



**SyOM is a distributed object middleware providing
the right function for the right job to reduce
IT development and administration cost.**

Product Overview

EC SOLUTION

A guide for initial use of SyOM middleware series



Corporate Headquarters
Maruoka bldg, Shinjyuku_ku 3-6
Tokyo, Japan (zip 162-0825)
TEL +81-3-3267-2581 FAX +81-3-3267-2584
E-mail webmaster@avail.co.jp
Web site www.avail.co.jp

本書は、協調分散ミドルウェアである Avail 社「SyOM」の機能概要を説明するための資料です。

SyOM は、電子商取引システム等の広域型ネットワークシステムのようなミッションクリティカルなアプリケーションシステムを、十分に実用性あるパフォーマンスで実現することを目指したミドルウェアです。SyOM では、アプリケーション間のトランザクション通信機能だけでなく、アプリケーションシステム全体のパフォーマンスを向上させる先進的かつユニークな様々な機能を提供しています。電子商取引システム等の広域型ネットワークシステムをリアリティのある、新たなビジネスアプリケーションとして創造されることを目指されている皆様方に、SyOM の機能とその背景をご理解いただき、是非とも次期システム構築の際にご検討いただければと考えております。

Ryuji Agari president Avail,Inc.

Table of Contents

1 . EC が求めるリアリティ	1
2 . SYOM が提供するリアリティ	1
3 . 機能概要～リアリティ実現のために	2
3.1 起動管理の最適化	
3.2 動的圧縮	
3.3 データベースの効果的接続	
3.4 メッセージキューイング	
3.5 相互接続性	
3.6 ジョブ自動化	
3.7 開発生産性の向上	
3.8 MICROSOFT 社 OFFICE オブジェクトのクライアント化	
3.9 信頼性	
4 . SYOM に与えられるべき役割	7
5 . 最後に、「なぜ、SYOM は生まれたか」	8

SyOM : Electronic Commerce Solution

1. EC が求めるリアリティ

新たな経済基盤としての電子商取引システムへの期待が、実用域に差し掛かろうとしています。それはバーチャルなシステムからリアリティのある実用的なシステムへの移行と言えます。Amazon.com、Yahoo など、既に十分実地的な評価を得ている企業も少なくありません。従来ネットワークビジネスで高い評価を得ているこうした企業の多くは、インタラクティブな情報提供サービス業乃至はそうした機能を中核的機能として内在している企業でした。しかし、これから新たに迎える電子商取引の世界は、あらゆる産業を巻き込んだものです。

我々は、リアリティある実用システムでは、バーチャルなシステムと比して以下のような点が重要視され、要求されるものと考えます。

- ・実用システムでは、既存システムより優れたパフォーマンスでなくてはならない
- ・実用システムでは、過去の取引コストよりも低廉な取引コストでなくてはならない
- ・実用システムでは、システムの改変、再編成が容易であること
- ・実用システムでは、過去の膨大な経験と知識の所産が生かせなくてはならない
- ・実用システムでは、既存システムとの連携が図られなくてはならない
- ・実用システムでは、他の外部システムとの連携が図られなくてはならない
- ・実用システムでは、取引継続性を維持する仕組みが組み込まれていなくてはならない
- ・実用システムでは、既存システムと同程度の信頼性が必要である
- ・実用システムでは、悪意の第三者からの侵入を防御できなくてはならない

EC が求めるリアリティとは、こうした要求に速やかに応じることができるだけの電子商取引システムを構築し、提供することであると考えます。

2. SyOM が提供するリアリティ

SyOM は、「1. EC が求めるリアリティ」にある実用システムを構築し、運用するためのソフトウェアです。それは、電子商取引システム等の広域型ネットワークシステムのようなミッションクリティカルなアプリケーションシステムを、十分に実用性あるパフォーマンスで実現することを目指したミドルウェアとして提供されています。SyOM を、アプリケーションシステムの構築に利用することで、リアリティある実用システムに要求されるこうした要件の多くを達成もしくは達成するだけの可能性を生み出すことができます。

パフォーマンス（スピードとコスト）

SyOM で最も重視する要素です。猛烈な勢いで拡大するネットワークは、それ以上の速度で、そこに取り交わされる情報の質的、量的拡大を級数的に進めます。リアリティある実用システムにとって、パフォーマンスの向上が EC ビジネス成功の最重要な課題の一つになることにそれほど多くの時間はかからないと考えます。

