WebOTX 概要編

WebOTX 概要編

バージョン： 8.1

版数： 第 5 版

リリース： 2018 年 7 月

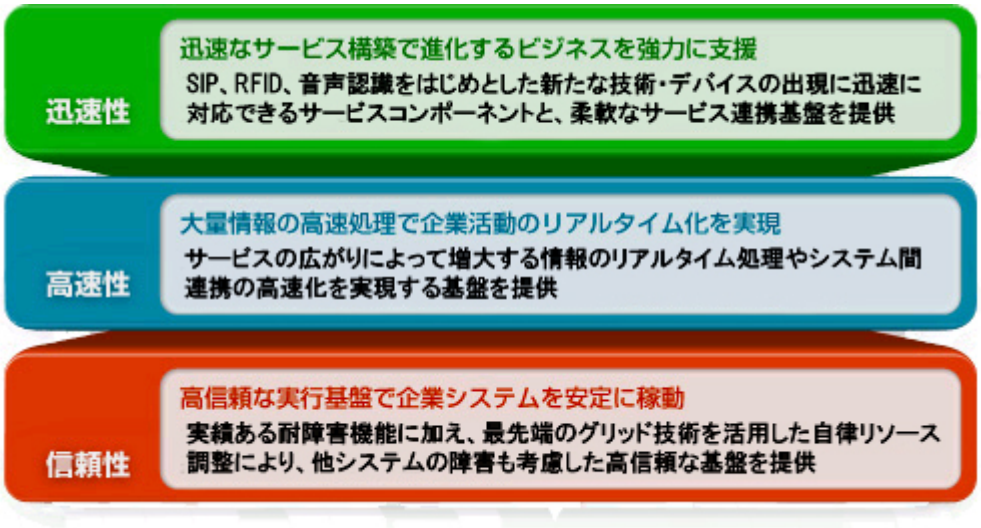
目次

1.はじめに

経営環境のグローバル化による進展、モバイル端末や RFID 等のユビキタス機器の普及、NGN によるネットワーク環境の改善など、企業を取りまく環境は大きく変化しています。

こうした環境で企業が勝ち残っていくには、多様化するサービスや情報・ユビキタス機器の、いち早い活用と、増大し続ける膨大な情報を高速かつ効率的に処理し企業経営にフィードバックできる企業システムの IT 化がなによりも重要となります。

そのような中、WebOTX は進化するお客様のビジネスを支える「サービス実行基盤」として、製品構成を拡張しました。企業の経営基盤強化と新しいビジネス創造に向けた企業システムの構築を先進テクノロジーで支援します。NGN やユビキタス領域、OMCS で培った技術を核に「迅速性」「高速性」「信頼性」に優れた多彩な製品群を提供します。



本書では、高い信頼性を保証する実行基盤として多くの実績を誇る「アプリケーションサーバ製品群」、および日々増大する情報を高速に処理したり、それらをもとに広がるサービスや種々システムを迅速、かつ効率よく連携するために必要となる「サービス統合製品群」について概要を説明します。

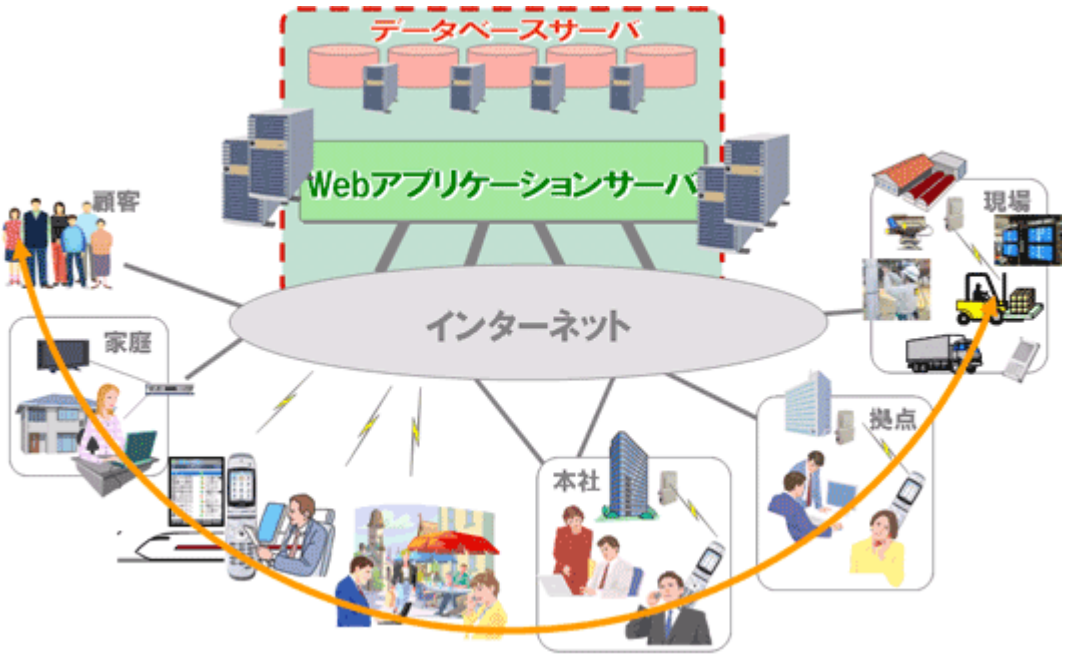
2.Web AP サーバとは

ここ数年でインターネットを中心とした情報技術(IT)の革新、およびブロードバンドやモバイルの普及によるユビキタス化が急速に進んでいます。

24 時間 365 日どこからでも接続できる、安価で高速なインターネットの利用が公共機関、企業、家庭、個人へと拡大し、さらに電子ショップ、電子商取引、ネットバンキング、ネット証券、音楽・映像配信、ネットオークションなどのシステムが当たり前のように社会に浸透しています。

さらに今後は、RFID 等のセンサー機器や IC カード、情報家電の普及、および NGN によるネットワーク環境の改善が急速に進んでいくでしょう。インターネットへの依存度が高まる一方で、流れるデータのサイズや、処理件数の幾何級数的な増加が予想される事から、今後さまざまな課題がでてくることも考えられます。

当然ながら、社会インフラ、および企業の顔として使われるインターネット業務システムは、高い信頼性が必要になります。また、不特定多数によるアクセスが生じる特性のため、ひとたび人気が出たり、注目されると急激にトラフィックが増加し、システムのレスポンス低下や、最悪の場合システム自体の停止といった問題も出てきてしまいます。さらに、悪意の利用者からのセキュリティアタックなど既存の企業内システムでは考えられなかった問題も出てくるでしょう。影響度の高いシステムでそのような事はビジネス機会損失、および企業イメージの失墜を招くため絶対に避けねばなりません。



また、このようなインターネットを中心としたユビキタス社会への進展は、公共機関や企業内の業務システムの変革を余儀なくしています。公共機関は電子政府や電子自治体などの効率的で市民へのサービス性の高いシステムの提供要求に、企業は世界レベルの競争に打ち勝つ効率的な調達や意思決定などの業務システム構築の要求にさらされています。

これらの厳しい要求に対応すべく誕生したのが、ウェブアプリケーションサーバ(Web AP サーバ)です。具体的にどのような事が Web AP サーバに求められるのかを次に示します。

2.1.信頼性

業務システムの信頼性の要件については、提供する業務や規模毎に異なるために一概に言えるものではありません。しかしエンタープライズレベルの業務システムやインターネット業務システムには最高レベルの信頼性が要求されるでしょう。

アプリケーションサーバにはサーバ、ネットワーク、ディスクなどのハードウェア障害や、アプリケーション

の予期せぬ障害などのソフトウェア障害を局所化して全体のサービスへの影響を最小にできる信頼性が必要です。ハードウェア障害に対してはクラスタ技術やフェイルオーバ技術などを使った冗長構成も必要となります。

2.2.可用性

インターネット業務では 24 時間 365 日運用を止めずに安定してサービスを提供し続けられるシステムが必要となります。また、急激な負荷変動があった時でもプライオリティの高い業務やユーザに対してのサービス継続を優先するサービスレベル保証の考え方も重要となります。たとえ現在の業務システムがイントラネットに制限され 24 時間運転をしていないとしても、ビジネスがグローバルに拡大している今日、いつインターネット業務システムへの対応を迫られるかわかりません。

アプリケーションサーバでは無停止運用技術が重要となります。つまり、無停止でサービスを提供し続けられるように各種カウンタやログファイルなどが慎重に設計されていることや、サービスを提供するアプリケーションを動的に追加・置換できることなどが必要です。

2.3.性能

インターネット業務システムは企業内システムでは考えられない不特定多数からのアクセスが発生します。これを効率良く処理するため、またサービス自体の高度化によるデータ処理量の増大に対応するためにも高性能なアプリケーションサーバが必要となります。

2.4.拡張性

トラフィックの増大に対して、最終的にハードウェアの増設が必要となります。システム全体で提供するサービスを停止することなく、サーバ、CPU、メモリなどのハードウェア資源を増設できる拡張性が重要です。

また、最適な機能分散により性能ネックが発生している部分を切り出し、そこに集中して効率的にハードウェアを増設することが可能です。これによりシステム全体のハードウェア投資を削減、結果としてハードウェアを有効活用することができます。システム内のアプリケーションの配置を自由に変更でき、ビジネスロジックがネックになっていればアプリケーションサーバを増設、プレゼンテーションロジックがネックとなっていれば Web サーバを増設するなどシステムを柔軟に変更できる高い拡張性が求められます。

2.5.柔軟性

たとえ慎重に設計したとしても、システムは運用を開始するとエンドユーザから多くのサービス追加や変更の要求を受けるものです。また、IT が急速に変動している現在、今後の新たな技術への対応が求められることは容易に想像できます。そのためには、最新の技術を採用するのは当然のこと、今後の新技術との親和性が高く新製品との相互接続性の高い汎用的な技術を使ったアプリケーションサーバが必要になります。

2.6.生産性

IT の革新はグローバルレベルの競争を現実のものとしました。このような環境では、他社に勝るサービスをスピーディに提供することが重要になります。そのためには生産性の高い開発環境と簡単に確実な評価を行える評価環境を持ち、アプリケーションのスピーディな開発を実現することが要求されます。また、業務システムを再構築するのではなく、既存のデータベースや業務アプリケーションを資産として有効利用して新たなサービスをスピーディに提供する、エンタープライズ・アプリケーション・インテグレーション（EAI）が容易に実現できるシステムが必要です。

2.7.運用性

インターネットを介したシステムでは、基盤として使われるハードウェア・ソフトウェア・技術などが広範囲に及びます。これらを 24 時間 365 日無停止でシステムを運用するには、運用者に多くの技術蓄積と監視・運用操作の負担をかけます。可能な限りシステムを自動的化することにより、必要最小限の操作で容易に監視・運用操作でき、操作ミスが発生しない運用環境を実現します。

2.8.保守性

慎重にシステムを開発しても、予期せぬ障害をゼロにするのは困難です。特にインターネット業務システムでは過負荷による障害発生の恐れがあります。障害解析に必要となる情報を最小限のシステム負荷で確実に取得し、平時でも容易に性能予測できることが重要です。また、アプリケーションの最適な配置、部品化はシステム全体のモジュラリティを向上し、変更による影響を明らかにします。そのためにもソフトウェアの構成管理や世代管理と連携した開発環境が重要になります。

