

# 基幹システムフレームワークによる変化に強い基幹システムの短期構築

## Short-term Construction of Mission-critical Systems

### あらまし

企業の基幹システムは、競争の激化、規制の緩和、多様な顧客ニーズおよび増加する業務量に対応して、分散化、大規模化および複雑化してきている。このような環境において、情報システム部門では、システム構築・運用に関しては、システム開発の効率化・スピード化・安定稼働、柔軟なシステム拡張・変更、TCO削減が最重要課題であると認識している。

本稿では、分散化、大規模化および複雑化していく企業の基幹システムを、柔軟に統合し、変化に強く、かつ短期間に構築するために、Interstage Business Application Server「基幹システムフレームワーク（MC基盤）」が解決する課題および提供する機能の概要について紹介する。

### Abstract

Today's stronger competition, looser regulations, diversification of customer needs, and greater workloads are encouraging enterprises to make their mission-critical systems distributed and increase the scale and complexity of these systems. As a result, system construction and operation requirements such as improved efficiency, faster processing, better system stability and flexibility, and lower TCO have become the most important issues for information system departments. Fujitsu's Interstage Business Application Server (mission-critical system framework) solution for enterprise mission-critical systems reduces development periods and adds greater flexibility to enterprise systems. This paper describes this system.



保西義孝（ほにし よしたか）  
ミドルウェアプラットフォーム事業  
部第三開発部 所属  
現在、Interstage Business Application  
Serverの企画、開発に従事。

ま え が き

企業の基幹システムは、競争の激化、規制の緩和、多様な顧客ニーズおよび増加する業務量に対応して、分散化、大規模化および複雑化してきている。

このような環境において、情報システム部門では、システム構築・運用に関しては、システム開発の効率化・スピード化・安定稼働、柔軟なシステム拡張・変更、TCO削減が最重要課題であると認識している。

本稿では、分散化、大規模化および複雑化していく企業の基幹システムを、柔軟に統合し、変化に強く、かつ短期間に構築するために、Interstage Business Application Server「基幹システムフレームワーク（MC基盤）」が解決する課題および提供する機能の概要について紹介する。

基幹システム構築の現状

基幹システムの構築の現状は、独立した業務システムを構築する形態から、分散した業務システムを連携させて構築するケースが増加しており、そのアプローチは大きく二つに分類することができる。

(1) 一つのシステムを分離し、それらを連携

もともと一つであったシステム内の業務をライフサイクルの違いで複数のシステムに分離し、分離した複数システム間を連携するケースである。このケースは、顧客ニーズにタイムリに対応するためであったり、またライフサイクルの短い業務の頻繁な入替えによる他業務への影響を局所化するためであったりする。

なお、ライフサイクルが短い業務とは、顧客の要

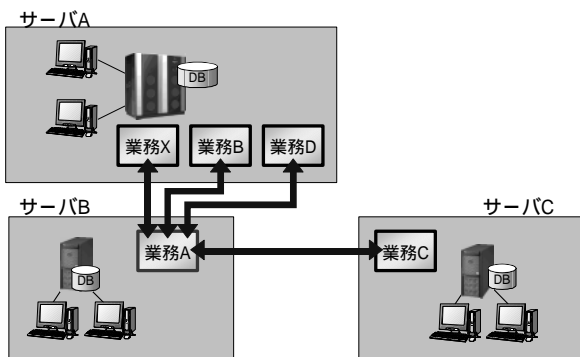


図-1 開発生産性の問題  
Fig.1-System development problem.

件や外部要因などで頻繁に変更のある業務であり、ライフサイクルが長い業務とは、顧客管理など変更がほとんどない業務である。

(2) 分散されたシステムを整理後、それらを連携部門ごとなどに構築されているシステムを統廃合後、それらのシステムを連携するケースである。

このアプローチの目的は、ITシステムのTCO削減を行うことである。

業務システムを構築する場合の問題点

これまで述べたように、分散した業務システムを連携させて業務構築するケースが増加してきているが、構築する場合の問題は大きく三つに分類できる。

(1) 開発生産性の問題

図-1に示すように、一つの業務Aだけを見ても業務X、業務B、業務Cおよび業務Dそれぞれとの組合せがある。業務が増加すると、その組合せが増えるため、業務間の連携パターンが多くなる。この一つ一つの組合せについて、連携するための接続処理を開発しなければならず、サーバ間にまたがるケースも存在するため、エラー要因の増加などにより、エラー処理の複雑化や開発量の増加という問題が発生する。