SyOM のほとんどの機能は、いかにアプリケーションシステムのパフォーマンスを向上させるかを目標としています。SyOM では、トランザクション処理、システム再構築、アプリケーション開発のそれぞれでパフォーマンス向上のための様々な機能を提供しています。

機能例：ほとんど全ての機能でパフォーマンス向上のための仕組みが含まれています。ここでは代表的なものを掲げます。起動時間の短縮と消費システムリソースの削減（3.1 起動管理の最適化）、動的圧縮（3.2 動的圧縮）、高速アクセスするデータベースアクセス機能（3.3 データベースの効果的接続）

システムの改変、再編成

SyOM では、SyOM に対応したソフトウェアを部品として再利用して、開発効率を飛躍的に向上させることができます。またソフトウェア配信やシステム環境再編成機能、リモートメンテナンス機能によって容易にアプリケーションシステム全体を再編成することができます。

過去の資産の利用と外部連携

過去の資産や外部連携を行うために SyOM では、多言語対応、異種データベース対応、ジョブ自動化機能、ファイル配信機能、他ミドルウェア間リアルタイム連携、非オブジェクト型言語での開発ライブラリ提供を行っています。

注：現行バージョンでは、汎用機とのリアルタイム連携、COBOL 言語開発ライブラリ提供はおこなっておりません。汎用機とのリアルタイム連携が必要な場合は、他ミドルウェアとのリアルタイム連携が必要です。

取引継続性の維持の支援

SyOM には、取引継続性の維持を支援するための仕組みは直接含まれてはいませんが、SyOM の高度なネットワーク機能は、こうしたシステム化ニーズにも柔軟に対応できるいくつかの機能を提供しています。

SyOM では、SyOM 対応のクライアントアプリケーションから WEB ブラウザや WEB サーバーを介さないで、インターネット経由で直接アプリケーションサーバーとの接続を行うことも可能です。それゆえプライベートなネットワークや独自のユーザインターフェイスを使ったシステムを構築することが容易です。

信頼性

SyOM には、アプリケーション系、システム系のそれぞれについて信頼性向上のための様々な機能を提供します。アプリケーション系では、ソフトウェアフォールトトレラントを実現するメッセージキューイング機能、アウトプロセスの利用、エラージョブイベントの発生を提供しています。システム系では、リモートメンテナンス機能、システム環境再編成、ソフトウェア配信機能などによって、システム系全体の信頼性の向上を図っています。しかし、SyOM にとって最も重要なことは、SyOM 自身が単一のアーキテクチャからなる高度に洗練されたミドルウェア層ともいべきネットワーク層を構成していることです。

セキュリティ

現在 SyOM での接続に SSL を介すること、暗号化ソフトを介することは可能です。今後より強力かつ効率的なセキュリティポリシーを提供してゆく予定です。

3.機能概要～リアリティ実現のために

3.1 起動管理の最適化

SyOM では、アプリケーションの利用状況に合致した様々な起動管理のための機能を提供します。起動管理では、起動時間の短縮と消費システムリソースの削減というトレードオフの関係を最適化します。また、異なる起動特性を持つインプロセスとアウトプロセスのそれぞれの特徴を生かし、アプリケーションに期待する高速化と信頼性の目標に合致したシステム構成を可能とします。さらに、異システム間のリアルタイム連携や特定処理の負荷分散を、プログラムコードを何ら変更することなしに起動管理の設定だけで容易に実現するリダイレクト処理も提供しています。

起動制御

アプリケーションの起動方法を制御することで、起動時間の短縮と消費システムリソースの削減というトレードオフの関係を最適化します。例えば売上トランザクションの登録処理のように発生頻度が高く、分散入力され、高速レスポンスが必要とされるアプリケーションでは、予め複数のサーバーアプリケーションを起動しておき、クライアントからのメッセージに直ちに回答できる状態にしておく必要があります。SyOM では、売上トランザクションの登録処理を静的起動に設定しておくことで、起動に伴うプログラムローディングの時間をなくし、レスポンスを高速化することができます。SyOM では、常駐させる同一種のプログラム数を 1 つのみならず複数とすることが可能なことが大きな特徴です。また、想定した以上の数のメッセージがきた場合は、動的に不足分を補うようにプログラムが起動され、それらは処理終了後に、直ちにデストラクトします。