2.9.安全性

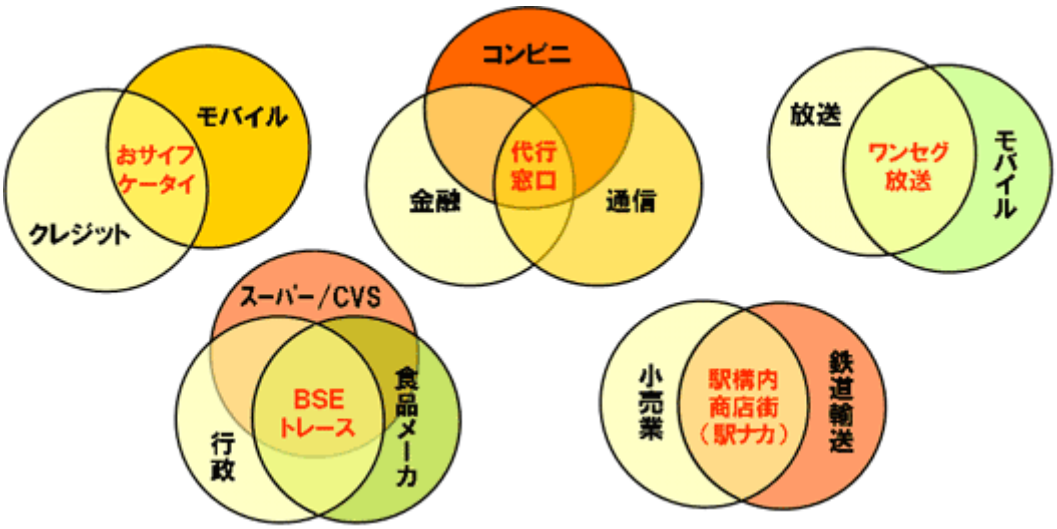
インターネット業務システムに限らず、セキュリティアタックは多数発生しています。その結果今日ではインターネットの End-to-End の透過性がファイヤウォールにより遮断されるようになりました。機密データの露呈や、不正アクセス、改竄、盗聴、成りすましなどアタック技術も日々進歩しています。現時点では十分なセキュリティ対策だとしても、将来のアタックには無効かも知れません。

アプリケーションは上記のようにスピーディに強化されることが必要ですが、一度作ったアプリケーションがセキュリティ攻撃に対応するために改造を必要としたのでは大変です。ファイヤウォールなどのアプライアンスサーバを活用して、セキュリティ強化の影響をアプリケーションから独立させることができる実行基盤が重要になります。

3.サービスインテグレーションとは

ブロードバンドやモバイル普及によるユビキタス化が加速度的に進んでいる中、24 時間 365 日どこからでも接続できる安価で高速なインターネットの利用が公共機関、企業、家庭、個人へと拡大し、さらに電子ショップ、電子商取引、ネットバンキング、ネット証券、音楽・映像配信、ネットオークションなどのシステムが当たり前のように社会に浸透しています。

さまざまな業界で多種多様な業務サービスがインターネット上で構築・運用されている中、お客様や市場・企業のニーズも多様化してきています。それにともなって複数のサービスを組み合わせ、新たなサービスを編み出し活用していく動きがでてきています。



一方、企業内の情報システムに目をやると、企業合併や内部統制・個人情報保護などの法規制に見られるような経営環境の大きな変化や、競争激化などの理由から新規の事業立上げ、経営の意思決定を支援するための新しいシステムの構築、既存事業の収益力強化のためのシステムの改修などがめまぐるしく進められてきており、その結果多種多様で、部門・システムごとに運用や構成がバラバラな業務が乱立した状態となっているものも少なくありません。

その結果システム保守工数の増大のみならず、実際の企業経営判断にも大きな悪影響を及ぼす状況に頭を悩ませている経営者の方も多いことでしょう。

こういった企業内外におけるビジネス連携・システム統合を柔軟かつ迅速に進めることや、事業環境の変化や戦略変更にすばやく適応できるような、ビジネスを支える真の情報システムを構築することが今後求められていくと考えられます。

そのような背景で脚光を浴びているのがサービス指向アーキテクチャ(SOA: Service Oriented Architecture)です。WebOTX が提供するサービスインテグレーション製品群 は、次に示すような SOA に基づく企業システム構築に求められる要件にお応えします。

3.1.サービスインテグレーション製品導入前に

企業における IT 投資の重点課題は単なるコスト削減から、「業務プロセス・システム再編」といった戦略的な投資への期待が大きく増加しています。それに伴い「SOA」への関心も日に日に高まっています。

企業の経営者・情報システム部門の方は、システム構築の方法論、プラットフォームのあり方、ビジネスプロセスの改善、アプリケーションのつくり方など、さまざまな観点から最適な仕組みは具体的に何かと考えた結果、標準インタフェースを持ち、着脱の容易なサービスの集合体としてシステムを構築する SOA が有効ではないか、と認識されているようです。

では、実際にどこから手をつければいいでしょうか。ただやみくもに「SOA に基づくシステムを構築するための」製品群を導入するだけでは真の解決は望めません。それどころか、システムのオーバースペック、それに伴う運用コストの増大の可能性も否定できません。

企業それぞれが目指すシステムのあるべき姿、ゴールによって対応パターンは変わります。

本質は、ビジネスの変化が激しくなっている中で、いかに業務面やシステム面での保守性を高め、長期間に渡って使えるシステムを構築できるかにあるといっても過言ではありません。ライフサイクルにおけるシステム変更コストをどのように最適化するかということです。できるだけ長期的な視点に立ってビジネスプロセスを考慮し、最適と思われるサービスの粒度に分割したシステムやアプリケーションを構築していくための「企画・コンサルティング」が必要です。

3.2.サイロ型の縦割りシステムからの脱却

「多種多様で、部門・システムごとに運用や構成がバラバラな業務が乱立した状態」に陥った企業において、新たな業務を追加する、あるいは既存の業務に修正を加えるとなった場合、全てのシステムで対応しなければならない、部門間で複雑な調整が必要など予想外の負担を強いられることになります。運用面でも新たな教育や作りこみが発生することになり、変化の激しさに対応できなくなることが考えられます。

逆に、一つの企業内で統一されたシステムを運用していたとしても、M&A や企業統合、新たなパートナー企業との取り引きが発生することにより、異なる運用形態のシステムと連携する必要もでてきます。

機能追加や方針変更などの要求が発生しても、短期間に対応できる柔軟性と高い生産性が求められます。それへの第一歩として、システムごとに実装されている共通機能を「サービス化」することで、他システムとの共同利用を可能とすることが求められます。

また、それらサービス化した部分を取りまとめ、運用方法を一元化するような仕組みも必要とされます。これは全社あげての一大プロジェクトとなる場合も想定する必要があります。

3.3.既存資産やサービスの再利用

「レガシー・マイグレーション」が注目されてから久しいですが、既存資産として抱えているメインフレームや旧来からのオープンシステムを活用しつつ SOA ベースのシステムに置き換えていくという要件は多くあります。いきなり全てのシステムを置き換えることは、価格・工数の面、リスク管理の面で難しいことも事実です。

既存資産を活用するために、それらのインタフェースを共通化するための手段や共通化したものを連携させるための仕組みが必要となります。

3.4.組み合わせた業務の稼働状況を見える化

企業内、および企業間でのシステムでは、状態に応じて必要となる業務を「プロセス」として段階的に実行していきます。これらの状態管理をアプリケーションやクライアント側で行うとなると、業務の組み合わせが複雑になればなるほど困難になります。

プロセスの制御や自動化、途中でエラーとなった場合でも処理結果を巻き戻すための後処理を効率よく実行してくれる環境が求められます。また、プロセスの変更・機能の追加が生じた際には、プロセスの再定義を行うだけで対応できるような仕組みが重要です。

3.5.高い信頼性

SOA に見られるシステム・業務モデルによって、企業内にとどまらない、インターネットを介しての業務共有化が進みます。そのため、共有利用している業務が停止することによる社会的な影響は大きく、ビジネスの機会損失も甚大なものになりかねません。

そのため業務アプリケーションが実際に動作するサービス実行基盤となる部分には高い信頼性が求められます。

4.提供サービス概要

WebOTXは、アプリケーション基盤としてさまざまなサービスを提供しています。ここではWebOTXが提供するサービスについて簡単に説明します。

4.1.Java EE

今日の企業では、ビジネス領域や組織を拡大させながら、そのコストを縮小、なおかつ顧客や従業員、サプライヤに対するサービスの応答時間を短縮させる事がますます重要になってきています。

サービスを提供するアプリケーションは、通常、広範囲のユーザにサービスを伝達する新事業の機能と既存の企業内情報システムとを組み合わせる必要があります。それらのサービスには、今日のグローバルな企業環境の要件を満たす高可用性と、ユーザのプライバシーと企業の情報資産を保護するための安全性、企業取引を正確かつ迅速に処理されることを保証するための信頼性と拡張性が必要です。

そのような企業サービスは、ほとんどの場合、多層の分散アプリケーション構成として実装されます。中間層では、既存の企業情報システムを新しいサービスのビジネス機能とデータに統合します。Web テクノロジーが成熟した現在、中間層の一部は分離され、サービスを受けるユーザがビジネスの複雑さから解放されて容易にアクセスできる層として利用されます。

Java EE (Java Platform, Enterprise Edition) は、多層アプリケーションとエンタープライズ領域のサービスにより開発場面で存在する高コストと複雑さの問題を解消します。Java EE アプリケーションは、企業が競争力を維持しながら、システムを迅速に展開し容易に増強することを可能にする能力を持ちます。

WebOTX は、Java EE 技術の中で Enterprise JavaBeans (EJB) の実行環境となる EJB コンテナおよび、サーブレット、JSP の Web アプリケーション実行環境となる Web コンテナを中心に全ての Java EE テクノロジーへ対応しています。開発したアプリケーションは、その移植性が保証されます。WebOTX の実行時環境について着目すると、RMI over IIOP (CORBA/IIOP による Java Remote Method Invocation の通信) は、高い相互接続性と世界最高レベルの通信処理性能を実現しています。トランザクション・サービスにおいては、旧バージョンよりも処理性能を数倍向上させると共に、コア・テクノロジーとなる分散トランザクション処理では業界で揺るぎない信頼性を維持しています。

さらに WebOTX Enterprise Service Bus では、Java EE の将来バージョンにおいて中核技術として組み込まれる予定の Java システム統合標準、JBI (Java Business Integration) 1.0 に商用製品として国内ベンダで初めて準拠しています。

4.2.Web サービス

これまで、COM や CORBA などのサービスが複数存在することで、各サービス技術間の連携が面倒なものとなっていました。このインターネットに分散しているサービスをもっと自由に結び付けるために「Web サービス」が生まれました。Web サービスにより、よりオープンなビジネスソリューションを実現することができます。各種サービスを SOA に基づいて柔軟かつ迅速に連携するための技術として利用が本格化されつつあります。

Web サービスは、データ交換のためのベースとして「Extensible Markup Language (XML)」を使用しています。また、XML データをやり取りするための決まりとして「SOAP」を使用しています。SOAP は、HTTP、SMTP、FTP などのプロトコルの上位に位置するため、既存のインターネット技術(プロトコル)に容易に乗せることができます。

4.3.CORBA

WebOTX は Object Management Group (OMG)が制定している Common Object Request Broker Architecture (CORBA)仕様に基づいた分散オブジェクト技術をサポートしています。CORBA で開発したアプリケーション(オブジェクト)は部品として再利用が容易です。また各種既存システムと連携するためのコネクタを提供しているため、既存のシステムをオブジェクト部品としてラッピングして再利用することも可能です。これにより、既存資産を生かした速やかなシステム構築を実現したり、システムの保守性を確保したりすることができます。これらオブジェクト指向による部品化技術と Windows や UNIX、Linux などの多様なプラットフォームへの対応により、急激な IT 技術の変化にも柔軟に対応することができま

