

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1: 目的と概念

5 G-4

佐藤和夫, 来住伸子, 鈴木美幸

日本アイ・ビー・エム株式会社

1 はじめに

近年の、コンピュータの性能の劇的な向上により、人とコンピュータの間の、ユーザーインターフェースの大幅な改良が可能になってきている。改良のための手段として、

- ウィンドウやメニューを駆使したグラフィカル・ユーザーインターフェース (以下 GUI と略す) の採用
- 統一された操作環境の提供

が広まりつつある。IBMにおいても、1986年より、GUIの設計指針として SAA/CUA (Systems Application Architecture/Common User Access) を提唱してきた。一方、ユーザーインターフェース設計ツールとして、

- EZVU-II
- OS/2TM のダイアログ・マネージャー
- Interactive Images 社の EASEL

などを提供してきた。そこで、仕様レビューを担当する我々の部門でも、過去数年にわたって、ユーザーインターフェース設計ツールを用いたプロトタイピング手法をいくつかの製品の仕様レビューに適用し、顧客満足度向上の観点から著しい効果があることを確認した。このことは、仕様レビューをする者に、レビュー・ツールとして GUI 設計ツールを与える事が、効果的なレビューに重要な役割を持つことを示している。

さらに、仕様レビューをする者の立場から考えると、従来の GUI 設計ツールの機能には、いくつかの問題点があることも確認された。GUI 設計ツールは、機能的には大別して、画面設計機能と操作構造設計機能に分かれる。ここで、画面設計とは、オブジェクトおよびオブジェクトの組合せを使用して、各画面を設計することをさす。操作構造設計とは、アクションおよびアクションの組合せを使用して、ある画面からつぎの画面へ移行する操作構造を設計することをさす。これらの機能のうち、画面設計機能は、従来の GUI 設計ツール

SUIT: Software User Interface generation Tool — Part I. Its Objectives and Concepts.
Kazuo Satoh, Nobuko Kishi, Yoshiyuki Suzuki.
IBM Japan, Ltd.

でも十分な機能が提供されていることが多いが、操作構造設計には、次のような問題点が見つかった。

- 多くのツールでは、操作構造設計機能はCなどの手続き型言語の記述により実現されるが、かなりの学習が必要であり、かつ、操作構造設計の改良のための試行が面倒である。ここで、試行とは、プロトタイプの作成・テスト・レビューの一連の作業の繰返しをさす。
- いくつかのツールでは、操作構造設計は対話型でなされるものがあるが、操作構造の全体像は頭の中に入れる必要があり、ユーザーインターフェースの複雑な製品の操作構造設計の試行はやはり面倒である。

今回、仕様設計者のみならず、仕様レビューをする者にとっても理想的な設計ツールの必要要件を検討した。さらに、必要要件を満足する新しい操作構造設計用エディターを含めた、新しい GUI 設計ツール: SUIT (Software User Interface generation Tool) を試作したのでここに紹介する。

2 仕様レビューにおける設計ツールの理想像

ここで、仕様レビューをする者は SAA/CUA の設計指針を理解しており、SAA/CUA 準拠の製品のレビューを行うとすると、次の必要要件を満足する設計ツールが理想的であるのではないかと考える。

まず、前章で述べた問題点を解決するために、

- 操作構造設計の改良のための試行が、SAA/CUA の操作方法で (手続き型言語の記述なしに) 実現できること。いいかえれば、習得容易であること。
- 操作構造設計・試行の全容が一望のもとに表示可能なこと。
- 操作構造設計・試行が容易なこと。

つぎに、仕様レビューをより詳しく、より正確に、より容易にするために、

- プロトタイピング手法を利用し、ユーザビリティ・テストを実施するために必要な次の機能を提供する。
 - ユーザーの入力に対する応答時間を自由に設定できること。
 - 自動操作記録および自動再実行機能を持つこと。
- 設計が、SAA/CUA 準拠かどうか、自動的に検査できること。

また、他の設計ツールでもすでに実現していることではあるが、

- 画面設計の改良のための試行が、SAA/CUA の操作方法で(手続き型言語の記述なしに)実現できること。 いかえれば、習得容易であること。
- 設計中に画面が WYSIWYG で表示・変更可能なこと。
- 画面設計・試行が容易なこと。

3 SUIT の基本概念

2章で述べた理想像を実現するために、新しい GUI 設計ツール:SUIT を作成した。SUIT の操作構造設計用エディタの設計には、次にのべる四つの基本概念を導入した。

1. オブジェクトの表現にノードを使用すること。

SAA/CUA では、操作可能なオブジェクトとして、ウィンドウ、ダイアログボックス、メッセージボックス などをもつ。これらのオブジェクトはユーザーからの入力によって起動される。これらを、すべて平等に扱い、ネットワークのノードとして扱うことにする。

2. 操作構造設計をフローとノードの結合設計ととらえること。

あるオブジェクトから次のオブジェクトへの移行は、あるノードから別のノードへのフローとして扱う。フローには、オブジェクト指向の考えを用いることとする。すなわち、ノードからノードへの一つの発火条件を一つのフローに記述する。このことにより、ノードからの移行先や移行条件の変更が簡単になる。

3. ノードの結合設計は GUI を用いること。

ノードをアイコンとし、フローを矢印として、ネットワークを視覚化する。ノード、フローの追加・変更・削除はマウスを使用して、「直接操作」で行う。

4. SAA/CUA に準拠した設計環境を提供すること。

SUIT では、画面設計環境は SAA/CUA に準拠する。操作構造設計環境も SAA/CUA に準拠し、オブジェクト・アクション方式を採用している。すなわち、つぎの3ステップを繰り返すことにより、設計・試行を進める。

- (a) オブジェクトの選択
- (b) アクションの選択
- (c) コンピュータからの応答

他方、画面設計用エディタとしては、EASEL を採用し、SUIT の中から自由に呼べるようにした。EASEL は、次の機能を持っている。

- SAA/CUA に準拠した画面設計エディタ
- SAA/CUA に準拠した GUI 製品の設計が可能 (CUA 準拠かどうかの検査ツールを含む)
- 設計された GUI の試行
- WYSIWYG での画面設計・変更

4 予想される効果および今後の計画

現段階の SUIT では、現実の製品に適用した結果が出ていないが、次のような効果があると予想している。

- GUI 設計ツールの習得容易性の向上
- プロトタイプ作成期間の短縮による仕様設計およびレビュー作業の効率化
- 仕様レビュー品質の向上
- 使い勝手に対する顧客満足度の向上
- SAA/CUA の実現性の向上

最後に、今後の拡張として、次のことを計画している。

- 設計に役立つだけでなく、製品の GUI にそのまま利用可能とする。
- CASE ツールとの関係を持つ事により、設計からテストまでの一貫した開発支援ツールとする。