

ミドルウェア：CSシリーズの戦略

Middleware: An Introduction

Peter Bye

1. はじめに (Introduction)

数多くのカンファレンス、セミナー、記事および書籍がミドルウェアという主題に対して、議論したり、記述したりしているという事実が、ミドルウェアがIT分野の大きな関心であるということを示している。しかし、ミドルウェアというものは一体なにものなのであろうか。ミドルウェアはどのような問題を、どのように解決するのか、またミドルウェアは多少なりとも問題を解決することを要求されているのだろうか。ユニシスのミドルウェアに対する姿勢はどのようなものか。本稿では、ミドルウェアを良く知らない読者に対して、これらの疑問への答えおよび、さらなる情報への指針を提供できればと思う。

本稿は、通常の企業および非営利組織が面している重要な問題の一つである、変化の速度およびその予測不能性、を考察することから始める。

2. 変化という問題 (The problem of change)

ビジネスの世界で働いている誰もが、この10年ほどの間の変化の速度に対応することが困難であったことを、確信しているであろう。変化は一定に進む、というようなことは通用しない時代になって来ている。そこには多くの理由があるが、技術における劇的な開発が、その中で最も重要である。マイクロ・エレクトロニクスにおける革命はコンピュータ・システムに大規模な改良をもたらした。また、マイクロ・エレクトロニクスは、光ファイバーやその他の開発と相まって、複雑なネットワーク・システムの構築を、以前よりも容易な作業にした。いまやインターネットは成年期に達している。

技術開発は、新しい作業形態を可能にし、既存の作業形態の経済性を変え、新しい作業形態をより広範囲のグループの人達が利用できるようにした。創造的な人達が新しい方法で技術を適用し、技術を供給する側に対し、さらに進んだ技術の提供を要求し、可能な限りの高度な技術が供給される、というある種の好循環が展開している。

最も顕著な技術の出現の一つがeビジネスの成長である。eビジネスは新しいアイデアではなく、古くは、航空業界におけるトランザクション処理や、銀行間のコミュニケーションにB2B (Business to Business) の起源を発する。技術開発の中でも、特にインターネットはeビジネスという選択肢をさらに多くの企業に広げた。かつては銀行、航空会社あるいは政府機関といった大組織に限定されていたものが、今ではこれからスタートしようとしているような会社でも利用可能なのである。しかし、一昨年(2000年)のドットコム企業のバブルがはじけている様は、いかに技術が進んでいようとも、ビジネスがリスクなしではない、ということを示している。

変化の速度による問題は営利企業に限ったことではなく、政府機関、部門は納税者から多くの税収入を得ようとしているし、自身もより効率的な政府となるよう求められている。政府機関、部門は国民を政府への請願者としてではなく、より良い便利なサービスを提供することにより、顧客として対応することが要求されている。さらに公営企業は、B2Bによるビジネスを強力に推進する民間企業から物を購入している。政府およびEUなどの組織は、かねてから民間企業との間のビジネス・コミュニケーションはeビジネス技術を使用することを指示してきている。

3. 変化への対応：ITへの要求 (Responding to change: the IT requirement)

将来、企業の大多数はITに依存して行くことは間違いなく、効果的なIT対応能力が企業の成功に必要であり、ITアプリケーションおよびITインフラ(基盤)は変化の圧力に対応できなければならない。単純化し過ぎるかもしれないが、企業のITシステムは次に挙げる二つの要求を満たすことができなければならない。

- 1) 企業の内外に向けて新しいチャネルを容易に追加できること。チャネルの接続先は、個別の消費者レベルであれば顧客一人一人であり、あるいは営利、非営利企業である。民間企業であれば、サプライヤー、金融機関、提携企業などがその例であり、公営企業/組織であれば国税、関税、物品税などである。
- 2) ビジネスの機会への窓がたとえ小さくとも、新しくアプリケーションを作成し素早く対応できること。これは、既存のビジネスの運営に妥協することなしに実行できなければならない。

