

## ビジネス系アプリケーションにおける通信手続き自動生成

3S-1

菅野幹人 石井篤 森山令子 吉村礼子 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

### 1. はじめに

パソコンをクライアント、ビジネス系の計算機をサーバとするクライアントサーバ型のアプリケーションシステムにおいて、ソフトウェア開発の下流工程を支援するための開発実行環境を提供している。本開発実行環境により、パソコンの持つGUI(Graphical User Interface)やマルチメディア機能と、サーバの持つ基幹業務処理機能を組み合わせ両者の特徴を合わせ持つシステムを効率よく構築することができる。

クライアントサーバ型の2つに分割されたプログラムを作成する上で核となる機能は、通信部分の通信フォーマットを定義するトランザクション設計機能と、それを参照して通信を行う通信部品である。本稿では、クライアント側のプログラムにおける通信方式について説明するとともに、今回新しく開発した通信手続き自動生成について方式と評価結果を説明する。

### 2. 通信方法

図-1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。サーバ側プログラムは当社独自のビジネス系の簡易言語で記述されている。クライアント側プログラムは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。通信はRPC(Remote Procedure Call)によって行われる。プログラム間のRPC通信の通信フォーマットは、トランザクション設計機能で設計する。トランザクション設計機能で定義されるトランザクションファイル

イルには、RPC通信のリクエストごとに上り、下りのバイト列が規定される。クライアント側プログラムでは通信手続きを記述することにより、通信部品を駆動することができ、これによりサーバプログラムと容易にRPC通信を行うことが可能となる。

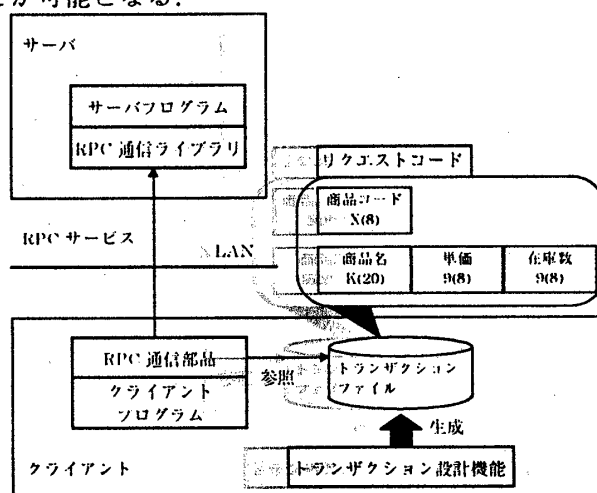


図-1 通信方法

### 3. 通信部品

ビジネス系計算機においてはRPCを用いた通信は一般的でなく、端末エミュレータによるホスト主導型の通信形態が普及している。そのためRPC通信を用いたクライアントサーバ型のアプリケーション開発のためには、通信方法の規約が必要であった。また、生産性が従来の開発方法より極端に悪くなることを防ぐ必要もあった。このような背景により通信部品は通信の標準化と通信ロジックの削減を目的としている。通信部品は以下の特長を持つ。

- データ型の差異やコード体系の変換をトランザクション情報により通信部品で解析する。
- 時間管理を行い通信のリトライを自動で行う。
- 通信部品の中にデバッグ機能の仕組みを持つ。通信部品における通信手続きは図-2の記述により行う。

An automatic generation of communication procedure for a business computer system  
Mikihito Kanno  
MITSUBISHI Electric Corporation  
5-1-1 Ofuna, Kamakura, Kanagawa 247, Japan

```

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 通信手続き

通信部品に対してリクエストコードの設定を行うことにより、トランザクション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。

従来この通信部品を駆動するための通信手続きは、コードを直接記述していた。サンプルプログラムの評価結果から、通信手続きの記述量はクライアント側プログラム全体の記述量の60%から70%になることが判明し、この作業の自動化が生産性向上に結びつく結果が得られた。

#### 4. 通信手続き自動生成

クライアント側プログラムの通信手続きを自動生成するためには、入力情報として、画面構成情報とトランザクションファイルが必要である。これは GUI 画面の部品で保存している内容が通信に使用されることと、通信が行われるタイミングが GUI 部品へのイベント発生時なので、GUI 部品のイベント処理部に通信手続きを作成し、記述する必要があるためである。また、通信手続き自動生成の開発に際し、使用方法を限定し、効果的に運用するために以下の条件を設定し開発を行なった。

- 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。
- ①の条件を満たすために、トランザクションファイルの中に記述されている、リクエスト

コードごとに通信手続き関数を作成する。通信手続きの作成では、GUI部品とトランザクション項目の英数字でのマッチング処理を行い、一致した部品名を上り、下りメッセージの構築に使用する。また通信エラー処理関数も自動生成する。次に②の条件を満たすために、あらかじめGUI部品を限定し、使用するGUI部品に応じて生成するコードを変更する。内部的にはこれらのGUI部品はC++のオブジェクトとして表現し実現している。③の条件を満たすために、GUI部品のイベント一覧と、生成した通信手続き関数一覧をダイアログ上で参照して、イベント内から通信関数を呼び出す手続きを自動生成し挿入する。

#### 5. 評価

通信手続きをコードで直接記述していた従来のクライアント側プログラミングと通信手続き自動生成を使用した場合の生産時間の割合について図-3と図-4に示す。

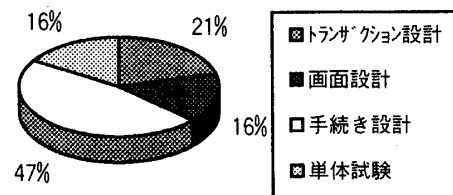


図-3 従来の生産工数

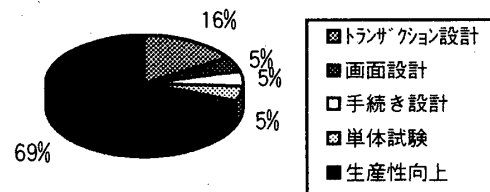


図-4 通信手続き自動化の生産工数

通信の手続きを自動化したことにより、従来より約3倍生産性が向上した。

#### 6. まとめ

通信部品における通信手続きを自動生成することにより、ビジネスアプリケーションにおけるクライアントサーバ型のアプリケーション生産性が向上することを示した。今後さらに機能を強化しプログラムレス化を推進していく。