

ビジネス系アプリケーション開発実行環境 —HYPERPRODUCE II / 基本部—

5 S - 2

菅野幹人 森山令子 斎藤俊司 大江信宏

三菱電機株式会社 情報通信システム開発センター

1. はじめに

HYPERPRODUCE II（以下HP II）は、基幹業務アプリケーションを効率よく開発する環境を提供するミドルウェアである。Windows NT^{*1}をベースとしたオープンシステム上で、オフィスサーバによる基幹業務システムの構築法を活かした設計手法と、Visual Basic^{*1}（以下VB）を利用したクライアント/サーバ連携技術、帳票出力技術を採用し、ソフトウェア生産性の大幅向上を実現する。HP IIは、①アプリケーション作成に必要なビジネスOCX機能、②システム要件に応じてダイナミックに変更されるパラメータ部分をビジネス部品情報として設計する機能、③これらの設計情報を解析し、ソースプログラムを自動生成するウィザード機能を特長とする。本報告では、HP IIの概要と特長、通信機能について報告する。なお帳票機能については別セッションで報告する。

2. 特長

図-1に、HP IIにおけるC/S型業務モデルと支援機能を示す。網掛けの部分がHP IIによる支援部分であり、白抜きの部分がSEのプログラミング部分である。支援機能はVB上の部品群と帳票支援機能、通信支援機能、デバッグ機能から構成される。

①部品機能 RPC 通信機能、帳票出力機能、ビジネス入出力機能を32ビットOCXとして提供している。これらはVBから属性を設定するだけ

で通信や帳票出力、GUIによる入出力チェックを行う高機能なオブジェクトである。

②各種設計機能 通信OCXではC/S間の通信データを規定するために、通信データ設計を行う。通信OCXは通信データ設計情報を参照して動作する。通信情報は通信ロジックとは切り離されているのでカスタマイズ作業時に効果を発揮する。また、帳票OCXは帳票を実際に生成する帳票サーバ機能と連携を取り動作する機能で、帳票サーバに帳票に出力すべきデータを転送する。帳票サーバは、帳票OCXから送られてきたデータと帳票設計情報を参照し、高速に帳票を出力する。

③ウィザード機能 通信機能及び帳票機能のビジネスOCXをVBのロジックで駆動するためには、VBのコーディングを行う必要があるが、これらは各種情報ファイルからすべて自動生成される。従って、通信情報や帳票情報を設計するだけで、必要なロジックを生成できる点で優れている。また、各種設計情報はドキュメント出力機能により、仕様書として出力することが可能で、SEの生産性に寄与している。

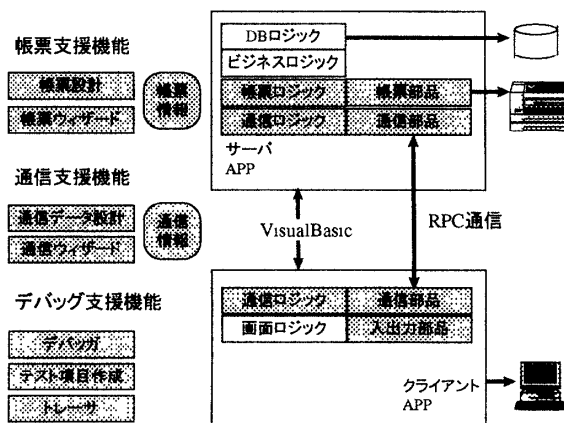


図-1 HP IIにおける支援機能

*1 米国マイクロソフト社の登録商標

An environment of executing and developing software for a business computer system
Mikihito Kanno

MITSUBISHI Electric Corporation

5-1-1 Ofuna, Kamakura, Kanagawa 247, Japan

3. 通信実現方式

図-2, 図-3 はサーバプログラムとクライアントプログラムの通信方法を示したものである。クライアント側プログラム, サーバ側プログラムともにパソコン上のプログラムであり, 通信部品の機能によって通信を行う。プログラム間の通信フォーマットは, 通信データ設計機能で設計する。通信データ設計機能で定義する通信情報は, データ項目名, データ型, データ長などであり, これらをリクエストコードごとに規定する。通信情報は通信部品によって自動的に読み込まれ, 通信時の通信データの送受信や整合性チェックなどの情報として使用される。

```
Rpc1.ProcNum = "00000001" 'リクエストコード設定
Rpc1.Buffer(0) = Scode    '上りメッセージ構築
Rpc1.Call = True         'RPC Call
Sname = Rpc1.Buffer(0)   '下りメッセージ出力
Tanka = Rpc1.Buffer(1)
Zaiko = Rpc1.Buffer(2)
```

図-2 クライアント側プログラム

```
Sub RPCS_EVENT(ProcNum as String)
If ProcNum = "00000001" Then
Scode = Rpcs1.Buffer(0) 'メッセージ取得
'データベースアクセス
Rpcs1.Buffer(0) = Sname '下りメッセージ構築
Rpcs1.Buffer(1) = Tanka
Rpcs1.Buffer(2) = Zaiko
End if
End Sub
```

図-3 サーバ側プログラム

図-2 において, クライアント側通信部品に対してリクエストコード (ProcNum) の設定を行うことにより, 通信情報ファイルからリクエストコードに該当するメッセージ形式を読み込む。通信部品の Buffer は通信フォーマットに従って項目ごとに動的に確保されるメモリエリアであり, Buffer の内容が通信内容となる。クライアント側で Call が行われると, 図-3 のサーバ側の通信部品において通信データが到着したことを知らせるイベントが駆動され, リクエストコードご

とに記述された処理が動作する。通常はデータベースアクセスルーチンが記述されるが, この部分は現状 VB の機能を利用して, データベース処理を行う。

4. 通信ウィザード

図-2 や図-3 のようなロジックを手で記述するのでは生産性は向上できないので, サーバ側, クライアント側用の通信ロジックを自動生成するウィザードを備えている。通信ウィザードは以下のような機能を持ち, 数百行の VB ロジックを一瞬にして生成することができる。

- ① RPC 通信のリクエストコードごとにエラー処理を含めた通信手続きを生成する。
- ② GUI 部品の種類を自動的に認識し, GUI 部品に応じて対応したコードを生成する。
- ③ 通信手続きを GUI 部品のイベントへ割り付けることができる。

ウィザードの方式として項目名のマッチングのルールを使用しており, GUI部品, 通信の項目名によって項目の割り当てを行っている。また, この際, グリッドのような配列型のGUI部品にも対応できるように考慮している。

5. 効果

プログラム仕様の一環として設計する通信情報から必要なロジックを自動生成することにより, SEの必須工程とプログラミング工程を一体化することが可能になり, 通信部分については仕様定義のみで複雑な通信処理を行うことが可能となった。このようにプログラム作成上必要な工程を支援し, かつ, プログラムの自動生成に結び付ける方法は, ビジネス系システム構築に携わるSEに対しては有効な方法である。

6. まとめ

通信系の OLE 対応などが今後の課題であり, 通信部品などから OLE Automation によるアプリケーション連携などを検討していく。