を持ったボタンなどを用い実行結果を簡単にユーザにとって理解しやすいグラフや表形式で表示することもできる。

参考文献

- [1] 渋谷正弘、田中譲 "スプレッド・シートを介した論理型プログラミング" 情報処理学会論文誌 Vol.30, No.6(1989)
- [2] 渋谷正弘、田中譲 "PROLOGのHyperCardインタフェース" 情報処理学会第40回全国大会(1990.3)
- [3] 渋谷正弘、田中譲 "PROLOGのHyperCardインタフェース" 情報処理学会第41回全国大会(1990.9)
- [4] "HyperCard User's Guide", Apple Computer, Inc. (1987)