一方、得意先マスターの登録処理のように発生頻度が低く、特定のクライアントから集中入力され、高速レスポンスが必ずしも要求されないアプリケーションでは、クライアントからのメッセージ毎に逐次起動することでも十分に要求を賅うことができます。SyOM では、得意先マスターの登録処理を動的起動に設定しておくことで、メッセージの都度プログラムローディングを行い、実行後は、直ちにデストラクトさせることができます。この方式では、静的起動に比して、レスポンスは高速ではありませんが、プログラム常駐に伴うサーバーリソースの消費を抑えることができます。

インプロセスとアウトプロセス

サーバーアプリケーションとして、インプロセスとアウトプロセスの両方を使えるようにし、アプリケーションに期待する高速化と信頼性のバランスに応じた最適な起動特性を可能とします。インプロセスは、アプリケーションをコンパクト化するために、起動時間が少なく高速な処理が可能になります。一方、複数のインプロセスが起動している同一のアプリケーションにて、そのいずれかが何らかの原因でハングアップした場合、そこで稼働している全てのインプロセスが実行できなくなってしまう。これに対してアウトプロセスでは、信頼性は高くなりますが、起動に伴う手待ち時間が掛かります。

SyOM では、次のような開発指針を推奨しています。アプリケーションサーバー上では、高信頼性のアウトプロセスを使い、高速レスポンスが要求される場合は、静的起動制御によって起動に伴う手待ち時間を排除します。一方クライアント内でサーバーアプリケーションを実行する優位性がある場合については、インプロセスを使い、少ないシステムリソースの元でも高速な処理が可能にします。

リダイレクト

例えば会計システムと販売システムがそれぞれ異なるアプリケーションサーバー上に配置されているとき、クライアントの販売システムの中から特定の会計情報をリアルタイムに照会できる機能を付加しようとする。SyOM では、販売システムのアプリケーションサーバーに、会計システム内の会計情報照会アプリケーションの参照を登録するだけで、会計システムの所在を販売アプリケーション上で何ら配慮することなく、他のサーバーアプリケーションを利用するのと同様に、オブジェクト参照するだけで利用することができます。

SyOM のリダイレクト処理は、このように異システム間のリアルタイム連携や特定処理の負荷分散を、プログラムコードの変更を何ら行うことなしに、起動管理の設定だけで、極めて容易に実現することができる機能です。異システム間のリアルタイム連携をするからといって、新たに連携のためのインターフェイスを追加したり、サーバー内のシステム構成を変更するようなことは必要ありません。

3.2 動的圧縮

SyOM では、サーバー及びクライアントオブジェクト間のメッセージを、そのセッション毎に圧縮することができます。セッション毎に圧縮することが可能なので、例えば、同一のアプリケーションを LAN と WAN でそれぞれ利用し、前者では圧縮を行わず、後者では圧縮を行うような使い方が可能です。圧縮及び解凍に伴う負荷とネットワーク負荷のトレードオフの関係から利用環境に応じた選択が可能です。圧縮に伴うプログラムコードの変更も全く不要にすることができますから、圧縮の有無によって複数のプログラムを作る必要もありません。

圧縮率は約 1.5 ~ 2.0 倍です。ゆえにインターネット経由する低速ネットワークで大量のトランザクションを扱うような場合の効果は絶大です。150KB のデータを、10KB 以下に圧縮するので、実転送レートが 30 ~ 40KBPS (ISDN) にて 1 分以上かかる転送をほんの数秒程度にまで高速化することになります。また、ソフト圧縮ですから圧縮のための特別な機器を備える必要もなく、設備コストを低く抑えることができます。もちろん、ハード圧縮の機器と併用して、より高速な転送を行うことも可能です。

3.3 データベースの効果的接続

アプリケーションをデータベースの種類に依存することなく、標準化された同一ないし同種のアクセスルールを適用して開発し、しかもできるだけ高速なアクセスを実現する。SyOM は、データベース接続に期待されるこうした本来トレードオフの関係にある開発者の期待に応える、極めてユニークかつ実効性のある機能を提供しています。

共通アクセスコア

SyOM が提供する共通アクセスコアは、特定のデータベースに依存することのない、標準化された同一ないし同種のアクセスルールを提供します。現在は、Oracle、SQL Server、Jet に対応しており、さらに他のデータベースへの拡張を予定しています。

