1Q−8

エキスパートシステム構築支援ツールKBMS-3
−ユーザインタフェース構築支援−

牛島　浩一　　木村　文宏　　柴山　徹哉
NTT情報通信網研究所

1．はじめに

ＥＳ（エキスパートシステム）を開発する上で最も重要なことは投入する知識の量と質であることは言うまでもない。しかし、実用的なシステムの開発においては、ＵＩ（ユーザインタフェース）の構築にも大きな労力が必要となる。

ＫＢＭＳ[1]では、このようなＵＩ構築を支援するため「ツールキット」と呼ばれる関数ライブラリを提供している。ツールキットは、(1)ポータビリティ、(2)ＵＩ構築の容易性、(3)メモリ管理の簡単化に主眼を置いて開発された。本稿では、このツールキットの基本的な考え方、および特徴について述べる。

2．ツールキットに必要な条件

2．1　ポータビリティ

ＫＢＭＳはMS-DOSをはじめ、X-Window、OS/2などさまざまな環境で稼働している。そのため、これらの環境間のポータビリティをどのようにして実現するかが大きな問題であった。

従来のＫＢＭＳでは知識のポータビリティは実現できていたが、ＵＩに関しては統一的に使用できるものが存在しなかった。そのため、ある環境で作成したＥＳを他の環境に移植する場合には、すべてのＵＩ部を作り直さざるをえなかった。また、ＫＢＭＳ自身のＵＩ部（知識エディタなど）も環境に合わせて複数開発する必要があった。

このような問題を解決するためには、すべての環境で共通に使用できるツールキットが必要となる。

2．2　変更の容易性

一般に、ＥＳを開発する場合には試行錯誤をしながらシステムの完成度を高めて行くことが多い。このような試行錯誤はヒューリスティクスを投入するためには必要不可欠なプロセスである。そのため、ＫＢＭＳでは知識の追加、変更をサポートするための様々な機能が用意されている。

しかし、ＵＩ部は手続き型の言語で記述されているため簡単に作り直すことは難しい。ツールキットには、このような仕様変更に柔軟に対処する機能が必要となる。

2．3　メモリ管理の簡単化

ＵＩのように扱うデータが不定長であり、頻繁にメモリの確保、解放が発生するようなプログラムではメモリ管理が大きな問題になる。無計画にmalloc、freeを行なうと、メモリのフラグメンテーションが発生し、処理が続行できなくなってしまうからである。

ＵＩの簡単な構築のためには、このような問題の発生しないメモリ管理機構を提供する必要がある。

3．ツールキットの特徴

3．1　ＡＰＩの高水準化

一般に、ポータビリティを高めるためには大量のＡＰＩ（Application Program Interface）を定義し、統一する必要がある。しかし、それぞれの環境には、固有のＵＩ構築ライブラリや、ウィンドウシステムがあり、その仕様を無視するわけにはいかない。また、他のアプリケーションとの操作性の統一などの問題もあり、すべての環境のＵＩ仕様を統一するのはかなり困難で

表1　抽出したＵＩ部品

ＵＩ部品		機　能
Window		他のＵＩ部品を貼り付けるシート
Menu		選択操作用の部品
	PullDown	メニューバーからプルダウンして使用する
	PopUp	マウスカーソル位置にポップアップする
Control		スイッチ類
	PushButton	マウスで操作するボタン
	CheckBox	マウスでＯＮ／ＯＦＦするボタン
	RadioButton	複数の中から一つを選択するボタン
	Slider	マウスで操作するスライドボリューム
Screen		文字や絵を表示する枠
	Text	文字を表示する
	TextEditor	文字列を編集する
	Table	表形式で表示、選択する
	Canvas	図形を表示する
	List	一覧表の表示と選択をする
	Network	ネットワークを表示、選択する

An Expert System Building Tool KBMS-3
- User Interface Building Facility -
Kouichi USHIJIMA, Fumihiro KIMURA, Tetsuya SHIBAYAMA
NTT Network Information Systems Laboratories

ある。

そこで、環境に依存すると思われるＵＩの外観および操作（いわゆるLook & Feel）に関してツールキットで規定することをやめ、かわりにもっと上位のＵＩ部品レベルで統一する方向で検討した。ＥＳやＫＢＭＳ知識エディタなどのＵＩを調査した結果、表１に示すようなＵＩ部品を抽出することができた。