す。WebOTX は CORBA 2.3/3.0(一部)仕様に対応しており、CORBA 2.0 以降仕様の他社製品に対する実証された相互接続性がありますので、マルチベンダー環境のシステムの統合も容易に実現できます。また、さまざまな言語でアプリケーションを作成することが可能なため、業務への適性とアプリケーション開発者の得手・不得手に対し、最適なものを選択することができます。

言語	クライアントアプリケーション	サーバアプリケーション
Java	○	○
C++	○	○
Visual Basic (※1)	○	―
COBOL (※2)	―	○
Holon (※3)	○	○

MEMO

※1:
CORBA ゲートウェイ, EJB ゲートウェイ (WebOTX Application Server Standard/ Enterprise Edition に添付) を使用
※2:
OpenCobolFactory21 (別途購入要)を使用
※3:
HolonEnterprise (別途購入要)を使用

4.4.COM/.Net フレームワーク

Microsoft 社の COM 技術に対応している WebOTX/COM という製品や CORBA ゲートウェイ および、EJB ゲートウェイを提供しており、COM や.NET Framework アプリケーションからシステムに接続することができます。これにより Visual Basic などの言語で容易にアプリケーションの開発を行うことができます。

WebOTX では、Visual Basic のクライアントアプリケーションから CORBA サーバアプリケーションの通信を実現する CORBA ゲートウェイと、Visual Basic のクライアントアプリケーションから EJB サーバアプリケーションの通信を実現する EJB ゲートウェイ機能を提供しています。

5.アーキテクチャ概要

5.1.製品構成概要

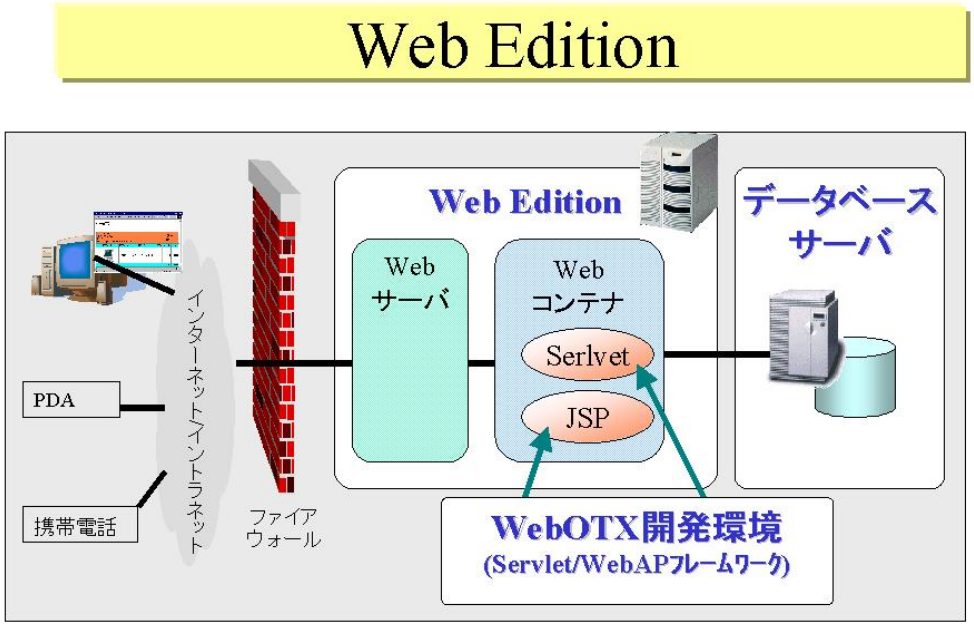
WebOTXが提供するアプリケーションサーバ製品には、システム形態、規模に対応して次の４つのエディションがあります。

- WebOTX Application Server Web Edition
  Web システムに業務アプリケーションを Java Servlet/JavaServer Pages (JSP)、Servlet Web サービス・エンドポイントとして追加する小規模システム向けのアプリケーションサーバ構成です。
- WebOTX Application Server Standard-J Edition
  Java platform Enterprise Edition (Java EE)に準拠した Java 標準構成のアプリケーションサーバです。Web Edition の提供機能を含んでいます。
- WebOTX Application Server Standard Edition
  Java EE アプリケーションだけでなく、C++言語や COBOL 言語のアプリケーションを動作させるなど、CORBA の特徴を生かしたシステムを構築できるアプリケーションサーバの標準モデルです。Standard-J Edition の提供機能を含んでいます。
- WebOTX Application Server Enterprise Edition
  全社レベルの基幹業務システムを構築できる高信頼、大規模システムに対応した構成のアプリケーションサーバ製品です。Standard Edition の提供機能を含んでいます。

次にそれぞれのエディションの特徴について説明します。

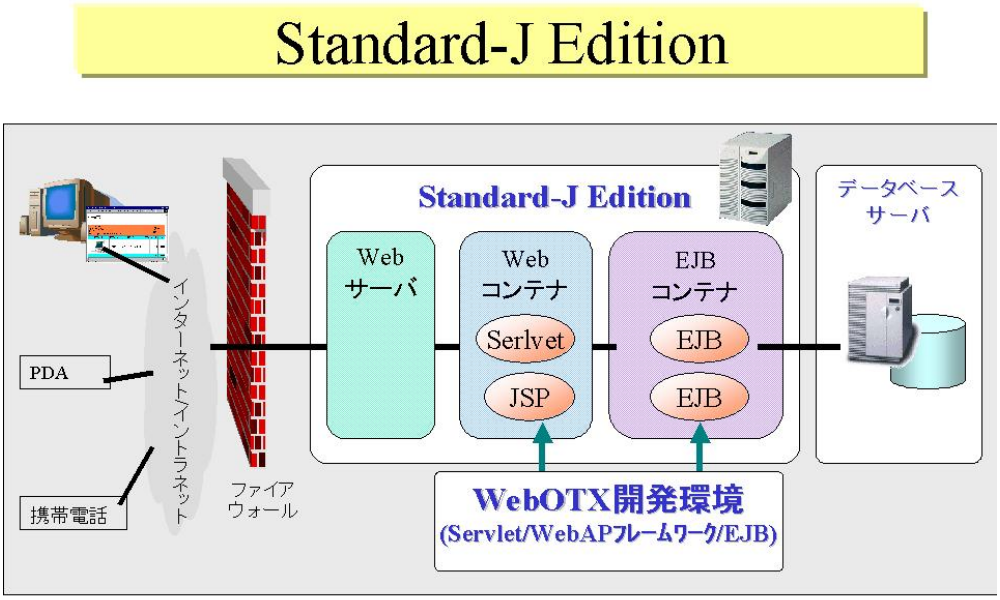
5.1.1.WebOTX Application Server Web Edition

Web ベースのシステムを短期間で構築したい。あるいは既存の業務処理を Web ブラウザから実行できるようにしたい。さらには既存の Web パッケージ製品を動作させるためにアプリケーションサーバが必要である。そのような場合に最適な製品です。



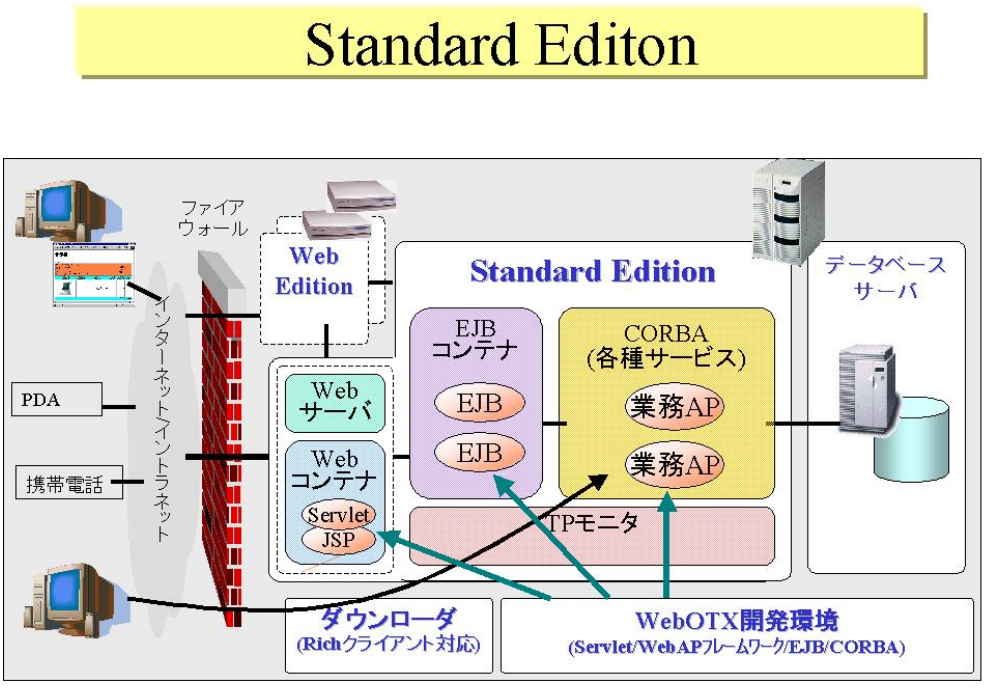
5.1.2.WebOTX Application Server Standard-J Edition

Servlet や JSP を用いた Web アプリケーションベースでのシステムを既に運用しているが、複雑なトランザクション制御処理をアプリケーションサーバが提供する、EJB などの Java EE 技術をベースにした本格的な Web 業務システムへの拡張を短期間に実施したい。そのような場合に最適な製品です。



5.1.3.WebOTX Application Server Standard Edition

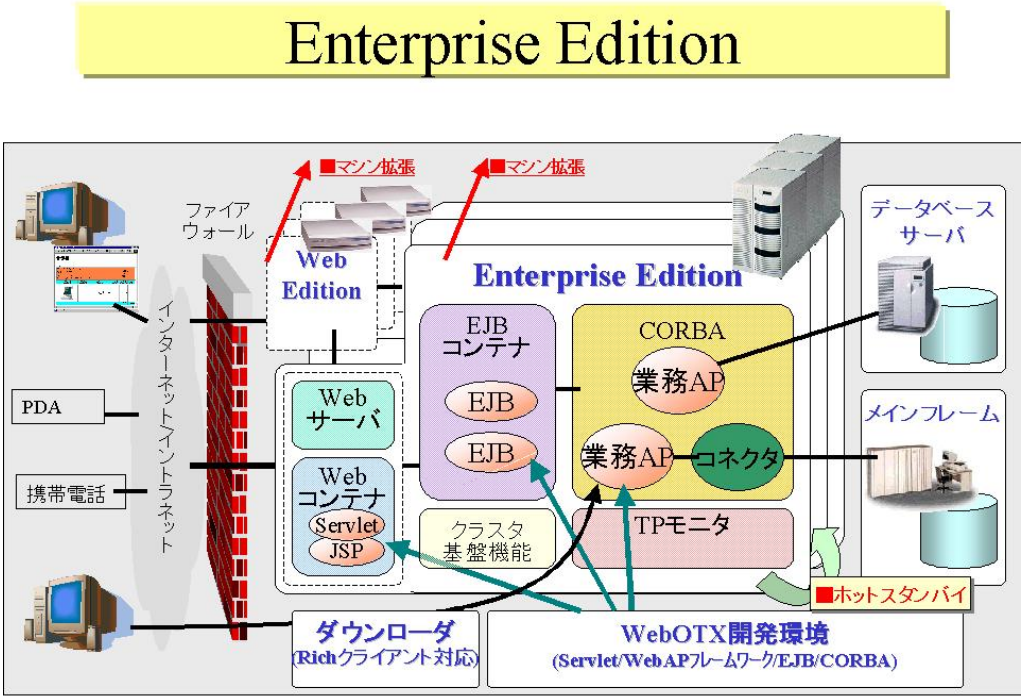
サービス品質を確保できる信頼性のあるシステムを構築したい。Java EE アプリケーションだけでなく、C++や COBOL アプリケーションが動作するシステムを構築したい。そのような場合に最適な製品です。