セッション管理機能

SyOM では、データベースとのセッション数を統制、管理するユニークな制御機能を持っています。直接接続方式は、プログラム毎にデータベースとのセッションを確立し、高速なアクセスを可能にします。一方、間接接続方式は、複数のプログラムを束ねて、1つのセッションでデータベースと接続する方式です。これによってアクセス速度は直接接続方式に比して落ちますが、利用できるセッション数に制限がある場合など利用可能なシステムリソースの制約が大きな場合に有効です。SyOM では、プログラムから接続方式を宣言することで、こうしたセッション管理機能を使うことができます。

高速アクセス

SyOM では、それぞれのデータベースの下位レイヤー（例えば Oracle では OCI、SQL Server では DBLibrary）との間で直接通信を行い、データベースアクセスを実現しています。また、それ自身非常にコンパクトな作りになっているために、ODBC (Microsoft 社) に比して機能面では劣るかもしれませんが、高速なアクセスを実現しています。

ライブラリ&コンポーネント

SyOM では、データベースアクセスのためのインターフェイスをライブラリ及びコンポーネントの2種類の形で提供しています。ライブラリを C++ や Java のソースコードから直接参照したり、VB も含めて、それらより ActiveX コンポーネントとして参照することができます。

必要に応じて

SyOM 上のアプリケーションにとって、本データベースアクセスモジュールを利用することは必須ではありません。

ODBC や他のアクセスモジュールを利用することも何ら問題ありません。私たちは、開発者がその開発ニーズに応じて、少しでも容易に開発ができるための環境を提供することを目標としています。従来のアクセスロジックをそのまま流用して開発するのであれば、例えば ODBC やその他のアクセスモジュールをお使いください。高速化を優先し、単純な機能でデータベースアクセスを実現しようというのであれば本データベースアクセスモジュールを利用されることを推奨致します。

3.4 メッセージキューイング

メッセージキューイングを利用した非同期処理は、システムの可用性、信頼性を高度化すると共に、プログラムを単機能化することで開発効率の向上とバグの少ないプログラミングを可能にします。キューイングされるメッセージにデータ型の制約はありません。通常のトランザクションの他 XML メッセージなどもキューイングすることができます。キューファイルは、アプリケーションからのリクエストに対してキューメッセージを戻し、内容を削除します。同時に他のキューファイルにメッセージを転送することも可能です。また、キューファイルへの登録時、更新時（削除含む）にキューファイルからイベントを発行して外部アプリケーションを起動することもできます。これらによりポーリング型のシステム、イベント駆動型のシステムのいずれも容易に構築することができます。

負荷分散

SyOM は負荷分散として、同系分散と異系分散をサポートしています。同系分散では、キューイングされたメッセージを、1つ又は複数のアプリケーションサーバーで起動する同一のサーバーアプリケーションによって処理します。例えば、分散入力された売上トランザクションを売上トランザクションキューに登録し、それを非同期に実行される複数サーバー上の販売データベース更新アプリケーションにてデータベースへ更新することが可能です。

異系分散は、垂直分散と特異分散に分けられます。前者は、処理プロセスを実施手順に従ってグループ化し、グループ間をキューで連携し、それぞれのグループを異なるアプリケーションサーバーで処理させるものです。後者は、例外処理や高負荷処理だけを別のアプリケーションサーバーで処理させるものです。

フォールトトレラント

同系分散下では、いずれかのサーバーアプリケーションが停止した場合でも、他のサーバーアプリケーションが稼動していれば処理を継続することができます。アプリケーションサービスを継続したまま、障害が発生したサーバー機を交換し、再稼動させることも可能です。また、アプリケーションの実行結果を監視するためのキューを用意することで、異常が発生した処理系を自動的に発見し、管理者に警告を発することも可能です。後述するバッチジョブとの連携機能を使えば、こうしたアプリケーション単位での極めて詳細な監視システムを容易に構築することができます。

