

ビジネスコンピュータにおける

アプリケーション開発支援ツールの開発*

7J-5

安田 智, 藤原聡子, 中村 敏, 萱野重実, 田村 順, 高山茂伸, 伊藤正裕, 杉本直也, 鈴木寿明†
三菱電機(株)情報システム研究所†

1 はじめに

今回、我々は当社ビジネスコンピュータ用アプリケーション開発支援環境 HYPERPRODUCE/SV[†]を開発した。HYPERPRODUCE/SVでは、帳表、画面、ファイルなどの入出力部品(以下I/O部品と記す)の設計機能に加え、プログラムの処理手続き設計を支援するプログラム設計機能をサポートしている。

本稿では、このHYPERPRODUCE/SVにおけるプログラム設計機能について、その特徴と実現方式を報告する。

2 開発の目的

業務アプリケーションの開発において、最も多くの人手と時間が費やされてきたのは処理手続きのプログラミング作業である。そこで、今回著者らは、この処理手続きの設計作業を効率化し、アプリケーションの生産性を向上させるための支援機能として、プログラム設計機能の開発を行なった。

3 プログラム設計機能の特徴

HYPERPRODUCE/SVのプログラム設計機能の主な特徴を以下に挙げる。

●日本語名称の使用

プログラム中で使用する項目名には日本語が使用でき、わかりやすい名称で項目を扱える。

●日本語表現を用いた設計

処理手続きの設計は日本語表現を用いて行なうため、プログラミング言語を意識せずに設計できる。また、穴埋め式の作業が主体であり、詳細な文法知識を必要としない。

●選択操作中心の設計

処理手続きの設計は、その殆どをマウスによる選択操作によって行なう。項目やI/O部品の指定は、全て部品イメージまたは一覧からの選択転記で行

なうため、タイピング誤りなどのケアレスミスが発生しない。

●デバッグレス

設計された処理手続きの正当性をチェックする機能により、設計作業の完了と同時にプログラム全体を通じて論理的正当性が保証される。

●ソースコードの自動生成

アプリケーションの完全なソースを自動生成することができる。生成するソースは、当社ビジネスコンピュータ用業務系アプリケーション開発で広く使用されている言語プログレスIIである。

●プログラムとドキュメントの一体化

ドキュメントとソースコードは、同一情報(プログラム設計機能が入力する情報)を入力情報として生成されるため、ドキュメントと実際のソースコードが常に一致する。

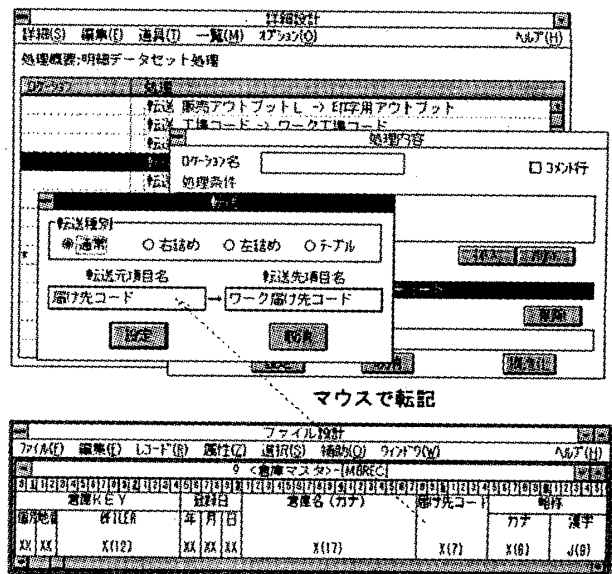


図1 プログラム設計の画面

* Development of Visual Application Design Tool for Business Computer

† satoshi YASUDA, satoko FUJIWARA, satoshi NAKAMURA, shigemi KAYANO, jun TAMURA, sigenobu TAKAYAMA, masahiro ITOH, naoya SUGIMOTO, toshiaki SUZUKI