5.1.4.WebOTX Application Server Enterprise Edition

数千から数万、さらにはそれ以上のクライアントからのアクセスが生じるような大規模業務システム、または高い信頼性が要求される基幹業務システムの構築を考えている。そのような場合に最適な製品です。

多数のアプリケーションサーバマシンから構成されるシステムにおいて、各マシン間で負荷を分散・調整を実施します。また、あるサーバがダウンした場合でも処理を別マシンで継続できるようにすることでシステム全体としての停止時間を最小限に食い止めます。



5.1.5.WebOTX Administrator

WebOTX Application Server の下層では、Java Management Extensions(JMX)を基盤とする運用管理フレームワークの元でサーバのライフサイクルを管理しています。全ての管理リソースは JMX で規定されている Model MBean として定義しています。またリモートの運用管理クライアントからのアクセスを可能にするため、JMX Remote APIもサポートしています。これにより標準のインタフェースにより各リソースのコンフィグレーション情報の取得、設定、統計情報の取得、アプリケーションの配備、配備解除が行えます。システム運用管理を行うために、ブラウザベースの運用管理コンソール、コマンドベースの運用管理コマンド、運用管理 API を提供しています。

5.1.6.WebOTX Developer

WebOTX 製品では、高い生産性と保守性を持つ統合開発環境を提供します。

デファクトスタンダードな統合開発環境 Eclipse をベースに、NEC で独自に開発した Java EE 5 対応アプリケーション開発機能として Web アプリケーション開発環境や、Web サービス開発環境、EJB 開発環境、OLF/TP アダプタ開発環境を提供します。ソース編集機能や CVS による版管理、JUnit によるテスト環境、テスト用 AP サーバなどを兼ね備えており生産性を向上させることができます。

5.1.7.WebOTX SIP Application Server

WebOTX V7 より、Java EE Web アプリケーションサーバの高信頼かつ高いポータビリティの上に、新たにマルチメディアコミュニケーション技術のシームレスな融合を実現した SIP Application Server を提供します。

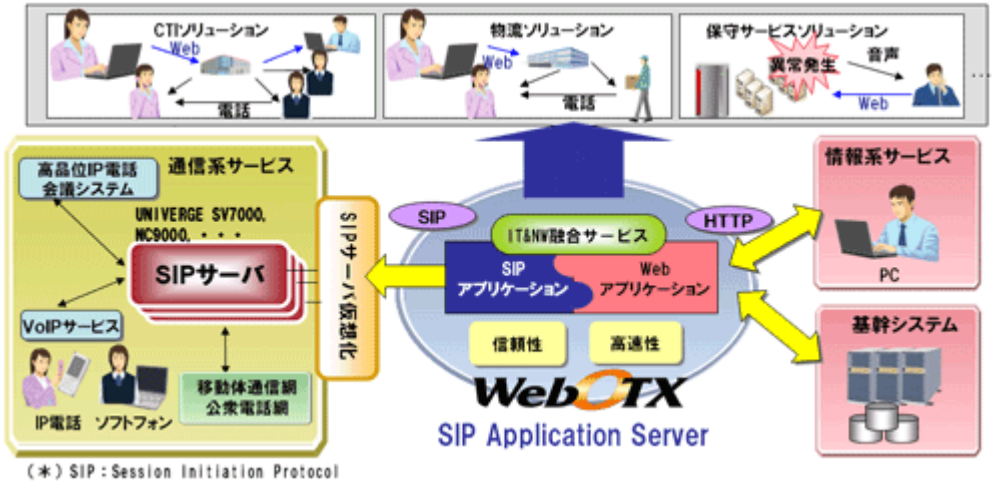
ブロードバンドインターネットの普及により、固定電話に代表された通信系サービスは、IP 電話やテレビ

電話などマルチメディアサービスへと進化しました。また、Web に代表される情報系サービスは、Java や Web2.0 技術、SOA により企業の業務における基盤となっています。

来るべき NGN・ユビキタス時代のネットワーク環境として、これらの通信系サービスと情報系サービスをインターネット上で連携し、お互いのメリットを活かした高付加価値サービスを迅速に構築することが求められています。

たとえば、企業や自治体においては、Web アプリケーションと音声・映像サービスを効率的に統合することで、お客様サービスの品質向上や付加価値サービスの拡大を図り、収益基盤の強化を実現したいというニーズが拡大しています。

また、通信事業者やISP/ASP 事業者でも、この技術と既存の通信インフラを融合させて、さらなるキラーサービスの提供を実現したいというニーズが拡大しつつあります。



5.2.提供機能概要

5.2.1.提供機能一覧

WebOTX の各エディションで提供する機能一覧を以下に示します。

カテゴリ	機能名称	Web Edition	Standard-J Edition	Standard Edition	Enterprise Edition
OS	Windows (x86、x64、IPF)	○	○	○	○
	HP-UX (PA-RISC、IPF)	○	○	○	○
	Solaris(SPARC)	○	○	○	○
	Linux(x86、x64)	○	○	○	○
	Linux(IPF)	○	○	×	×
Web サーバ	HTTP 1.0/1.1	○	○	○	○
	SSL 2.0/3.0、TLS 1.0 (※17)	○	○	○	○
Web サービス	SOAP 1.1/1.2	○	○	○	○
	WSDL 1.1	○	○	○	○
	UDDI 2.0/3.0	○(※1)	○(※1)	○(※1)	○(※1)
	WS-Security 1.0	○(※2)	○(※2)	○(※2)	○(※2)
	WS-Reliability 1.1	×	○	○	○
	WS-ReliableMessaging 1.0	×	○	○	○
Java SE	JDBC 2.0/3.0/4.0	○	○	○	○
	RMI-IIOP 1.0	○	○	○	○
	JNDI 1.2	○	○	○	○
	JAXP 1.3	○	○	○	○
	JMX 1.2	○	○	○	○
	JMX Remote API 1.0	○	○	○	○
Java EE	EJB 3.0	×	○	○	○
	Servlet 2.5/JSP 2.1	○	○	○	○
	JSTL 1.2	○	○	○	○
	JSF 1.2	○	○	○	○
	JMS 1.1	×	○	○	○
	JPA 1.0	○	○	○	○
	JTA 1.1	○	○	○	○
	JAXR 1.0	○	○	○	○
	JavaMail 1.4/JAF 1.1	○	○	○	○
	JCA 1.5	○	○(※3)	○(※3)	○(※3)
	JAX-WS 2.1(※18)	△(※4)	○	○	○
	JAX-RPC 1.1	○	○	○	○

MEMO

※1:
プライベートレジストリを利用するためには、WebOTX UDDI Registry を別途購入する必要があります。

※2:
SAML によるシングルサインオンを実現するためには、WebSAM SECUREMASTER V4.0 を別途購入する必要があります。

※3:
ACOS VIS と連携する場合には WebOTX OLF/TP アダプタを別途購入する必要があります。

※4:
EJB コンテナへの配備は不可

※5:
EJB 関連の操作は不可

	SAAJ 1.3	○	○	○	○
	StAX 1.0	○	○	○	○
	JAXB 2.1	○	○	○	○
	Java EE Management 1.1	△(※5)	○	○	○
	Java EE Deployment 1.2	△(※5)	○	○	○
	JACC 1.1	○	○	○	○
CORBA	CORBA 2.3/3.0(一部)	○	○	○	○
	Event サービス	○	○	○	○
	名前サービス	○	○	○	○
	インタオペラブル名前サービス	○	○	○	○
	Transaction サービス	○	○	○	○
	Notification サービス	×	×	○ (※6) (※7)	○ (※6) (※7)
	Persistent State サービス	×	×	○(※6)	○(※6)
XML	XML 1.0 Fourth Edition	○	○	○	○
	XML 1.1 Second Edition	○	○	○	○
	Namespaces in XML 1.1	○	○	○	○
	XInclude 1.0	○	○	○	○
	DOM Level 3	○	○	○	○
	SAX 2.0.2	○	○	○	○
	XML Schema 1.0	○	○	○	○
	XSLT 1.0	○	○	○	○
	XPath 1.0	○	○	○	○
オンライントランザクション処理	流量制御	×	×	○	○
	オブジェクトの事前生成	×	○	○	○
	マルチプロセス実行	×	×	○	○
	プロセス障害監視/閉塞	×	×	○	○
運用・実行制御	システム構成情報の設定・変更・追加	○	○	○	○
	動的多重度変更	×	×	○	○
	アプリケーションの登録・変更・削除	○	○	○	○
	アプリケーションの配備、配備解除	○	○	○	○
	サービス単位の活性・非活性	○	○	○	○
	Webベースの管理ツール	○(※8)	○(※8)	○(※8)	○(※8)
	バッチを含めた自動運転	○	○	○	○

MEMO

※6:
WebOTX CORBA 共通サービス準拠製品を別途購入する必要があります。

※7:
WebOTX Notification Service は次期バージョンでの提供になります。

※8:
きめ細かい運用管理を行ないたい場合は WebOTX Administrator を別途購入する必要があります。

※9:
WebOTX Cluster を別途購入する必要があります。

※10:
Open COBOL Factory21 もしくは Micro Focus COBOL を別途購入する必要があります。

※11:
WebOTX VIS Connector もしくは WebOTX OLF/TP Adapter を別途購入する必要があります。Standard-J Edition については、WebOTX VIS Connector を利用できません。

※12:
WebOTX Print Kit を別途購入する必要があります。

※13:
WebOTX Enterprise Service Bus を別途購入する必要があります。

※14:
WebOTX Process Conductor を別途購入する必要があります。

	診断情報(ログ)採取	○	○	○	○
負荷分散	ラウンドロビン	×	×	○	○
	重み付けラウンドロビン	×	×	○(※9)	○
	動的負荷分散	×	×	○(※9)	○
フェイルオーバ	スタンバイ	○	○	○	○
	高速スタンバイ	×	×	○(※9)	○
性能計測	性能分析用 API	×	×	○	○
	ログアナライザ	×	×	○	○
	プロファイラ	○	○	○	○
開発言語	Java	○	○	○	○
	C++	×	×	○	○
	COBOL	×	×	○(※10)	○(※10)
連携機構	既存システム連携	×	○(※11)	○(※11)	○(※11)
印刷機構	帳票印刷	×	×	○(※12)	○(※12)
SOA	JBI 1.0	×	○(※13)	○(※13)	○(※13)
	WS-BPEL 2.0	×	○(※14)	○(※14)	○(※14)
ディレクトリサービス	LDAP サーバ	○(※15)	○(※15)	○(※15)	○(※15)
映像・音声コミュニケーション連携	SIP Servlet API 1.0	×	○(※16)	×	×

MEMO

※15:
エントリ登録数が制限されています。制限を越えるエントリ数を登録する場合別途 Enterprise Directory Server を購入する必要があります。

※16:
WebOTX SIP Application Server を購入する必要があります。

※17:
WebOTX SIP Application Server V8.13 (Linux(x64)) の場合は、SSL 2.0/3.0、TLS 1.0/1.1/1.2 となります。

