

3K-1

オブジェクト指向アプローチに基づく  
ラピッド・プロトタイピングツール

(株)日立製作所 システム開発研究所 近藤 博文  
 (株)日立製作所 ビジネスシステム開発センタ 遠藤 篤  
 日立西部ソフトウェア(株) 出原 正裕

1. はじめに

実システム開発以前に、実システムの操作環境を模擬し、操作性の事前検証を可能にするプロトタイピングツールの開発を進めている。

多種の入出力デバイスを有する専用端末では、操作者一端末間の会話処理が複雑で、素人には理解しづらいという点から、プロトタイピングツールによる要求定義支援が有効である。文献[1][2]では、専用端末開発以前での検証を目的に、汎用端末を用いた要求定義支援方式、および、要求定義のためのラピッド・プロトタイピング環境の実現手段について述べた。

ビジネスシステムにおけるアプリケーションソフトは、一般に、汎用端末上で実現される。ビジネスシステムを対象とし、汎用端末を用いたプロトタイピングツールを開発するに当たって、仕様記述力の向上が課題となった。本論では、仕様記述力の向上を目的に、オブジェクト指向の導入を検討した。オブジェクト指向を導入した場合のラピッド・プロトタイピングツールについて述べる。

2. プロトタイピングツール開発の動機と発想

実システム開発以前に、プロトタイプを開発し、実システムの操作環境を模擬し、操作性を事前検証することは重要である。種々のプロトタイプを開発するに当たって、我々には、以下の問題意識があった。

- (1) ちょっとしたプロトタイプでさえ、開発に時間がかかる。
- (2) 開発後、必ずと言っていいほど、プロトタイプの手直しが発生する。
- (3) 開発担当者がいなくなると、プロトタイプのメンテナンスが不可能となる。

これに対し、以下の要求に答え得るプロトタイピングツールが欲しいとの願望があった。

- (1) 短期間でプロトタイプが作れる。
- (2) 手直しが簡単にできる。
- (3) 誰にでも、後から手が加えられる。

この願望をかなえるためには、アプリケーションソフトの世界で、単純な要素(パターン)の組合せで複雑なバリエーションを作り出す仕掛けがあればよいのでは、との着想を得た。このような仕掛けを実現するためには、アプリケーションソフトの会話処理を「画面遷移→画面表示」という同一手順の繰返しで記述することが自然であり、判り易い。

以上のような発想に基づき、専用端末向けのプロトタイピングツールを開発した。

3. 専用端末向けプロトタイピングツールの開発と残された課題

専用端末向けプロトタイピングツールの開発に当たって、以下を基本方針とした。

- (1) テーブルウェアへの処理定義による個別処理組込の容易化
- (2) テーブルウェアを画面対応に保有することによる処理定義の簡素化
- (3) デバイス動作を模擬することによる仕様表現のビジュアル化
- (4) 模擬と定義の交互実行による試行錯誤の迅速化

開発したツールをいくつかのプロトタイプ開発に適用した結果、以下の課題があると判明した。

- (1) 複雑な動きが模擬できない。
- (2) キー入力データの取込みが不便である。
- (3) 仕様定義量が膨大になる。

これは、会話処理を「画面遷移→画面表示」という形で定式化したため、画面構築手順は統一できたものの、単純な仕様しか記述できなくなっていることが原因である。

ビジネスシステムにおけるアプリケーションソフトは、一般に、汎用端末上で実現される。ビジネスシステムを対象とし、汎用端末を用いたプロトタイピングツールを開発するに当たって、仕様構築手順の統一という従来のメリットを活かしつつ、複雑な仕様でも簡潔に記述できる仕掛けが必要であるとの結論に到った。

