SUIT: ソフトウェア・ユーザーインターフェース設計ツール その2: 機能と特徴

5G−5

来住伸子, 鈴木美幸, 伊藤圭一, 平野一路
日本アイ・ビー・エム株式会社

SUITは, 仕様レビュー段階で使用することを主目的としたグラフィカル・ユーザーインタフェース(GUI)設計ツールである. ここでは, SUITの機能と特徴を操作構造設計部分を中心に説明する.

1 SUITの開発環境

現在のSUITの主目的は, OS/2^{TM} Presentation Manager上のアプリケーションが提供する, SAA/CUAに準拠したユーザーインタフェース仕様をレビューする作業を効率化することである. そこで, SUIT自身も, OS/2 Presentation Managerのアプリケーションとして開発した. さらに, オブジェクト指向なユーザーインターフェースの記述が可能なこと, 画面設計ツールがすでに存在するなどの理由から, Interactive Images社のEASELという開発システムを利用する事にした.

EASELは, GUIだけでなく, 通信機能などもサポートするアプリケーション開発システムで, EASEL言語で記述されたプログラムの実行環境と, EASEL言語のプログラムを出力する画面設計エディタなどの開発環境とで構成される. SUITの操作構造エディタは, EASELの画面設計エディタの出力したEASELプログラムを読み込み, 操作構造の作成変更をした後, 再びEASELプログラムを出力する. 出力されたEASELプログラムは, コンパイルするとユーザーインターフェースのプロトタイプとして使用できる.

GUIをオブジェクト指向システムとして作成することが多くなりつつある. その背景には, マルチプロセス環境のもとでの処理の軽減や, オブジェクト指向言語の普及など色々な理由がある. ここで, GUIをオブジェクト指向システムとして取り扱うとは, ウインドウ, ボタンといったユーザーが操作できるユーザーインタフェース要素を, オブジェクトとして扱い, ユーザー入力の処理は各オブジェクトのメソッドとして記述していくことを意味している. 記述言語としては, Smalltalk, C++, Objective Cなどのオブジェクト指向言語などが使用されることが多い. 今回, 我々が使用したEASEL言語は, SAA/CUA準拠のユーザーインターフェース要素を, 言語仕様として組み込みのオブジェクトとして提供している点を特徴の一つとしている. EASEL言語のこの特徴は, EASELプログラムの構文解析によってオブジェクト間の関係を抽出する事を容易にするので, ユーザーインターフェースの操作構造の視覚化に非常に有用であった.

SUIT: Software User Interface generation Tool — Part II. Its Functions and Features
Nobuko Kishi, Yoshiyuki Suzuki, Keiichi Itoh, Kazumichi Hirano
IBM Japan, Ltd.

2 ユーザーインターフェース設計ツールとしての特徴

SUITの設計ツールとしての特徴は, ユーザーインターフェースの視覚化と直接操作を可能にする点である. 前述したように, SUITはEASEL言語のプログラムとしてユーザーインターフェースを生成するので, 通常はSUITの使用者はEASEL言語プログラムを編集する必要は無い. 個々のウインドウの画面設計や, 一つのウィンドウ上でのユーザー操作については, 画面設計エディタによって設計する. 複数のウインドウ間にわたる操作については, 操作構造設計エディタによって設計する. SUITの操作構造エディタは, 以下で述べるCUAアプリケーションの操作の特徴を利用して, 操作構造の視覚化と直接操作を実現している.

2.1 CUAアプリケーションの操作の特徴

CUAで採用されているユーザーインターフェース設計指針のうち, 操作の手順として関係するものは, オブジェクト・アクション方式と呼ばれるもので, 計算機システムとユーザーの対話を,

1. ユーザーが対象物(オブジェクト)を選択する.
2. ユーザーがそれに対する操作(アクション)を選択する.
3. システムが選ばれた操作を選ばれた対象物に対して実行する.

の順序で繰り返すことを原則としている. ただし, マウスからのダブルクリックによって, 対象物の選択と

デフォルトの操作の選択と実行を一度に行うといった省略形や, その他の変形も可能である.