バッチジョブとの連携

SyOM では、キューファイルを介して、トランザクション処理とバッチ処理とをシームレスに連携する機能を有しています。トランザクション処理で特定のキューファイルにメッセージが登録されると、それをイベントとみなしてジョブ自動化機能で登録されたバッチ処理を起動することができます。これらによって従来にない、トランザクション処理とバッチ処理をイベント連携したアプリケーションを構築することができます。現実のほとんどの業務は、トランザクション処理とバッチ処理との連携です。SyOM を使うことでこれらをシームレスに連携したアプリケーションを構築することができますと共に、それぞれを異なるサーバーで実行させることができます。

プログラムの単機能化と高速化

トランザクションを処理するためには、トランザクションデータの生成、データベースへの登録、不整データの通知、トランザクションデータの修正及び削除などの一連の機能が必要です。従来の多くのオンラインシステムでは、これらを1つのトランザクション処理プログラムで構築しているがゆえに大量のコードを抱えるメンテナンス性が低いものとなっています。メッセージキューを利用すれば、これら機能を分別し、単純化された機能単位にプログラミングすることができます。ゆえにデバッグが大変、容易になります。また、プログラム単位での機能や使い方の習得が、容易なため、開発されたプログラムを他のシステム内でプログラム部品として活用することが容易です。

単機能に区分化されたプログラムとすることで、トランザクションの登録はメッセージキューへの登録にて完結し、以降のデータベースへの更新処理等からは分別されるために、利用者には極めて高速なレスポンスなシステムを構築することができます。

3.5 相互接続性

SyOM を利用することで、短期間で異アプリケーション間を相互接続し、統合化することが可能になります。SyOM では、CORBA 対応ミドルウェアである Inprise 社の VisiBroker 上のサーバーアプリケーション（サーバーオブジェクト）を独自の CORBA ゲートウェイを介して直接参照し、実行させることができます。また、異アプリケーションの実

行モジュールを直接実行したり、データファイルを自動生成及び配信するファイル連携での間接的な接続も可能です。

CORBA ゲートウェイ

SyOM では、アプリケーションサーバー上の CORBA ゲートウェイ介して、CORBA 対応ミドルウェアである Inprise 社の VisiBroker 上のサーバーアプリケーション（サーバーオブジェクト）を直接参照し、実行させる方式（ブリッジ接続方式）を採用しています。これによってクライアントアプリケーションでは、接続先のサーバーアプリケーションがどのようなプラットフォームにあるかを全く知らなくても、プログラムを生成することができます。また、クライアントアプリケーションとサーバーアプリケーションとの間では、SyOM の高度なオブジェクト間通信サービス（圧縮、起動制御その他）をそのまま利用することができます。

CORBA ゲートウェイの利用は極めて容易です。CORBA IDL ライクなインターフェイスを定義ファイルに記述するだけで、直ちに利用することができます。

非分散オブジェクトシステムとの連携

SyOM の高度なジョブ自動化機能は、非分散オブジェクトシステム自体をジョブとしてとらえ、その実行モジュールを直接実行したり、データファイルを自動生成及び配信することでファイル間連携を、スケジューリングすることができます。スケジューリングは、時間指定の期間スケジューリングの他、ファイルイベントスケジューリングとメッセージイベントスケジューリングの2つのイベント駆動スケジューリングを行うことができます。

ファイルイベントスケジューリングは、ファイルの生成をイベントとしてジョブを実行させる機能です。一方メッセージイベントスケジューリングは、非同期処理を行うメッセージキューのキューファイルにメッセージが生成されたことをイベントとしてジョブを実行させる機能です。これらイベントスケジューリングを利用することで、従来汎用機の世界でオンラインバッチと称されてきた、リアルタイムに近い非同期のバッチ処理を実現することが可能です。

また、データファイルを自動生成及び配信し、受信先の異アプリケーションで自動取込の処理を起動する、いわゆる完全なバッチ型のファイル間連携による相互接続を実現することも容易に実現できます。

3.6 ジョブ自動化

ジョブ自動化機能は、SyOM の大きな特徴の一つです。他の分散オブジェクトミドルウェアの多くが、オブジェクト間通信のための通信機能を提供することに機能限定されているのに対して、SyOM では、現実システムを実現するための統合的プラットフォームという視点からバッチジョブの自動化機能を予め組み込んでいます。しかも、バッチジョブを高度に汎用的なジョブスクリプト言語で記述する機能及びバッチジョブと分散オブジェクト間の通信機能とをシームレスに連携する機能によって、バッチジョブであっても、実用性と汎用性に加えて、かつてない高度な相互接続性を得ることができます。