ＡＰＩで Look & Feel に関して規定しなくてもよいように、これらのＵＩ部品はすべてユーザの操作に対して自律的に反応するものとして定義している。こうすることによって、ＡＰＩの数を小さく抑えることができ、ポータビリティの実現が容易になっている。

また、プログラムから細かい制御を行なわなくてもよいため、ＵＩ構築が簡単になっている。例えばテキストエディタは、ウィンドウの上に貼り付けるだけでただちに文字列の編集が可能になる。プログラムからは編集された文字列の取り出しをいつ行なうかを指示するだけでよい。このようにＵＩ部品のＡＰＩは一般的なツールキットに較べて高水準なものになっている。

３．２　プログラムとデータの分離

ＵＩの作成においては、画面のレイアウトや、表示するメッセージなどの変更が頻繁に発生する傾向にある。これらの変更のたびに、プログラムをリコンパイルしていては開発効率は上がらない。

そこで、変更の可能性のあるＵＩ関連のデータをプログラムからできるかぎり分離し、「リソース」と呼ばれる別ファイルで管理することにした。

リソースは、ウィンドウの形状、ＵＩ部品の配置情報、エラーメッセージなどの文字列、使用する色、表示する図形、使用するフォントなどの情報を含む、一種のデータベースになっており、ＫＢＭＳは実行時にこのデータベースを参照しながら動作する（図１）。

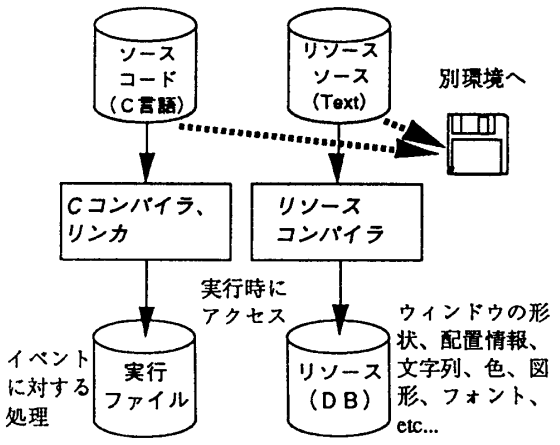

図１　プログラムとデータの分離

このようにデータとプログラムを完全に分離することで、簡単な修正作業がプログラムと独立に行なえるようになり、開発効率を上げることができる。

また、リソースはアクセス速度を上げるために機種依存のファイル構造を持っているが、「リソースソース」と呼ばれるテキスト形式ファイルを経由することで別環境へのポータビリティを実現している。リソースソースは各環境のリソースコンパイラでデータベースに変換して使用する。

３．３　ウィンドウアプリケーションに特化したメモリ管理機能

ウィンドウを使ったＵＩでは、メモリをウィンドウの作成と同時に確保し、ウィンドウの削除と同時に解放することが多い。また、ユーザが操作している対象のウィンドウ番号から、それに対応した作業領域を探すという処理が頻繁に発生する。ツールキットでは、このようなウィンドウアプリケーションに特化したメモリ管理機能を提供している。

この機能では、(1)ウィンドウ毎にサイズを指定してメモリを確保することができ、(2)ウィンドウ番号から簡単に参照でき、(3)フラグメンテーションを抑えるためのGarbage Collection の機能を持ち、(4)ウィンドウの削除と同時に解放されるためメモリ解放のタイミングを気にする必要がなくなる。

また、ウィンドウに貼り付けて使用するＵＩ部品はそれぞれ単体でメモリ管理機能を持っているため、文字列などの不定長のデータはテキストエディタやテーブルに格納すれば、メモリ管理の必要はなくなる。

これらの機能により、ＵＩ構築の労力を大幅に削減することができる。

４．おわりに

ＫＢＭＳのユーザインタフェース構築支援機能について説明した。高水準のＵＩ部品を定義しポータビリティの向上を狙っている。また、リソースの導入やメモリ管理の自動化により、開発の容易さとともに仕様の変更にも強くなっている。このＵＩ機能は、実際にＫＢＭＳ知識エディタの開発に使用されている。

参考文献

[1]服部他，“KBMS における ES 開発支援機能”，情報処理学会第39回全国大会，3B-6..9，1989.