※18:
WebOTX SIP Application Server V8.13 (Linux(x64)) の場合は、JAX-WS 2.2 となります。

5.2.2.Web サーバ

インターネットの普及とともに World Wide Web (WWW)技術は急激に進展しています。当初、WWW は情報共有・情報発信を目的とした静的なコンテンツ配布が中心でしたが、今日では顧客との新しいチャネルや企業間取引き、外出する営業マンのアクセスポイントなどの動的なコンテンツを使った業務システムとして利用されています。アプリケーションサーバは、動的コンテンツを効率良く生成・処理する基盤として発展してきました。

従来、企業内の業務システムで多く使われてきたクライアント・サーバシステムはサーバコンピュータの負荷をクライアントにオフロードし、高い操作性を提供するというメリットがありましたが、技術変化が激しい今日では、クライアントのハードウェア・ソフトウェアの頻繁な入れ替えによる Total Cost of Ownership (TCO) の増大という新たな問題を発生させました。そのため、WWW 技術をクライアント通信に利用する WebComputing をベースとした業務システムの再構築が行われるようになっています。

HTTP サーバ

Web システムの中核を成しクライアント上の Web ブラウザから HTTP リクエストを受信し応答を返却するソフトウェアです。静的な HTML に対しては単独で Web サービスを提供することができます。WebOTX では Web サーバとして Java ベースの HTTP サーバと Apache Software Foundation の Apache HTTP Server 1.3.39/2.0.61 (※1) をバンドルしています。Secure Socket Layer (SSL) 2.0/3.0 および Transport Layer Security (TLS) 1.0 (※2) をサポートしており、インターネットを使った安全な通信を提供します。他にも次に示す Web サーバを利用することができます。

Web サーバ	Windows	HP-UX	Solaris	Linux
Microsoft Internet Information Server (IIS)	5.0、6.0	―	―	―

※1:
WebOTX SIP Application Server V8.13 (Linux(x64)) の場合は、Apache HTTP Server 2.2.34 となります。

※2:
WebOTX SIP Application Server V8.13 (Linux(x64)) の場合は、SSL 2.0/3.0、TLS 1.0/1.1/1.2 をサポートします。

Apache HTTP Server	1.3.39 以上 2.0.61 以上	1.3.39 以上 2.0.61 以上	1.3.39 以上 2.0.61 以上	1.3.39 以上 2.0.61 以上 (※1)
Sun Java System Web Server	6.1	6.1	6.1	6.1
Sun ONE Web Server	6.0 以上	6.0 以上	6.0 以上	6.0 以上

Web コンテナ

Web コンテナは、Web アプリケーションである Servlet/JSP を動作させるための実行基盤です。Web アプリケーションは、Web コンテナの提供する各種機能を使って、クライアントからの入力データを解析し動的な HTML を生成するサーバサイドの Java アプリケーションです。2 層のモデルでは Web アプリケーションはプレゼンテーション層からアプリケーション層(ビジネスロジック層)までを実現します。3 層モデルではプレゼンテーション層のみを実現します。

WebOTX の Web コンテナは、この Web アプリケーションの実行環境に加えて、Web コンテナの運用管理を行う機能を提供します。WebOTX の提供する Web コンテナは、Jakarta Tomcat 6.0 をベースに以下の機能を独自に強化しています。

- 性能向上
  Web コンテナに求められるのは何より性能です。Servlet/JSP の互換性を維持しつつ、可能な限りの性能向上を図っています。
- クラスタ対応
  トラフィックの増大に対応するために Web サーバマシンをクラスタ化することができます。クラスタ上の Web コンテナ間でセション情報とセション情報に格納されたデータの共有を行うことで、セション情報にアクセスする Servlet/JSP をどのクラスタ上でも動作させることができます。
- Web 版統合運用管理コンソール
  Web ブラウザからドメインの管理ができる運用管理ツールです。次のような操作が行えます。
  ー Web アプリケーションの管理(状態表示、閉塞、配備、配備解除)
  ー 統計情報の表示
  ー ログ、デバッグ情報設定
- OLTP モニタ上でのマルチプロセス実行
  Standard Edition/Enterprise Edition では、Web アプリケーションを動作させる Web コンテナを OLTP モニタ上でマルチプロセス動作することができます。これにより、Web アプリケーションについても、高信頼基盤である OLTP モニタ上での動作が可能となり、信頼性・可用性が向上し、ミッションクリティカル性が向上します。

SIP コンテナ

SIP(Session Initiation Protocol)は、インターネット上で音声通話やテレビ電話などのアプリケーションを構築するために考案された通信手順であり、IP 電話の普及によって注目されてきた技術です。NGN 社会への進展に伴いさらに需要が増すと考えられています。

WebOTX SIP Application Server では、多くの Web 開発者が標準アプリケーションインタフェースとして使用している「JavaServlet」を元にしたマルチメディアコミュニケーション用インタフェースである、「SIP Servlet API 1.0」に準拠。これにより、これまで音声・映像通信開発の経験のない Web アプリケーション開発者でも、容易に Web アプリケーションとコミュニケーションを連携させることができるようになります。

5.2.3.AP サーバ

EJB コンテナ

EJB コンテナは 3 層モデルにおいてビジネスロジックを Enterprise JavaBeans (EJB) アーキテクチャに従って実装した場合のコンテナ処理を行います。アプリケーションはセッション制御やトランザクション制御、セキュリティ制御などをなるべく意識しないで、ビジネスロジックに注力することが望まれます。EJB コンテナはこれらの制御をアプリケーション(Enterprise Bean と呼ぶ)に代わって実現します。WebOTX では EJB 2.1 仕様に準拠した EJB コンテナを提供しています。

WebOTX V5 以前の Standard-J Edition では、EJB コンテナの動作プロセスは 1 ホストにつき 1 プロセスに制限されていました。V6 からは 1 ホストに複数ドメインが設定可能となり、1 ドメインにつき 1 プロセス

で動作するように変わりました。

なお EJB コンテナは Standard-J Edition 以降の製品で提供されます。Web Edition では提供されません。

OLTP モニタ

業務システムを運用する場合に問題となるのは、システム自身の安定稼動です。特にインターネット業務システムでは負荷の予測が困難なため、過負荷時にも安定して動作できるアプリケーションの実行環境が求められます。

WebOTX はメインフレームで培われたオンライントランザクション処理(OLTP) 技術を取り込んだアプリケーションサーバです。OLTP 技術により、CPU、メモリなどのコンピュータ資源に対し、大量に発生するトランザクション要求を効率良く割り当て、高い並列性・独立性を保ちながら実行することができます。また、OLTP 技術でシステムの安定動作を実現しており、各種障害の検出、アプリケーションの実行プロセスの分離などの障害の局所化、デッドロック発生時の再実行などの各種のリカバリを実現しています。

インターネットで発生する想像を超えた負荷に対し、システムを恒久的に安定動作させることは困難であり、最終的には CPU、メモリ、サーバ、ネットワークなどのハードウェアの増設が必要になります。しかし、一般に増設は短期間で完了するものではなく、その間システムが不安定のままでは業務システムとして問題です。WebOTX はトランザクション毎のプライオリティ付けと流量制御により、資源の枯渇を防ぎながら、重要な業務を過負荷環境下でも安定して動作させることができます。さらに不要な業務を完全に閉塞することで、重要業務以外の負荷を無くしてしまうこともできます。

高負荷に対応するためにも、業務の実行基盤は高性能であることが求められます。WebOTX は世界最高水準の高速動作が可能な Object Request Broker (ORB)コアを採用しており、マルチスレッドやコネクションプーリング、ローカルスタブ(分散オブジェクトのプロセス内呼び出し)により、高性能なアプリケーションの実行環境を提供しています。

システムを安定して 24 時間 365 日稼動させるためにはハードウェアの多重化による冗長構成が必要です。一般にハードウェアは平均故障間隔(MTBF)で平均リカバリ時間(MTTR)停止すると考えられます。ネットワークやサーバ、ディスクなどを多重化して障害発生時にシステム全体に影響を与えるような Single Point of Failure (SPOF)を排除すると同時に障害発生時の影響範囲を局所化するようなシステム全体の設計が望まれます。

WebOTX は、Web サーバや Web アプリケーションサーバで多く使われている負荷分散型クラスタとメインフレームなどの業務システムで使われているフェイルオーバ型クラスタをサポートしています。負荷分散型クラスタは同一の構成のサーバ台数を増加することで簡単にアプリケーションサーバの可用性を高められ、性能もネットワークや共有リソースの限界まで拡張できますが、多数のサーバハードウェアを構成するとデータベースなどの共用リソースに対する障害発生確率を高くします。そのため、業務やアプリケーションで特定のデータ(レコード)へのアクセスの集中を避ける必要があります。一方、フェイルオーバ型クラスタは集中してアクセスされるデータに対して障害発生確率を低く抑えることができますが、フェイルオーバ発生時の切り替え時間はサービスを停止または保留させざるを得ません。WebOTX では業務特性に合わせて、これらを最適に組み合わせてシステムを構築することができます。

WebOTX ではシステムを安定稼動させるために、メインフレームで培われた OLTP 技術を Standard Edition と Enterprise Edition に取り込んでいます。これにより、サーバのコンピュータ資源(CPU、メモリ)を効率良く利用することができ、一時的なソフトウェア障害に対してもサービス全体を停止することなく安定してサービスを継続できるようにしています。Web Edition、Standard-J Edition は、サーバを並列して安定稼動を図るスケールアウトの考え方でシステム構築することに向いています。そのため、システムの軽量化を図って OLTP 機能は一部だけ取り込んでいます。ここでは WebOTX の Standard Edition、Enterprise Edition が提供する OLTP 機能について説明します。

マルチプロセス実行

WebOTX ではアプリケーションを複数のプロセスに分散して実行するマルチプロセス動作を行うことができます。各プロセスはシングルスレッドでもマルチスレッドでも動作させることができます。すべてのプロセス上のスレッド群は、単一のキューに対してデータの到着を待ち合わせるので、空きスレッドに効率良くトランザクション要求をディスパッチすることができます。

マルチプロセス実行をすることにより、アプリケーションの一時的な障害に対してサービスの継続が可能になります。データや環境、タイミングなどの要因でアプリケーション障害が発生しても、そのプロセスだけが異常終了し、他のプロセスは影響を受けません。したがって、システム全体ではサービスを中断することがありません。

流量制御

WebOTX では、アプリケーション毎に実行多重度(プロセス数、スレッド数)を設定することができます。クライアントから実行多重度を超えた要求があった場合、要求はキューイングされ空きスレッドができるまで処理を保留します。キューイングできる要求数もアプリケーション毎に設定できます。キューイング上限を超えた要求に対してはクライアントにエラーを返します。サーバの資源に応じて実行多重度、キューイング数を適切に設定することにより、想定外の大量のトランザクション要求に対しても、サーバリソースが枯渇して動作が不安定となることを回避し、リソースの範囲内で安定した動作を行わせることができます。またトランザクション毎にプライオリティ(実行優先度)を設定でき、高負荷で処理がキューイングされている状態においても、重要な処理は優先的に行うことが可能です。

オブジェクトの事前生成

WebOTX では効率的に処理を行うためオブジェクトの事前生成や再利用を行うことができます。ステートレスのアプリケーションやファクトリオブジェクトを事前に生成しておき、トランザクション要求発生時にコストのかかるオブジェクト生成をスキップして高い応答性を確保することができます。