汎用的ジョブスクリプト言語

SyOM では、ジョブスクリプト言語として汎用的なジョブスクリプト言語（Microsoft 社の VBScript、Jscript（JavaScript）、MS-DOS コマンド、Batch ファイル用スクリプト、Perl）が利用できます。汎用的なジョブスクリプト言語を採用したことで、既存技術の延長としてスクリプトの開発を行うことができます。Microsoft 社の VB や VBA の開発者であれば、VBScript を習得することは容易です。他のスクリプト言語についても同様のことが言えます。また、スクリプトのテンプレートやウィザードも用意しているので、それらを利用すれば、開発経験の無い一般ユーザであっても、習得は決して困難なものではありません。

Microsoft 社 Office オブジェクトの生成

SyOM では、Microsoft 社の VBScript、Jscript をそのまま使うことができるので、ジョブの実行結果を Microsoft 社の Excel 表や Word 文書、ホームページなどとして出力することができます。これら書類作成にあたっては、テンプレートファイルを利用したり、マクロ言語として VBA や、関数設定も可能です。最近では、多くのビジネスアプリケーションで Excel や CSV 形式のデータ出力が可能となっていますが、エンドユーザにとってそれらは最終成果物ではなくさらなる加工を施しています。SyOM を使うとエンドユーザが、その後何ら加工を施すことなく、そのまま利用ができるドキュメントを自動生成することができるのです。

トランザクション処理（分散オブジェクト間通信機能）との連携

ビジネスシステムの処理系は、多くの場合、逐次処理するいわゆるオンラインリアルタイムなトランザクション処理とタイムスケジューリングされた一括処理であるバッチ処理に分けられます。それぞれは独立して実行されることを前提としてプログラミング化されています。しかし、例外処理や集計処理のように、特定のイベントが発生したときのみ、非同期でかつ大量のデータを一時にまとめて集計、分析するようなことがあります。障害発生時のエラー情報の収集、分析などがその典型でしょう。SyOM では、トランザクション処理とバッチ処理を連携するユニークかつ実用的な機能を持っています。メッセージイベントスケジューリングは、トランザクション処理からメッセージキューファイルにメッセージ

が生成されたことをイベントとしてジョブを実行させることができます。SyOM によって、従来汎用機の世界でオンラインバッチと称されてきた、リアルタイムに近い非同期のバッチ処理を実現することが可能になるのです。

ファイル転送及びメール送受信機能

ジョブ自動化機能から SyOM のファイル転送機能を実行すると、作成された Excel 表や Word 文書、その他ファイルを特定のクライアントやアプリケーションサーバーに自動転送することができます。また、メール送受信する ActiveX コンポーネントでメールサーバーとの間でメール情報の授受が可能です。携帯電話等、インターネットメールに対応しているものであれば、相互通信が容易に実現できます。作成された Excel 表や Word 文書などをメールの添付書類とする 것도可能です。データベースアクセス機能と組み合わせることで、実用性の高い付加価値情報をタイムリーに送り出すことが可能です。

データベースアクセス機能

より質の高い付加価値情報とするためには、データベースとの情報交換は避けられません。SyOM では、トランザクション処理のためのデータベースアクセス機能と同様の機能を、ジョブスクリプトの中でも使えるようにコンポーネント化しています。また、データベースアクセススクリプト生成ウィザードを用意しているので、スクリプト言語に不慣れな開発者やエンドユーザでも、容易にデータベースアクセスするスクリプトを生成することができます。

分散ジョブの実行

SyOM のジョブ実行機能は、特定のアプリケーションサーバー上でジョブを実行させるだけでなく、複数の分散したアプリケーションサーバーを同時に管理し、それぞれのアプリケーションサーバーでジョブを実行させることが可能です。ジョブサーバーは、ジョブのスクリプトやスケジュールを管理するリポジトリサーバーと、そこに記されたジョブを実際に実行する複数の実行サーバーから構成されています。またジョブの管理クライアントでは、複数のリポジトリサーバーを管理することも可能です。