3のステップでは, あらゆる計算処理が考えられるが, ユーザーインターフェースに大きく影響を与える物としては, 対象物を開く (open), 対象物を閉じる (close) の二つが考えられる. 例えば, ウインドウを開く, ウインドウを閉じるといった操作である.

2.2 CUA アプリケーションの視覚化と直接操作

SUIT では, 操作手順の全体像を視覚化するために, EASEL で記述されたオブジェクト指向のプログラムから, 対象物を開く処理, 対象物を閉じる処理にあたる情報を抽出し, ノードとフローで表現することにした. ウインドウ, ダイアローグボックス, メッセージボックスといった, ある一定の種類のユーザーインターフェース要素に対応するオブジェクトをノードとする. これらのユーザーインターフェース要素は画面を書き換える面積が大きいので, ユーザーインターフェースに与える影響が大きいと考えられるためである. また, ノードに対応するユーザーインターフェース要素の画面表示状態や選択可能状態に影響を与えるような, イベントやコマンド列をフローとする. たとえば, ノード A がノード B を起動するイベントをノード B に送り, ノード B が自分自身を画面に表示したら, ノード A からノード B まで open フローがあるとする. ノード B が自分自身を画面から消去し, 自分を呼びだしたノード A に制御を移したら, ノード B からノード A に close フローがあるとする. そして, ノードをアイコン, フローを矢印として, 画面上に表示することにした (図 1).

ユーザーインターフェース設計を変更するときに, プログラムを EASEL 言語によって変更する負担を軽減するために, SUIT ではノードとフローを直接操作できるようにした. つまり, 画面上に表示されたアイコンや矢印を編集すると, それに対応するように, ノードやフローを変更し, さらに, 対応する EASEL プログラムテキストも自動的に変更される.

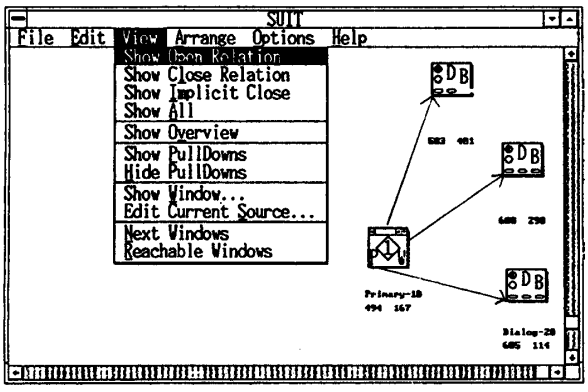

図 1: SUIT 操作構造エディタの画面例

3 仕様レビューツールとしての特徴

SUIT はユーザーインタフェース仕様レビューを次のような3つの側面からサポートをしている.

1. 画面設計の段階
2. 操作構造設計の段階
3. プロトタイプ実行の段階

1 の部分は EASEL 開発システムの画面設計エディタの機能をそのまま使用している. キーの使い方やメニュー項目に使用するテキスト表現など, 一個のウィンドウ内で検査できる項目を自動的にある程度チェックする.

2 の段階では, 操作手順についての問題点, たとえば, ユーザーが操作をキャンセルしたときの戻り先の妥当性などをチェックする. また, 操作構造の全体像を眺めることによって, ユーザーがある機能を使用するのにどれ位の操作が必要か, ユーザーのタスク構造と操作構造に一貫性があるか, 処理の取消をするときの戻り先は妥当なものかどうか, というチェックが効率的にできる.

3 の段階では, 生成されたプロトタイプを動かしてチェックする. 最終製品に近い形のプロトタイプの場合, 実際のユーザーになりそうな人たちにも使ってもらうことが可能なため, 設計者や仕様レビューをする者が気づかなかった問題点を発見できる可能性が非常に高くなる. また, プロトタイプを使用したときの操作を記録しておく機能もあるので, 操作の手順のチェックをより正確に行うことができる.

4 今後の方向

現在, SUIT のユーザーインターフェース設計ツールとしての機能の実現を終えた段階であり, 仕様レビューツールとしての機能の詳細および効果については, 次の機会に報告したい.

OS/2 は, IBM Corp.(米国) の商標です.