† Mitsubishi Electric Corporation

4 実現方式

4.1 構成

HYPERPRODUCE/SVは、ビジネスコンピュータ(サーバ)とパソコンとで機能分散した形で実現している。I/O部品設計機能、プログラム設計機能、ソースコード生成機能等の設計系機能をパソコン側機能とし、コンパイル～実行、ドキュメント印刷などをサーバ側の機能としている。また、サーバにはプログラムやI/O部品の設計情報の保管庫としての役割も持たせている。パソコン側の機能は、Windows™²上で動作するアプリケーションとして開発し、Windows™のGUI³操作環境を設計系機能の設計インタフェースに取り入れることにより、操作性の向上を図っている。

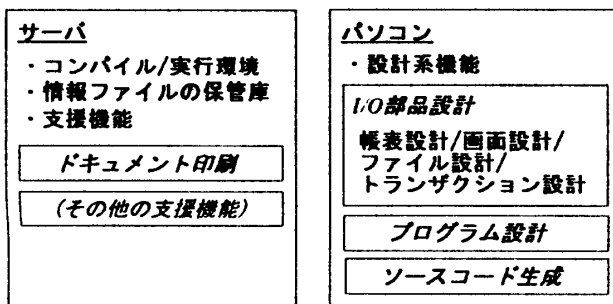


図2 サーバとパソコンの機能分担

4.2 プログラム設計の構成

プログラム設計は、5つの画面より構成されている。画面構成と設計ステップを図3に示す。

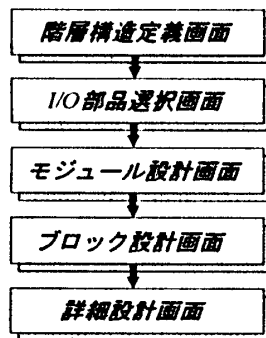


図3 プログラム設計の構成

4.3 プログラム設計の実現方式

●プログラムの階層化

アプリケーションの構築は、プログラムを階層的に組み合わせ、プログラム間の関係定義(呼び呼

ばれの定義)を行なうことで実現する。プログラムを部品的に扱うことで、流用性が高まり、既存プログラムの再利用による効率的開発が可能になる(階層構造定義画面)。

●処理フローのテンプレート化

各プログラムは、処理フローテンプレートを使用して概要設計する。処理フローテンプレートは、プログラムのスタイルをモデル化するために、処理フローのボタンをテンプレート化したものであり、今回は6種類を用意した。プログラムを処理毎に分割したブロックをこのフローにはめ込むことでプログラムの大まかな流れを設計する(モジュール設計画面)。

●設計作業の階層化

プログラムの設計は、モジュール設計画面以下の3画面で、次の様なステップで階層的に行なう。

1) モジュール設計画面

プログラムを処理ブロックに分割し、概要設計(モデル化)を行なう。

2) ブロック設計画面

各処理ブロック毎に、ブロック内の実際の処理手続きを設計する。具体的には、処理手続きのステップを日本語で記述する作業である。

3) 詳細設計画面

1つ1つの処理手続きの内容を命令レベルの記述に展開する。命令の記述は、各命令毎に用意された処理命令指定用ダイアログの穴埋め作業によって行なう。

5 おわりに

本稿では、HYPERPRODUCE/SVのプログラム設計機能について報告した。現在、性能/効果についての評価作業を継続中であるが、現時点では、このプログラム設計機能を使用した場合、従来の人手による開発の約3/4の工数でアプリケーションの作成が可能であるというデータが得られている。

今後の課題として、既存資産の移行機能やサーバ側機能のパソコン側への移行、テスト支援機能などを考察していく必要があると考えている。また、操作性/機能等については、フィールドの要求を反映した改良を加えていく予定である。

*1 HYPERPRODUCE/SV: HYPERPRODUCE for Server

*2 Windows™は米国Microsoft Corporationの商標

*3 GUI: Graphical User Interface