3.7 開発生産性の向上

開発生産性の向上は、SyOM の最も重要なテーマの一つです。SyOM は、技術、経験がそれぞれに異なるアプリケーション開発者の誰もが、既存技術の延長で、広域ネットワークに対応した本格的なビジネスアプリケーションを開発、運用できるプラットフォームとなることを目指しています。我々は、こうしたビジネスアプリケーションの開発生産性向上を支援するものとして、1) マルチリンガル対応、2) 標準化されたコーディング作法、3) 異種データベースへのアクセスコードの標準化、4) ビジネスアプリケーション開発に特化した開発環境、5) ソフトウェアの部品化を提供しています。

マルチリンガル対応

SyOM では、クライアントアプリケーションは、C++、Java (Servlet 及び Applet)、VB、VBA、Perl にて開発することが可能です。サーバーアプリケーションは、C++、Java、VB、VBA にて開発することが可能です。ジョブスクリプトとしては、VBScript、JavaScript、MS-DOS コマンド、Batch ファイル用スクリプト、Perl を利用することができます。クライアントアプリケーションには、RCC (リモートコールクライアント)、サーバーアプリケーションには、RCS (リモートコールサーバー) が、ライブラリ乃至コンポーネントとして、それぞれの言語に対応して提供されています。

標準化されたコーディング作法

マルチリンガルであったとしても、そのコーディング作法がそれぞれに大きく異なっていたのでは、設計工程の生産性低下、技術移転コストの増加を招来する可能性は少なくありません。SyOM では、いずれの開発言語を採用したとしても、共通したアルゴリズムを持つ、洗練された、最小限のコードで、プログラミングすることができます。特に、アプリケーション開発者が、ビジネスロジックやユーザインターフェイスの開発に注力できるように、ネットワークプログラミングや起動制御プログラミングについては、徹底したコンテキスト化を図り、開発者が直接それらをプログラミングすることをできる限り回避できるようにしています。

異種データベースへのアクセスコードの標準化

ほとんどのリレーショナルデータベースが、標準化された SQL を利用したアクセスを行うことができます。しかし、それぞれの異なるアーキテクチャは、異なるアクセスのためのアルゴリズムを開発者に要求します。SyOM では、こうした異種データベース間のアルゴリズムの相違を吸収し、標準化されたアクセスコードを利用すること可能にします。データベース毎の SQL 方言を除けば、データベースの種類を変えるだけで、作成されたアクセスコードをそのまま異なる種類のデータベースにも適用することができます。

ビジネスアプリケーション開発に特化した開発

SyOM では、ビジネスロジック、ユーザインターフェイス、データベース構築などビジネスアプリケーションの開発に特化した開発を支援します。ネットワークプログラミングや起動制御プログラミングをアプリケーション開発者が、開発すること、意識することをできるだけ排除するようにしています。それらは、徹底したコンテキスト化を図り、運用環境設定として、事後に、自在に設定することができます。こうして SyOM は、SyOM 上に構築されたアプリケーションに、高度なネットワーク機能や起動制御機能を付加します。

ソフトウェアの部品化

SyOM で開発されたアプリケーション、特にサーバーアプリケーションは、ソフトウェア部品として、その後の開発効率を飛躍的に向上させます。クライアントアプリケーションでは、サーバーアプリケーションのメソッドのコールは、内部関数コールと同様の単純な形式でコーディングし、デバッグすることができます。

3.8 Microsoft 社 Office オブジェクトのクライアント化

SyOM は、最も普及し、使い慣れた利用者環境である Microsoft 社の Excel や Word などの Office オブジェクトを直接、分散システムのクライアントにすることができます。SyOM では、クライアントアプリケーションの開発のために ActiveX コントロールを提供しています。これを Office オブジェクト内に挿入することで、Microsoft Visual Basic for Applications にて利用できます。その結果、例えば Excel をクライアントにして、トランザクションの登録、検索など一連のトランザクション処理を行うアプリケーションを開発することができます。SyOM の提供する動的圧縮、メッセージキューイングなどの高度な機能も使うことができます。SyOM を利用することで、Excel クライアントでのインターネット経由の十分に実効性を保証された画期的なアプリケーションシステムを構築できます。

C++、Java 等の言語系アプリケーションに比して、パフォーマンスは落ちますが、Office オブジェクトをクライアントに利用することは、エンドユーザにとって、使い慣れた、しかも極めて高度なインターフェイスが利用でき、その効果は、定式化された従来のクライアント・サーバー型アプリケーションの評価基準では一概に評価できないものを持っています。優れたユーザインターフェイスとネットワーク情報の結合は、真の「操作性」と、ユーザの「情報力とその加工力の向上」をもたらすものと考えます。