プロセス障害監視/閉塞

WebOTX は AP サーバ上で動作しているプロセスの稼動状況を監視し、障害が発生したときには速やかにリカバリを行います。

- アプリケーション例外監視
  クライアントからのトランザクション要求に対してサーバアプリケーションが不正な処理を行い例外が発生した場合に、該当スレッドを閉塞し、例外が発生したことをコールバック API によりアプリケーションに通知します。アプリケーションは、データベースのロールバック前に障害情報の保存などの後処理を行うことができます。
- アプリケーション実行時間監視
  クライアントからのトランザクション要求に対して指定した時間が経過してもサーバアプリケーションが応答を返さない場合に、アプリケーションを異常終了させて該当プロセスを閉塞します。これによりクライアントに応答が返らない事態やサーバの資源を無制限に消費する事態を回避できます。
- プロセス異常終了監視とリカバリ
  アプリケーションプロセスが異常終了した場合、そのプロセスが使用していたリソース(コネクション、共有メモリ、データベース資源)を WebOTX が解放します。指定に従ってプロセスの再起動を行うことができ、障害発生後もそれ以前と同じ処理能力を維持することができます。

5.2.4.アプリケーションサービス

Web サービス

Web サービスは、「Extensible Markup Language (XML)」をデータ交換やインタフェース定義の手段として採用した標準技術、それらを使って提供されるコンテンツ(サービス)を総称したものです。XML を用いることで、人の手を介さないでプログラム同士が連携することも可能になっています。

Web サービスのもっとも基礎に位置づけられる標準技術は次のようなものです。

- SOAP

  アプリケーション同士が XML で会話をするための通信プロトコル仕様。
- Web Service Description Language (WSDL)

  Web サービスアプリケーションのインタフェースを XML で記述するための記述言語仕様。
- Universal Description, Discover and Integration (UDDI)

  Web サービスアプリケーションを探し出すための手段と、そこにプログラムを登録しておくための方法を定義した仕様。

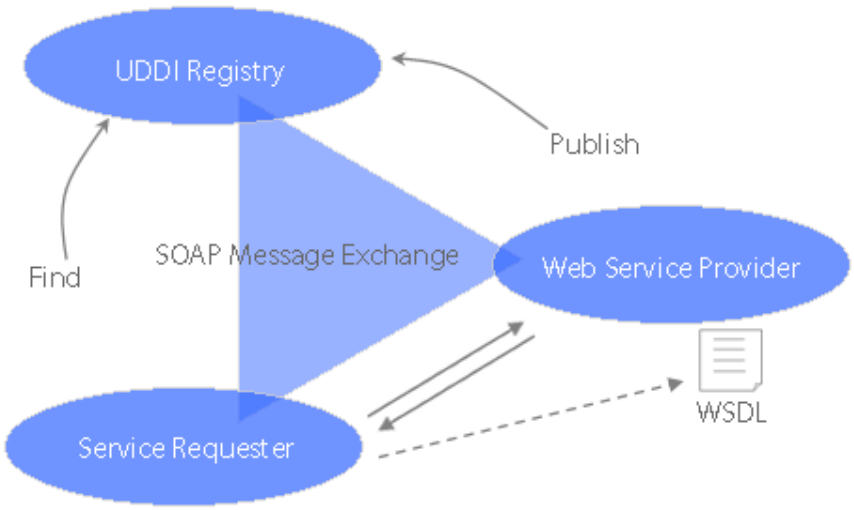


図.Web サービスの基本技術

また、これらの基礎技術を使ってセキュリティやトランザクションなどの応用技術を実現するために、「WS-〇〇」という名前の数多くの仕様の策定が日々進んでいます。

WebOTX は、EJB、CORBA、メインフレームなどで蓄積してきた従来のアプリケーション資産をスムーズに Web サービス化する機能を提供し、よりオープンなシステム構築を支援します。

SOAP 仕様、WSDL 仕様に準拠したエンジン、UDDI 仕様に準拠した UDDI レジストリと UDDI クライアント API を提供するとともに、Java EE に定義されている Web サービス技術、WS-Security で定義されている認証・XML 署名・XML 暗号にも対応しています。

ネームサービス（JNDI）

Java EE 環境下におけるネームサービスは、Enterprise Bean や Web コンポーネントに、JNDI ネーミング環境へのアクセスを提供します。

Java EE コンポーネントは、ネーミング機能とディレクトリ機能を持つサーバへのアプリケーションインタフェースとなる JNDI を使用して Java プログラミングします。任意の部品コンポーネントを JNDI で登録することで名前を指定してターゲットのクラスオブジェクトを取得することができ、アプリケーションのポータビリティの向上が図れます。すなわち、プログラムのポータビリティや再利用性を高めるためにも、汎用性が高いスタンダードレベルのアプリケーションから使うことが推奨されます。

WebOTX では、JNDI のサービスを提供するサーバ(JNDI サーバ)として CosNaming サーバ(CORBA 用)と Serial Context Provider (RMI 用)を提供しています。WebOTX Enterprise Edition を使うことで JNDI サーバ(CosNaming サーバ、Serial Context Provider)の二重化運用が可能になり、システム全体の可用性を確保することができます。WebOTX は JNDI 1.2 仕様に準拠しています。

JDBC データソース

トランザクションの実行性能を上げるためには、JDBC コネクションのプール管理によって、コストの高いデータベースとの物理的な接続および切断処理を、繰り返し行わないようにすることが重要です。

WebOTX では、JDBC のコネクションをプール管理するための JDBC データソースを提供しています。この JDBC データソースは、Web コンテナや EJB コンテナ上の Servlet や EJB から利用することができます。さらに、CORBA の Java アプリケーションからも利用することができます。

JDBC データソースは、JTA 仕様を実現する WebOTX Transaction Service と組み合わせることで 2 フェーズコミットメントにも対応でき、複雑に分散したデータベースにまたがるトランザクションをデータベースの一貫性を保障して簡単に実現することができます。

メッセージング・サービス（JMS）

WebOTX は、J2EE 1.4 で追加された Java(TM) Message Service(JMS) v1.1 に準拠したメッセージング・サービスを提供します。JMS とは、複数のオブジェクト間においてイベント通知を実現するサービスであり、キューやトピックと呼ばれる宛先を介して非同期に通信を行うことができます。 また、EJB 2.0 で導入されたイベント受信処理を行うメッセージドリブン Bean は JMS との連携により実現されています。利用者

は標準 API を使用して、メッセージ作成、宛先へのメッセージ送信、宛先からのメッセージ受信を容易に行うことができ、統一された構造のメッセージを抽象化された宛先を介してやり取りするため、システムとアプリケーションの結合を疎に保つことができるという特徴があります。

トランザクション・サービス （JTA）

WebOTX では、JTA 1.1 に準拠した分散トランザクション用のインタフェースを提供しています。この JTA インタフェースは、JNDI を使用して EJB や Servlet などの Java EE アプリケーションから簡単に利用することができ、さらに CORBA の Java アプリケーションから使用することもできます。JTA インタフェースを使用することで、複雑に分散して実行されるトランザクションの一貫性を簡単に保つことができます。

また JTA では、Java Transaction Service (JTS)のインタフェースを通して、CORBA Transaction Service (OTS) とも連携することができます。これによって、C++など他の言語で記述した CORBA アプリケーションや、X/Open XA インタフェースを提供するデータベース、MQ シリーズなどと一貫したトランザクション制御を行うことができます。

OTS の機能は、WebOTX の各エディションで提供しています。

EIS 接続サービス （JCA リソースアダプタ）

インターネット業務システムや企業内の業務システムを強化するために、既存の業務システムがカバーできていた部分もすべて作り直していたのでは、市場に対するスピードが問題となります。また、多くの場合、既存の業務システムは企業の戦略システムとしてデータベースや業務アプリケーションに有益な情報やノウハウが詰まっています。Enterprise Application Integration (EAI)は、このような基幹となっているエンタープライズ業務システム(EIS) を有効利用して新しいサービスを提供しようとする考え方です。企業内の複数のサブシステムを統合したシステムを構築し終えた後も、サブシステム毎に部分的な改良を続けながら、システム全体を強化していくことができます。

WebOTX は、企業システムで使われている既存の業務システムをオブジェクトに見せてラッピングする、JCA(Java EE Connector Architecture) 仕様の各種コネクタ製品を提供しています。既存の業務システムはまったく影響を受けないか、わずかな変更で新しいシステムの部品として使うことができるようになり、スピーディな新サービスの提供、システムの部分的な改良を現実のものとしています。

CORBA サービス

・ORB

WebOTX の ORB は CORBA が規定する Internet Inter-ORB Protocol (IIOP)を使用します。WebOTX では IIOP 1.2/1.1/1.0 をサポートしているので IIOP に準拠する他の ORB 製品と容易に通信ができます。また IIOP over SSL をサポートしているので IIOP の通信をセキュアに行うこともできます。 WebOTX は Java,C++とも CORBA 2.3/3.0(一部)仕様の言語マッピングに準拠した開発環境を提供します。通信に関する部分は ORB で管理されるので、アプリケーション開発者は通信部分をプログラミングする必要がありません。また Object Adapter には Portable Object Adapter (POA)を使用しているので他の ORB 製品からの移植もスムーズに行うことができます。

・名前サービス

名前サービスはネットワーク上に分散するオブジェクトリファレンスを一箇所に登録して、階層的な名前を付けて管理するための CORBA のサービスです。WebOTX では名前サービスを実現するサーバプロセスを提供します。このプロセスを名前サーバと呼びます。名前サーバではファイルシステムのように階層化してオブジェクトリファレンスを管理します。名前コンテキストはファイルシステムのディレクトリに相当するもので、このコンテキストが集まり木構造を作ります。名前コンテキストのうち、ファイルシステムのルートディレクトリに相当するものを特にルートコンテキストと呼び、ファイルシステムと同様に名前コンテキストの下に名前コンテキストを置くことができます。

・インタオペラブル名前サービス

WebOTX の名前サービスは CORBA 2.4 で規定されているインタオペラブル名前サービスに対応しています。そのため他社製 ORB 製品を WebOTX の名前サーバに登録したり、他社製の名前サーバへオブジェクトの登録、参照をしたりすることができます。

・Transaction サービス

Transaction サービスとはオブジェクト指向トランザクションを実現するためのサービスです。Transaction サービスを使用することで複数の WebOTX 上でデータの更新は安全に 2 フェーズコミット処理が行われます。一方のデータベースの更新に失敗した場合は Transaction サービスのリカバリ機能によりロールバックされるか、更新が正常に完了されるまでリトライされます。WebOTX で提供する Transaction サービ

スは、CORBA Transaction Service 1.4 に対応しています。そのため他の CORBA Transaction Service をサポートしているシステムと相互運用することができます。

また、ACOS とオープンサーバのリアルタイム連携を実現するためのランタイムライブラリ「ACOS Access Toolkit」が提供する JDBC ドライバと連携することで、本来は 2 フェーズコミットメントをサポートしていない ACOS 上のトランザクションを WebOTX が管理する 2 フェーズコミットメントトランザクションに参加させることができるようになっています。

つまり、これによって ACOS 上のデータベースと、オープンサーバ上のデータベース(Oracle など)の同時更新が可能になります。昨今のレガシーマイグレーション需要の増加に伴い、従来の業務系システムの一部をオープン化し、メインフレームとオープンシステム間でアプリケーション連携を実現する事例が増えています。例えば、従来のデータはメインフレーム上で管理、新規に追加された部分のデータ管理はオープンサーバ上で、という要件には適しています。