4. 課題解決の基本方針

仕様記述力の向上を目的に、以下を設定した(図1参照)。

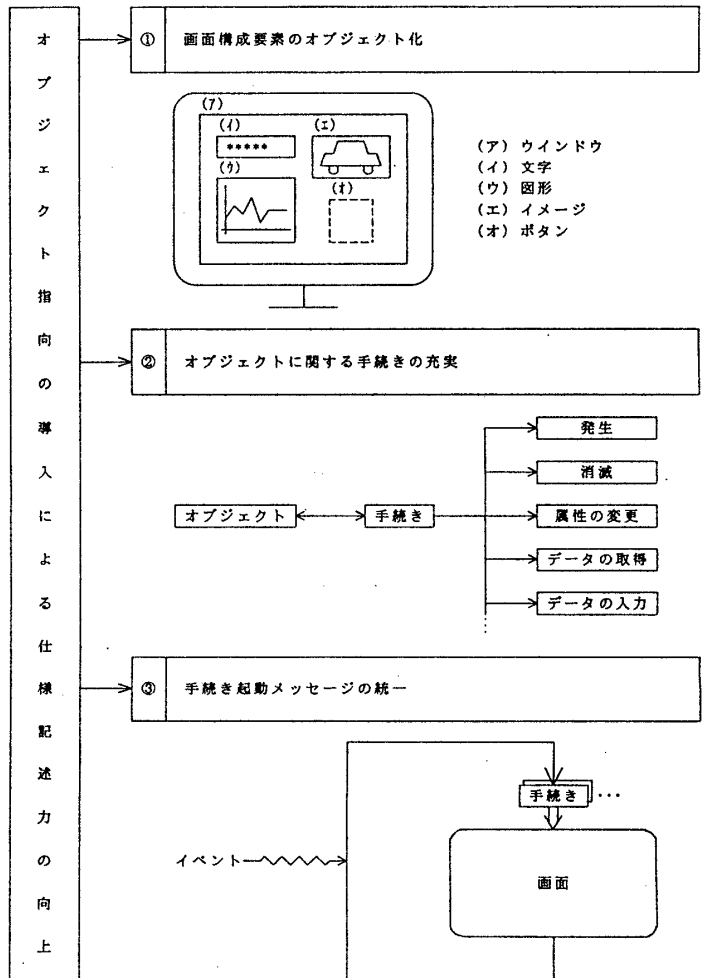


図1 基本方針

(1) 画面構成要素のオブジェクト化

従来のツールでは、画面に表示されたデータは画面遷移の度に消えてしまうので、画面毎に、データの表示領域を新たに定義しなければならず、定義量が膨大になるという問題があった。そこで、画面構成要素である表示領域を、画面間に渡って存在するオブジェクトとして捉えることにした。これにより、画面構成に必要なオブジェクトを一度だけ定義すればよく、何度も同じ領域を定義する必要がなくなる。オブジェクトとしては、以下が考えられる。

- ①ウインドウ：画面を構成する枠。
- ②文字：文字や数値を表示する領域。
- ③図形：罫線、グラフなどを表示する領域。
- ④イメージ：イメージ、動画などを表示する領域。
- ⑤ボタン：マウス入力などのイベントを受付ける領域。

(2) オブジェクトに関する手続きの充実

従来のツールでは、画面表示のみでデバイス動作を表現していたので、単純な動きしか記述できなかった。そこで、オブジェクトの振舞いを記述する手続きを充実させることにした。これにより、複雑な動きでも、表現できるようになる。オブジェクトの手続きは、オブジェクトにより異なるが、一般的には以下が考えられる。

- ①発生：オブジェクトが画面上に発生する。
- ②消滅：オブジェクトが画面上から消滅する。
- ③属性変更：オブジェクトが属性を変更する。
- ④データ取得：オブジェクトがキーボードからデータを取得する。
- ⑤データ出力：オブジェクトが画面上にデータを出力する。
- ⑥閉じる：オブジェクトが一時的に画面上から隠れる。
- ⑦開く：隠れたオブジェクトが再び画面上に現れる。

(3) 手続き起動メッセージの統一

オブジェクト間のメッセージのやり取りは自由であり、手続きを起動するのに何ら制約がないのが一般である。本ツールでは、アプリケーションソフトの会話処理を同

一手順の繰返して記述するため、手続き起動メッセージを画面遷移のトリガーとなるイベントに統一することにした。これにより、仕様構築手順の統一という従来のメリットを活かすことができる。

5. 機能構成

本ツールは、以下の機能で構成する。

(1) 仕様定義機能

画面对応、オブジェクト対応のテーブルウェアにパラメトリックに手続きや演算処理を定義し、簡易な処理定義、個別処理の組込を実現する。オブジェクトとオブジェクト手続きは、以下の流れで定義する(図2参照)。

- ①オブジェクト集合の作成
- ②オブジェクトの画面への登録
- ③手続き内容の定義
- ④手続き順序の定義

(2) 仕様実行機能

テーブルウェアの定義に応じて、「画面遷移→手続きの実行」を繰り返すことにより、仕様をビジュアルに表現する。また、仕様実行/定義の切替制御を行い、実行と定義の交互実行による動的仕様定義変更を可能とする。