3.9 信頼性

SyOM では、アプリケーション系、システム系のそれぞれについて信頼性向上のための様々な機能を提供します。アプリケーション系では、ソフトウェアフォールトトレラントを実現するメッセージキューイング機能、アウトプロセスの利用、エラージョブイベントの発生を提供しています。システム系では、リモートメンテナンス機能、システム環境再編成、ソフトウェア配信機能などによって、システム系全体の信頼性の向上を図っています。

しかし、SyOM にとって最も重要なことは、アプリケーション層と TCP/IP ネットワーク層との間に、ミドルウェア層とも呼ぶべき、独自のネットワーク層を形成しているところにあります。トランザクション間通信機能、システムリソースの制御機能、システム環境の編成機能を一体化することで、アプリケーションのシステム全体の系の安定化を図ることを目指しています。ミドルウェアと分類される多くのソフトウェアが、例えばトランザクション通信機能やジョブ管理機能などをそれぞれ独立した製品として提供されています。しかし、ネットワークシステム全体から見て、それは果たして有益な手段といえるでしょうか。ネットワークリソースやシステムリソースを奪い合う。アプリケーション間よりもミドルウェア同士で錯綜し、システム系全体を保証することはなおざりになっていないでしょうか。SyOM の最も洗練された機能とは、システム系全体を単一のアーキテクチャで構成し、保証する機能といえます。

4.SyOM に与えられるべき役割

SyOM は、リアリティこそがアプリケーションシステムに要求される評価基準と考えています。

SyOM は、アプリケーションシステム全体の信頼性を単一のアーキテクチャを採用することで、高度に高めることを目標に開発されたミドルウェアです。その意味で SyOM にとって最も重要な評価基準は、実効性であり、実用性です。個々理論的に優れた仕組みであったとしても、それを適用したシステムの全体が、十分に実用的なものでなくては意味がありません。SyOM に与えられるべき役割とは、現在及び将来に渡って、先進的技術を取り込みながらも、常に安定した基本構造を維持しつつ、高度な信頼性を継続して実現してゆくことと考えています。

5.最後に、「なぜ、SyOM は生まれたか」

SyOM は、ビジネスアプリケーションを「協調的に分散問題を解決するための支援機構である」と考えています。

SyOM は、「協調」分散システムと称しています。なぜ、「協調」が付くのか。実は、SyOM は、アプリケーション、特にビジネスアプリケーションの基本モデルが、「環境適応を行う、デジタルネットワークを介在した、自己組織化モデル」と考えています。それゆえ、最初から SyOM の対象は、アプリケーションシステムの全体であり、個ではないのです。また組織は、デジタルネットワークを介在した分散されたオブジェクトから構成される以上、システムは、分散システムでなくてはならないのです。さらに、環境との対話のための、様々な接続機能があり、システムの系全体を再編成させるシステム環境編成機能がなくてはならないのです。そして、その間に発生する様々なデータ通信が、いかに高速に伝達するか、遅延すればするだけ、系の均質性は阻害されます。これを除去するためのスピード、リアリティが要求されるのです。

SyOM は、ビジネスアプリケーションの最重要課題は、パフォーマンス、クオリティ、セキュリティにあると考えています。特にパフォーマンスは、トラザクション処理の高速化だけでなく、いかに開發生産性を高めるかが重要と思います。ここでの高い生産性とは、簡易開発言語やその開発環境という意味ではなくて、ネットワークやデータベース技術とビジネスロジックを切り離し、アプリケーション開発者が、利用者の本来的ニーズを高度かつ的確に反映する開発を効率的に行うことを可能にするという意味です。システムリソースを最大限に効率化することを目指す ASP の場合、ここでの本来的な開發生産性の向上は大変重要な意味を持つものと考えています。

/*

開発元である Avail 社は、7 年前（1993 年）に設立した若い企業ですが、ネットワークビジネスとネットワークテクノロジーとの結合をめざし、様々な態様のミドルウェアを開発しております。是非とも、SyOM の機能とその背景をご理解いただき、リアリティのある、新たなビジネスアプリケーションをご検討いただければと思っております。

*/

会社名、製品名などの固有名詞は、一般に該当する会社もしくは組織の商標または商標登録です。