・Event サービス

標準的な CORBA 通信ではクライアントアプリケーションからサーバアプリケーションのメソッド呼び出しは同期呼び出しで行われます。すなわちクライアントアプリケーションは呼び出したメソッドの終了を待ち続けることになります。しかしながら、CORBA 通信を用い、非同期の通信を実現することも可能です。これを Event サービスと呼びます。Event サービスではサーバークライアント間で受け渡すデータを「イベント」と呼び、クライアントアプリケーション(サプライヤと呼びます)からのイベントを一旦「イベントチャネル」にキューイングします。キューイングが完了した直後にクライアント側には処理が戻ります。一方サーバアプリケーション(コンシューマと呼びます)にはイベントチャネルからそのイベントが送信されます。これによって非同期の通信を可能にします。

・Notification サービス

Notification サービスとは、前述の Event サービスをさらに拡張したサービスのことです。Event サービスが提供する機能のほかに、サービス品質の設定やフィルタリングといった機能を提供します。サービス品質の設定では、優先度が高い順にコンシューマにイベントを送信することや、イベントチャネルにキューイングされてから一定の期間が経過したイベントを自動的に廃棄すること、コネクション情報とイベントメッセージの永続性を保証して、システム障害などでイベントを失わないようにすることができます。また、イベントのフィルタリングを行うことで、コンシューマに送信するイベントの種類を限定することができます。

・Persistent State サービス

Persistent State Service (PSS) は CORBA サーバアプリケーション開発者のためのサービスです。アプリケーションを構成する Servant から Datastore domain (データベースなど)にアクセスするために利用するインタフェース internal interface を提供します。

WebOTX では Persistent State Service 2.0 仕様に準拠したデータベースを利用するサーバアプリケーションの開発環境を提供します。

5.2.5.Java ランタイム

WebOTX では、次の Java API やその実装、さらには開発基盤や Java API によって提供されるサービスをサポートしています。

Java 2 SDK, Standard Edition (J2SE)

WebOTX は Java の開発・実行環境として、J2SE 1.4.2 と 5.0 に対応しています。また WebOTX 製品媒体には、Sun Microsystems 社とのライセンス契約により Java 2 SDK v1.4.2 をバンドルしています。ただし HP-UX 版については再配布権の問題によりバンドルをしていません。別途 HP 社の Web サイトなどから入手してください。

http://www.hp.com/go/java/

他の OS (Windows、Solaris、Linux)で最新の JVM バージョンが必要な場合は、次の Web サイトから入手してください。

http://java.sun.com/j2se/

Servlet

サーバサイド Java プログラムとして、入力データを解析し、業務を行い、動的な HTML を生成するプログラムです。動的 HTML は、Servlet として作ると print メソッドが大量に発生してプログラムの可読性が下

がること、画面イメージの変更のたびに影響を受けることなどから、現在では JSP で記載されることが多くなっています。

WebOTX の Web コンテナは Servlet 2.5 仕様に準拠しています。

JavaServer Pages (JSP)

Servlet で記述したように、動的 HTML の生成部分を可読性が高いように HTML にスクリプトを埋め込む形で Java プログラムを記述できるようにしたものです。サーバサイドの Java アプリケーションプログラムすべてを JSP で記述することもできますが、逆にビジネスロジックを実現するプログラムや HTML 部分の可読性が落ちてしまうので、動的 HTML 生成部分のみを JSP で記述することが望ましいと考えられています。JSP は実行時にコンパイルされて Servlet プログラムとして動作します。

WebOTX の Web コンテナは JSP 2.1 仕様に準拠しています。

JavaBeans Activation Framework (JAF) / JavaMail

JavaBeans Activation Framework (JAF)は、データの種類(MIME タイプ)に応じて、適切な JavaBeans を呼び出す仕組みを提供する API です。Windows で実現している、ファイル拡張子の関連づけなどと同じように、ファイル名から種類を判別して「編集」、「閲覧」などのメニューを表示し、その種類に対応した JavaBean を起動することができます。JavaMail でも利用されています。

JavaMail は、Java プログラムから電子メールを発信するためのアプリケーションインタフェースです。非同期で結果を通知するなどの利用が可能です。JavaMail は Java VM が動作すれば動作しますので、Web コンテナ、EJB コンテナどちらでも動作させることができます。

WebOTX では、JAF 1.0.2 と JavaMail 1.3.1 について Sun のリファレンス実装を提供しています。

Java API for XML Processing (JAXP) / Xerces

Java API for XML Processing (JAXP) は、Java で XML 文書を扱うための標準的な API です。XML 文書を扱うための標準的な API である DOM や SAX では、プログラミング言語に依存しない仕様になっているため、実際にプログラミングを行うときに使用する API という形式ではありません。JAXP は、Java 言語で XML ドキュメントの読み書きや構文解析をするときに使用する API を提供します。これにより、DOM、SAX、XSLT、XPATH などを駆使して XML の処理をすることができます。

JAXP は、アプリケーション・コードを変更することなく、任意の XML プロセッサを変更できます。WebOTX では、Apache XML Project が提供する JAXP 実装モジュールである「Xerces」と「Xalan」をバンドルしています。

Enterprise JavaBeans (EJB)

Enterprise JavaBeans (EJB) は、コンポーネント・ベースのビジネス・アプリケーションの開発と適用(配備)のためのアーキテクチャです。EJB 2.1 では、開発と配備の場面に Web サービスの利用を仕様に加えました。一般的なサーバアプリケーションに必要な機能である、トランザクション管理、セキュリティ管理、コンカレンシ制御などは EJB コンテナと呼ばれる実行環境が提供します。このためサーバアプリケーション開発者は、それら下層レベルの複雑な知識を必要とせず、ビジネスロジックの記述に集中することができます。

EJB で開発したサーバアプリケーションは、一度記述すれば、EJB 仕様をサポートする、どのサーバ・プラットフォームにでも配備し実行することができます。

EJB 仕様には、サーバ側の Enterprise Bean コンポーネント・タイプとして次の 3 種類があります。

- Session Bean
- Entity Bean
- メッセージドリブン Bean

Session Bean はクライアントと対話する、永続性を持たないオブジェクトです。Session Bean は、さらにステートフル Session Bean とステートレス Session Bean の 2 種類に分けられます。ステートフル Session Bean はクライアントとの対話の状態を保持します。つまりステートフル Session Bean とクライアントとは一対一の関係になります。ステートレス Session Bean はクライアントとの対話の状態を保持しません。

Entity Bean は、データベースのような永続的な記憶メカニズムに対応する業務オブジェクトを表わします。永続化の方法によって Entity Bean は、Bean による永続化とコンテナによる永続化の 2 種類に分類され、それぞれ Bean Managed Persistence (BMP)と Container Managed Persistence (CMP)と呼ばれます。BMP では EJB 提供者がデータを永続化するコードを EJB 実装クラスに記述する必要があります。CMP ではコンテナがデータを永続化するため、配備情報に永続化を行うフィールド名やプライマリ・キー

のクラス名などを定義するだけで永続化を行うことができます。

メッセージドリブンBeanは、非同期のメッセージ・コンシューマであり、メッセージの到着と共にコンテナから呼び出されます。コンテナから呼び出されるため、他の 2 つの Bean タイプとインタフェースを持たない点で異なります。

クライアントは、メッセージング・サービスにメッセージを配信することから始まります。メッセージング・サービスが JMS の場合、ディスティネーションとしてトピックかキューのいずれかを選択します。JMS サーバに着信したメッセージは、次に EJB コンテナ内のメッセージドリブン Bean へメッセージ送信します。その結果、メッセージリスナのメソッドがコールバックされる形で該当するメッセージドリブン Bean にメッセージが着信します。

EJB 2.1 仕様では、次の点が強化されました。

- ステートレス Session Bean を Web サービスのエンドポイントとして実装できる
- コンテナ管理によるタイマーサービスの提供
- EJB QL への ORDER BY と集約関数の追加
- JMS 以外の他のメッセージング・タイプと連携可能なメッセージドリブン Bean

JDBC

Java プログラムからのリレーショナル型データベースへの標準アクセスインタフェースです。WebOTX には JDBC ドライバが含まれていません。データベースベンダ製品、もしくは JDBC ベンダ製品の、JDBC ドライバを使用してください。基本的に、JDBC 2.0 または、JDBC 3.0、JDBC 4.0 の仕様に準拠している JDBC ドライバであれば、使用することができます。

動作確認済みの製品およびバージョンについては、セットアップガイドの「使用上の条件」、または提供機能ガイドの「5.3.7. データベースと JDBC ドライバの対応バージョン」をご参照ください。

5.2.6.XML

XML パーサ

XML パーサは、アプリケーションから XML 文書を取り扱うために必要な手段を提供します。一般的に、XML パーサは次のような機能を提供します。

- XML 文書の読み込み、書き出し
- XML 文書が整形式(Well-Formed)かどうかのチェック
- XML 文書の妥当性(Valid)チェック
- DTD や XML スキーマなどのスキーマ文書の読み込み、書き出し

XML 文書の読み込みには、DOM や SAX といったプログラミング言語に依存しない API が存在します。これらを含めて、Java 言語から XML パーサを利用するための標準的な API として JAXP があります。

WebOTX は、XML パーサとして Apache XML Project が提供している「Xerces2」をバンドルしており、JAXP にも対応しています。実際に XML パーサを利用する場合には、JAXP で定義されているインタフェースを利用されることを強く推奨いたします。

XSLT プロセッサ

Extensible Stylesheet Language Transformations (XSLT)は、主に XML 文書の構造変換を行うための仕様です。XSLT スタイルシートに変換規則を記述し、それを XML 文書に関連付けることで違った構造の XML 文書に変換することができます。また、XML 文書だけでなく、HTML やタグのないテキスト形式にも変換することができます。

XSLT プロセッサは、XML と XSLT スタイルシートを読み込んで変換を行い、結果を出力する機能を提供します。XSLT スタイルシートの中では、XML 文書中の特定のノードを指定するために XML Path Language (XPath)を使用します。XPath は式(expression)によって XML 文書中の特定のノードを指定するための方法です。

WebOTX は、XSLT プロセッサおよび XPath に関する機能を実装している Apache XML Project の「Xalan」をバンドルしています。

5.2.7.ネットワーク通信

RMI over IIOP

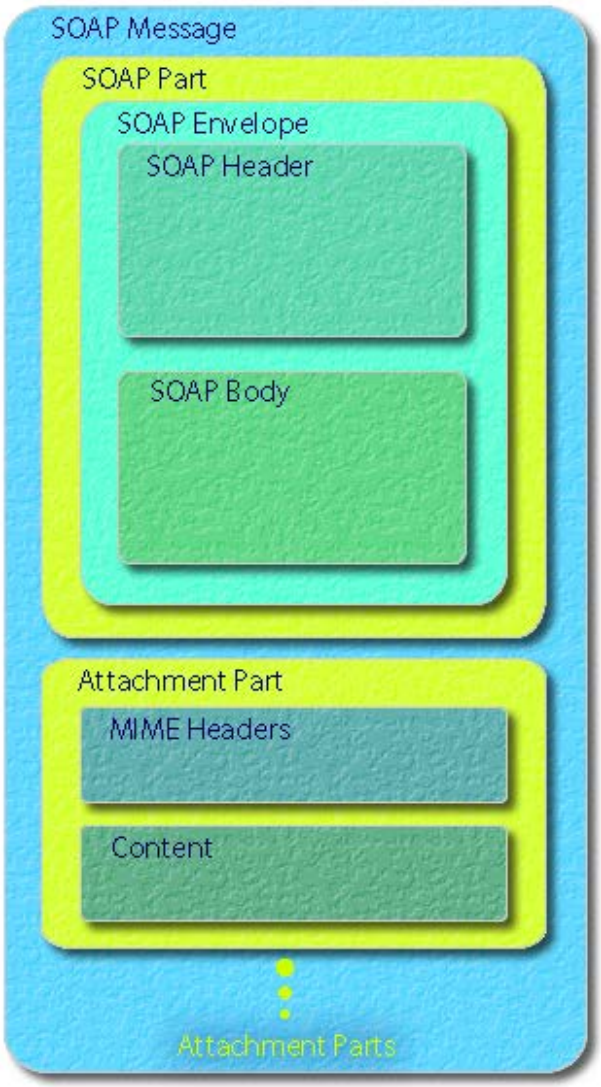
従来、Java のプログラム間通信手段としては RMI が使われていましたが、Java EE が要求するトランザクション・セキュリティ技術に対応するには独自プロトコルによる拡張が必要であったため、ベンダ固有の通信となってしまい相互接続性がありませんでした。RMI over IIOP は、これらの問題を解決し、CORBA の高い相互接続性と Java 言語への親和性の高い通信を両立するものです。