(3) テーブルウェア管理機能

上記機能を実現するのに必要なテーブルウェアを共通的に管理する。

6. おわりに

ビジネスシステム向けのプロトタイプングツールを開発するに当たって、仕様記述力の向上を目的に、オブジェクト指向の導入を検討した。本アプローチによれば、仕様構築手順を統一しつつ、複雑な仕様でも簡潔に記述し、模擬実行することが可能となる。

<参考文献>

- [1] 近藤、山下：「モックアップ・ソフトウェア」によるラビッド・プロトタイプング環境の実現，情報処理学会 第41回全国大会講演論文集(5) pp.319-320 (1990)
- [2] 近藤、山下：実環境で稼動する仕様実行部の導入によるラビッド・プロトタイプング，情報処理学会 第42回全国大会講演論文集(5) pp.192-193 (1991)

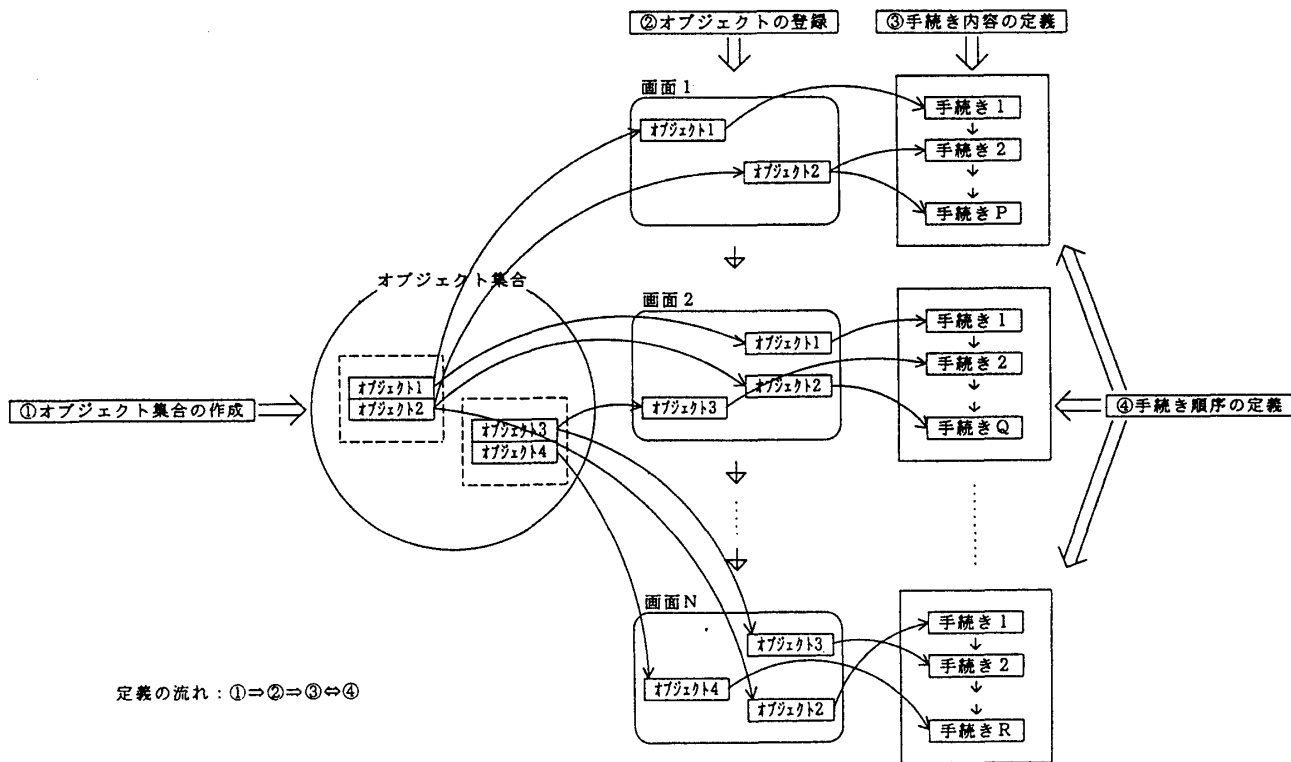


図2 オブジェクトとオブジェクト手続き定義の流れ