

分散オブジェクト指向業務システム向け アプリケーションフレームワークの開発

谷村 武洋*、石井 武夫*、土田 修己**、柳 忠弘***、伊東 鎮範***

*日立製作所 ソフトウェア事業部、**日立製作所 ビジネスソリューション開発本部、

***日立ソフトウェアエンジニアリング

1. はじめに

近年、コンポーネント技術を利用したアプリケーションの開発が増えつつある。これは機能別のソフトウェア部品(コンポーネント)を組み合わせてアプリケーションを開発する手法であり、短期間で容易に目的のアプリケーションを構築できるメリットがある。^[1]

しかしその反面、コンポーネントを利用したアプリケーションの開発は、コンポーネント技術を支える通信基盤の規約に従ったプログラミングが必要となり、また使用する各々のコンポーネントの使い方のノウハウの習得期間が必要になるため、コンポーネント技術のメリットを得る足かせとなってしまふ。

そこで上記の問題を回避するため、分散オブジェクト基盤上で動作するアプリケーションの雛形を、業務システムで利用する典型的なコンポーネントの組み合わせ別に抽出し、これをフレームワーク^[2]として提供するアプローチを採用した。

開発したフレームワークは実際の業務システム開発プロジェクトに提供し、効果を確認した。

2. サーバアプリケーションのフレームワーク化

2.1 高負荷業務処理対応のフレームワーク

今回作成したフレームワークは銀行オンラインシステムのような高負荷の業務システム内で使用

されるサーバアプリケーションを想定し、以下の機能を組み込んだ。

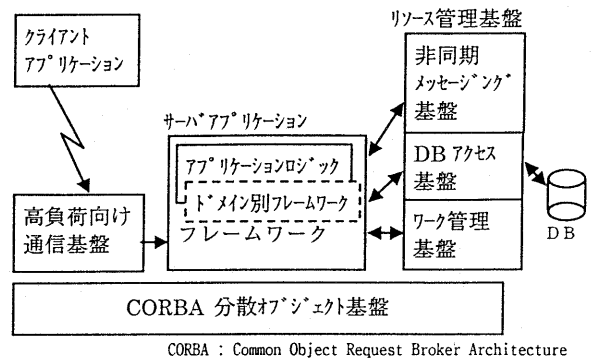


図1. フレームワークの実行時構成

(1) 高負荷向け通信基盤対応

サーバアプリケーションは高負荷向け通信基盤のコンポーネントを介して分散オブジェクト基盤から呼び出される。該コンポーネントでは高負荷に対応するため複数サーバマシン間での処理の負荷分散、サーバアプリケーションのスレッド常駐などの機能を持つ。

このため該コンポーネントに対応するアプリケーションは業務処理の他に業務システムの運用開始・終了時に実行される通信基盤固有の常駐開始・終了処理をプログラミングする必要があるが、この部分をフレームワークで隠蔽した。

(2) リソース管理基盤対応

データベースやワーク管理基盤(ワークフローを管理するコンポーネント)、非同期メッセージング基盤との接続はフレームワークの中でサーバアプリケーションの常駐開始時に予め行っておくことにより業務処理の中でのレスポンスを向上させている。

これらのリソースに対するトランザクション

Development of Distributed Object Oriented Application Framework for Business System.

Takehiro TANIMURA*, Takeo ISHII*, Nobuki TSUCHIDA**
Tadahiro YANAGI***, Shizunori ITO***

* Software Division, Hitachi, Ltd.

** Business Solution Systems Development Division, Hitachi, Ltd.

*** Hitachi Software Engineering Co., Ltd.

制御などの運用環境に依存する情報については、フレームワークの中で集中管理することによりコンポーネントに含まれる業務処理の実装から排除した。

また、これらリソース管理基盤に対しては性能面を考慮し分散オブジェクト基盤経由での呼出しの他、直接呼出しも可能とした。

2. 2 フレームワークによる開発環境

フレームワークを使用した場合、サーバアプリケーションの開発は図2に示す手順で行う。

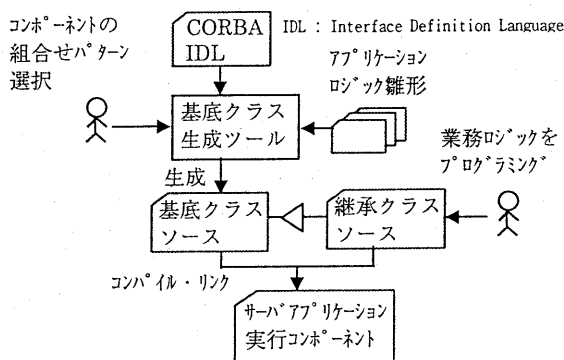


図2. フレームワーク上でのアプリケーション開発方法

- (1) サーバアプリケーション基底クラス生成
 利用するコンポーネントの組合せのパターン毎に用意されたアプリケーションのロジックの雛形から、ユーザが選択したパターンの基底クラスソースを生成する。
 この際、サーバアプリケーションの外部インターフェースは入力されたCORBAのIDL^[3]を参照して生成する。
- (2) 業務ロジックの追加
 生成された基底クラスにアプリケーション固有のロジックを追加する手段としては、オブジェクト指向の開発スタイルに準じてクラスの継承を用いる。

3. 適用効果

今回開発したフレームワークを銀行業務システムの開発プロジェクトに適用した。なお、該プロジェクトでは顧客固有の運用ルールをさらにドメイン別フレームワークとして一元化した。

フレームワークを使用した結果、使用せずに開発した場合に予想される開発工数に比べ表1のような削減効果が得られた。

表1. フレームワークの適用効果

項目	効果	備考
開発規模	約 1/2	数百 kLOC を今回開発
開発工数	約 1/2	
発生不良率	約 1/4	顧客納入後の発生不良件数
開発期間	約 2/3	構造設計～顧客テストの期間

4. まとめ

コンポーネント技術を用いた分散オブジェクトのアプリケーション開発に、フレームワーク化のアプローチが有効であることが確認できた。

なお、今回開発したフレームワークは「Framework - Core Kernel for C++」という名称で製品化している。

参考文献

- [1] 今城 監修, "ビジネスオブジェクト入門", ソフト・リサーチ・センター, 2000
- [2] 田坂 他, "WWW と分散オブジェクトを用いたシステムの構築を支援する WWW 連携分散オブジェクトフレームワーク", 日立評論, pp. 47-50, 1998.5
- [3] OMG, "The Common Object Request Broker : Architecture and Specification", Minor revision 2.3.1, 1999