また、図-2に示すように、新規追加サーバに新しい業務Dを追加する場合には、業務Dと連携のための開発が必要になる。さらに、業務Dだけではなく、連携するそれぞれの業務A、業務Cなどほかの業務にも変更の影響が発生する。このように、業務を新規に追加する場合には非常に時間やコストがかかるという問題が発生する。

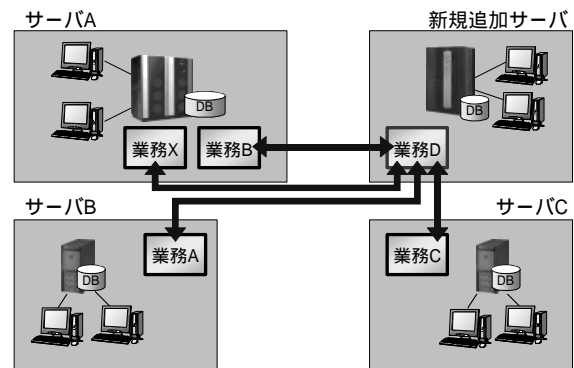


図-2 新規業務の追加  
Fig.2-Addition of new service.

(2) 業務間の整合性の問題

図-3に示すように、業務A、業務B、業務Cを連携させて一連の業務を形成する場合に、業務Cで異常が発生すると業務Cの異常のリカバリ処理が必要だけでなく、すでに処理が完了した業務Aおよび業務Bについても取消しを行う処理が必要である。これらの処理の中でDBのトランザクションが完結していた場合には、トランザクションをもとの状態に戻す必要がある。このような業務の取消処理は、業務ごとに異なるため、業務の連携パターンが多いと処理が複雑となり、また非常に難しい。

(3) 業務の運用上の問題

一連の業務が複数のシステムにまたがって実行されるため、例えば一連の業務のどこかの業務が遅延している場合、どのシステムでどのような理由で遅延しているかの把握が難しく、性能ボトルネックやスローダウンの早期検出が難しい。

基幹システムフレームワーク (MC基盤)

これまで述べてきたような問題解決への取組みとして、分散したシステムにおいて効率的な業務構築を実現する制御機構Interstage Business Application Server「基幹システムフレームワーク (MC基盤)」の三つの機能を紹介する。

(1) 業務間の容易連携

アプリケーションは、制御ロジックと業務ロジックに大別できる。制御ロジックとは、アプリケーションの実行制御、トランザクションの管理およびデータベースのアクセス制御などを行うためのプログラムを意味する。また、業務ロジックとは、顧客情報や注文内容をデータベースに反映するなど業務

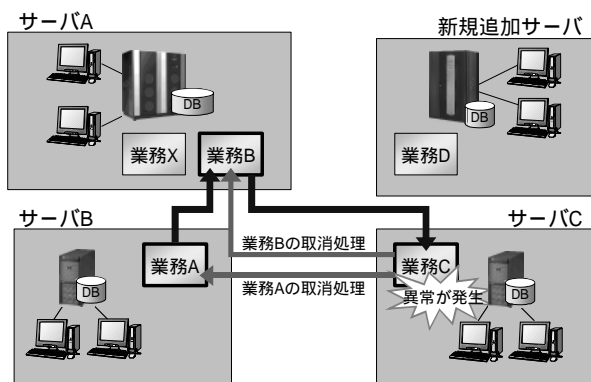


図-3 業務間の整合性の問題  
Fig.3-Problem about consistency between services.

に特化した制御を行うプログラムを意味する。

オープンシステムのアプリケーションでは、従来業務ロジックに加え制御ロジックも開発が必要だったのを、図-4に示すようにMC基盤では、制御ロジックと業務ロジックを分離し、分離した業務ロジックの正常系処理および異常系処理の構造を規定することで、プラットフォーム、ミドルウェアおよびオンライン形態 (同期処理/非同期処理) に依存しない業務部品として、アプリケーションの汎用性および再利用性を高めた。

また、トランザクション制御などの制御ロジックをMC基盤の実行環境として、システム形態ごとにオンラインの同期処理、非同期処理およびバッチ処理の3種類を用意した。

ほかの業務との連携を容易に実現する仕組みとして、フロー定義をベースとした実行環境を提供した。フロー定義では、どの業務をどのような順番で呼び出すか、業務の処理結果によってどの業務を呼び出すかといった条件分岐などが定義できる。