WebOTX では、WebOTX Object Broker (従来製品名称：ObjectSpinner)という高性能な ORB の通信コアを含んでおり、RMI over IIOP 1.0 仕様に準拠しています。RMI over IIOP による高い相互接続性と世界最高水準の通信性能を実現しています。

SOAP

SOAP は XML 文書を交換するためのプロトコル仕様です。HTTP や SMTP などの既存のプロトコルに乗せて使うことが想定されている上位プロトコルです。図に SOAP メッセージの構造を示します。

- SOAP Message
  HTTP のヘッダなども含めた SOAP メッセージ全体を指します。
- SOAP Part
  XML で書かれるメッセージ本体の領域を指します。
- Attachment Part
  添付データの領域を指します。Attachment Part は、複数の領域をあとに続けて確保することができます。
- SOAP Envelope
  SOAP メッセージの一番外側の要素です。
- SOAP Header
  SOAP 通信や Web サービスのアプリケーションが動作する上で必要な付帯事項を書き込む要素です。
- SOAP Body
  通信したいメインの内容を書き込む要素です。
- MIME Headers
  添付データの MIME エンコーディングに関するヘッダを書き込む部分です。
- Content
  MIME エンコーディングした添付データを書き込む部分です。



5.2.8.セキュリティ

インターネット業務システムでは、外部からの不正侵入に対するセキュリティが重要です。また、イントラネットの業務システムでも、社員 ID の一元管理などの Single Sign On 技術が要求されています。

WebOTX では、各種のセキュリティ製品との連携でセキュリティの高い業務システムを構築できます。例

えば、ファイヤウォール製品との連携が容易にできるようにサーバ側のポートの固定化や、インターネット上に仮想の専用線を構築する Virtual Private Network (VPN)を実現する製品 SOCKSVPN との連携、インターネットで最も多く使われている暗号化技術 Secure Socket Layer (SSL)を使った CORBA 通信、HTTP 通信、などができます。

アプリケーションは、セキュリティ技術を意識しないか、標準のセキュリティアプリケーションインタフェースを使うことができますので、将来、セキュリティ技術が変更されても、柔軟に対応することが可能です。

5.2.9.運用管理

インターネットを介するシステムでは、ファイヤウォールサーバ、Web サーバ、Web アプリケーションサーバ、業務アプリケーションサーバ、データベースサーバ、ストレージサーバ、ネットワーク機器など多数のハードウェアやソフトウェアが使われます。これらの装置で使われる技術や機能は多様なため、各サーバのシステムを容易に管理できることは運用コスト削減のためにも重要です。

WebOTX では、Java Management Extensions (JMX)を基盤とした運用管理フレームワークを提供しており、全ての管理リソースは JMX で規定されている Model MBean として定義しています。また、システム運用管理を行うために、ブラウザベースの運用管理コンソール、コマンドベースの運用管理コマンド、運用管理 API を提供しています。

運用管理ツールのウィンドウ上でシステムの稼動状態を簡単に監視することができ、マウスを中心とした操作により、システム構成情報の作成・変更・追加、アプリケーションの登録・変更・削除、起動・停止、などの運用操作を簡単に行うことができます。また、自動運用を行えるようにスクリプトで記述できる運用コマンドを提供しており、時間指定の業務の開始・終了や、障害発生時に業務運用に合わせたリカバリを自動化するなどの設定が容易に行えます。さらに該当システム特化した運用管理ツールを JMX の API を利用して開発することもできます。

また WebOTX は、統合運用管理ソフトウェアである WebSAM と連携することで、システムの負荷状況(CPU 負荷率、メモリ使用率、ディスク使用率、LAN のトラフィックなど)を統合的に監視し、障害(ハードウェア、OS、ネットワーク、WebOTX システム)をいち早く検出することができます。

次に WebOTX の運用管理の特長について説明します。

マルチドメイン管理

WebOTX では業務システムをドメインとして管理します。業務システムはその規模により 1 つのドメインで構成させることもできますし、また複数のドメインで構成させることもできます。ドメインは物理的な 1 つのホストにマッピングされるのではなく、複数のホストで構成されるシステム全体の中で存在する 1 つの構成単位(インスタンス)となります。よってシステムがクラスタ環境である場合、1 つのドメインが複数のホストを移動したりすることができ、また 1 つのホスト上に複数のドメインを配置することも可能です。

システム管理ツール

WebOTX では業務システムを容易に構築し、運用管理を行うためにさまざまなシステム管理ツールを提供しています。システム管理ツールにより、マルチドメインで構成された業務システムを一元的に運用管理することが可能になります。

- GUI ベース統合運用管理ツール

  マルチドメインで構成された WebOTX システムを一元的に運用管理を行なうための GUI ツールです。コンフィグレーション、アプリケーションの配備、状態監視、障害監視など全ての運用操作がこの統合運用管理ツールで行なうことができます。

- ブラウザベース運用管理コンソール

  Web ブラウザを利用して管理するツールです。Web ブラウザと FlashPlayer プラグインが動作する環境があれば、リモートから簡単に管理が行えます。

- コマンドベース運用管理コマンド

  WebOTX ではオペレータを介することなく自動で運用を行うためのバッチファイルやシェルスクリプトで使用できるコマンドを提供しています。これにより自動でアプリケーションの展開、サービスの開始、停止、設定の変更などを行うことができます。

- 運用管理 API

WebOTX では JMX Remote API インタフェースを提供しています。これにより該当システムに特化したコマンドや運用管理ツールを JMX の API を利用して開発することができます。

MBean

WebOTX では、全ての管理リソースは JMX で規定されている Model MBean として定義しています。これにより次の操作を JMX インタフェースで行うことが可能です。

- ドメインの作成、削除
- ドメインの開始、停止
- サービスの開始、停止
- コンフィグレーション情報の取得、設定
- 統計情報の取得
- リソースの登録、削除
- アプリケーションの配備、配備解除
- JMX Notification の取得

JMX コネクタ

WebOTX では、JMX Remote API で規定されている次のプロトコルに対応したコネクタを提供しています。

- JMXMP コネクタ
- RMI コネクタ

アプリケーションの配備、配備解除

Java EE では、アプリケーションの登録・削除を配備(Deployment)、配備解除(Undeployment)と呼びます。WebOTX では、GUI ベースの配備ツール(従来製品名称：J2EE 展開ツール)を用いて AP サーバ、Web サーバへアプリケーションを配備・配備解除できます。これにより容易に開発環境から実行環境のサーバにアプリケーションの展開が行えます。またオペレータ操作を介することなく自動で配備・配備解除するためのコマンドも提供しており、シェルスクリプトやバッチファイルに記述することができます。

5.2.10.アプリケーション開発

業務システムの構築において益々重要度が高くなっている要素の一つとして、短期間開発、があります。市場の変化が激しい現在において、業務システムを市場や企業戦略に合わせてスピーディに変更・強化していくことが望まれます。

WebOTX では、サーバアプリケーションの開発言語として Java、C/C++、COBOL、Holon を使うことができ、クライアントアプリケーションの開発言語として Java、Visual Basic、C/C++、COBOL、Holon を使うことができます。これらの開発言語は、業務への適性とアプリケーション開発者の得手・不得手から考えて最適なものを選択することができます。それぞれの言語で使用する統合開発環境(IDE)製品は、プラットフォームに依存して最適なものを選択できます。

現在 WebOTX では、Java 言語の開発は WebOTX Developer’s Studio を、Visual Basic や C/C++の開発では Microsoft 社の Visual Studio を推奨しており、これら IDE 上で容易に WebOTX 上のアプリケーションを開発できるようにするプラグインを提供しています。COBOL は Open COBOL Factory 21 を、Holon は HolonEnterprise を使うことで、統合された開発を行うことができます。

WebOTX では、オブジェクト指向などの新技術に対する開発者の負担を下げるために、各種開発支援ツールを提供しています。Rich クライアント・アプリケーションを自動生成するクライアント AP や Flash 通信ライブラリ、Thin クライアント用のプレゼンテーションロジックを生成する WebAP、各種コネクタを自動生成するコネクタ開発環境を提供します。データベースアクセスについては、Persistent State Service や Enterprise JavaBeans の Entity Bean を使うことで WebOTX が自動的に行い、開発者はビジネスロジックに専念することができます。

5.2.11.負荷分散

WebOTX ではスケーラビリティを向上するためにアプリケーションサーバ、Web サーバを複数台並列に配

置した負荷分散運用を行うことができます。負荷分散運用を行うことにより、大量のトラフィックに耐え得る安定したシステム運用が可能になります。また負荷分散運用を行うことによりサーバ障害時にも縮退運転を行いシステムに対する影響を最小限にすることができます。

負荷分散装置による負荷分散

負荷分散装置を利用して、Web サーバの負荷分散を行うことが可能です。現在では、負荷分散装置も多種多様なものが出回っており、これらを利用することで複数台の Web サーバを負荷状況によって切り分けることができます。セッションを維持して同じクライアントからのリクエストは、常に同じ Web サーバで処理することも可能です。また、障害が発生した場合も、動作可能な Web サーバを自動的に選択します。負荷分散の方法は、単純なラウンドロビン方式や、重み付けを付加したものなど、装置によりさまざまな方法が提供されています。

5.2.12.性能計測

WebOTX ではシステムの稼動状態を確認するため統計情報を提供しています。

MBean 統計情報

WebOTX では稼動状況を把握するための各種統計情報を MBean として実装しています。この統計情報を JMX API やコマンドを利用して取得することができます。

運用管理ツールを通して、リアルタイムに稼動状況を確認することができます。また閾値を設定すると、閾値を越えた時点で JMX Notification 通知を受けることができます。

ジャーナル

ジャーナルは WebOTX の AP サーバの稼働状況を評価するための性能及び統計情報を各種レポートとして提供します。

一日を通した、AP サーバの稼動状況(トランザクション件数、レスポンス時間、CPU 時間)を確認することができます。

イベントジャーナル

サーバでの処理の流れをチェックポイント毎に解析できます。これにより各オペレーション処理の詳細な流れを追うことができます。

ストールや CPU ループの解析、レスポンス遅延の要因の究明に利用することができます。

プロファイラ

JVMPI を利用した GUI ベースのプロファイリングツールを提供します。

CPU プロファイル、ヒーププロファイル結果が容易に確認することができ、Java アプリケーションの性能ネックとなっている箇所やメモリ消費の多い箇所を調べることが出来ます。