これらの機能により、複数の業務を連携して新たな業務を構築する際の開発量の削減とアプリケーションの独立性の向上が図れる。また、業務の追加・変更が発生した場合でも、既存のアプリケーションを修正することなく、フロー定義を変更するだけで容易に対応することができる。

(2) 業務間の整合性保証

図-5に示すように、業務A、業務B、業務Cの順番で呼び出す業務があった場合に、業務A、業務Bが正常に終了し業務Cで異常が発生すると、業務Bと業務Aの取消し処理を行わなければならない。MC基盤では、フロー定義に従って、業務の呼出し

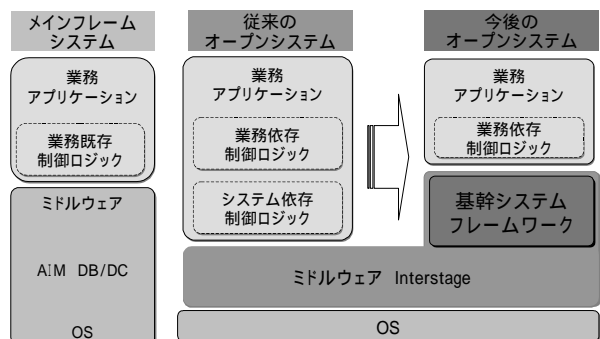


図-4 MC基盤によるシステム構築  
Fig.4-Building of robust system by Interstage.

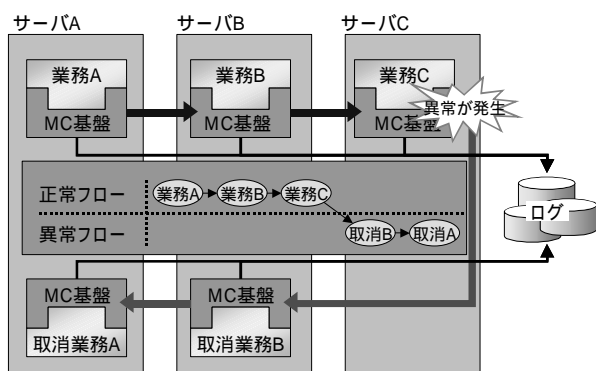


図-5 業務間の整合性保証  
Fig.5-Insurance of consistency between services.

を行い、また、今までどのような順序でどの業務を呼び出したかの管理を行っている。そのため、業務Cで異常が発生した場合には、今まで実行してきた順序を自動的にさかのぼり、それぞれの取消し処理を実行する。これにより、異常処理の簡易化を実現し、また、一連の業務の整合性を保証する。

### (3) 業務の稼働監視

一連の業務を実行した場合に、アプリケーションの呼出しから応答までの時間、パラメタおよびエラー内容などの稼働状況について、利用者の識別情報を自動的に付加して各サーバ単位に出力するロギング機能を提供する。

また、採取したログ情報をSystemwalkerなどの運用管理ソフトウェアで監視、管理することにより、性能のボトルネックやスローダウンなどの異常事象を早期に検知でき、さらに、今後のシステム拡張のための情報としても活用できる。

## む す び

本稿では、分散化、大規模化および複雑化していく企業の基幹システムを、柔軟に統合し、変化に強く、かつ短期間に構築するために、Interstage Business Application Server「基幹システムフレームワーク (MC基盤)」が解決する課題および提供する機能の概要について紹介した。

今回は、オンラインの同期制御やバッチ機能についての説明は、本誌の2006年1月号や2005年1月号で紹介されているため割愛したが、これらのシステム形態にも活用できる<sup>(1),(2)</sup>

すでに、MC基盤を先行提供しているプロジェクトにおいては、業務インフラ開発を行う際の開発効率を約30%向上できるという結果が出ている。今後新たなシステム開発においてMC基盤を活用することで、富士通のSIの開発効率の向上を図り、堅ろうな顧客システムの早期構築を実現することにより、お客様の要望に応えていく。

なお、将来のエンハンスでは、他社製品との相互接続を実現する予定である。これにより、他社モデルウェアプラットフォーム上に構築された業務とも連携させて新たな業務構築が可能となる。

### 参 考 文 献

- (1) 車井 登ほか：フレームワークによる開発作業の効率化。FUJITSU, Vol.57, No.1, p.36-44 (2006)。
- (2) 小川俊雄ほか：スピード時代のシステム構築基盤。FUJITSU, Vol.56, No.1, p.22-30 (2005)。