企業はこのような難問にどのように応えることができるのだろうか。ドットコム企業のような新興企業は別にして、eビジネスの要求に合ったデザインの全く新しいシステムを、ゼロから導入できるほどの贅沢さを持つ企業は非常にまれである。典型的なIT環境では、長年に渡って開発してきた数々の基幹システムを、念頭にあるパフォーマンスのような他の目標と妥協させている。企業の合併、買収はビジネス革命の結果といえるが、それにより、異機種システムの適合といった、複雑さがすぐさま増加する。

新しいビジネスに対応するために、基幹システムのアプリケーションの書き換え、入れ替えといったことを考慮したくなるような誘惑にかられる。しかし、これは非常に危険なアプローチといえる。なぜなら、生産性の向上が真に劇的でないのならば、新アプリケーションは、単にやりかたを変えただけになってしまうからである。さらにソフトウェア業界が約束するものを提供してきたかという、到底そうとは思えず、アプリケーションの書き換えは失敗する可能性が高い。どんなに作業リソースを注ぎ込んでも、新アプリケーションの能力向上になんらの利益を生み出さないかもしれないのである。

書き換えよりも良いアプローチは、新機能と組み合わせた既存アプリケーションの再利用である。こうすることにより、既存システムやデータベースを新しい方法で使えるという、全体的な機能性の向上を提供する。システムは特定の問題に対応した方法を使っても統合できるし、統合時の問題は将来のことを考慮しなくとも解決できる。そうすると、次の問題がまた同じように近づいてくる。つまり繰り返してである。こういった方法はある時

点まではうまく働くが、いつかは管理不能な複雑さを増加させるに他ならない。こういった問題の結果として、企業は上手に構成された IT インフラがどうしても必要であるという認識にたどり着くのである。インフラは新しいチャネルおよびアプリケーションの追加を支援すべきでもあり、新しいチャネルおよびアプリケーションは既存アプリケーションと統合できるし、利用もできる。

ミドルウェアはインフラ構築へのキーである。企業は、ミドルウェアの土台の上に、要求されたインフラを構築するために「フレームワーク」あるいは「技術アーキテクチャ」を定義しているのである。

4. ミドルウェアとは/ミドルウェアはなにをするものか

(Middleware: what it is and what it does)

ミドルウェアはアプリケーションを構成する異なるコンポーネント間の通信を行うインフラを提供する。コンポーネントは同一の物理システム内に存在することもあれば、異なるシステム上に存在することもある。各アプリケーション・コンポーネントはお互いの存在する場所を知ることはなく、必要とするコンポーネントの探索はミドルウェアが担当する。この定義は Chris Britton 氏の提唱する「ミドルウェアは分散システムを実際に構築することを可能にするソフトウェアである」*1 という定義に一致する。

ミドルウェアというものは新しい概念ではない。私が知っている限りでは、この言葉が最初に使われたのは、British European Airways 社 (BEA、といってもミドルウェアの先進ベンダーである BEA Systems 社と混同しないように) の IT 部門においてであり、それは 1970 年頃であった。BEA 社のその部門は、ミドルウェアと呼ぶソフトウェア一式を開発した。ソフトウェアの目的は、トランザクション処理のアプリケーション開発を、上手に定義されたコンポーネントから構築でき、実行時にはお互いに共同して動作するようにすることにより、以前よりも数段簡単にすることであった。

ミドルウェアには様々な種類があるが、ここでは次のように分類する。

アプリケーション実行環境.....この分類における技術および製品は、前述したミドルウェアの定義に適したサービスおよびツールを提供するものであり、分散アプリケーションの構築、特にトランザクションシステムの構築を可能にする。それらは

- 1) トランザクション・サーバー
- 2) メッセージ・キューイング
- 3) 単純な、下位レベルの技術

である。

最初のグループは、分散トランザクションを支援する必要な機能を持った、コンポーネント・ベースのアプリケーションの開発を支援するものである。四つの主要な標準に対応した製品が、様々なベンダーにより開発されているので、それらを以下に記述する。

- 1) Open Group Distributed Transaction Processing (Open Group DTP, 旧 X/Open DTP).....この標準は業界標準ではなく真にオープンである。BEA Systems 社の Tuxedo の開発によって、この標準の起源が作成され、Unisys 社からは OS 2200 お

よび MCP/AS 上で稼働する Open/DTP が対応し、ICL 社等の他社からも製品が出されている。Open Group DTP は手続き型プログラミングモデルに基本をおいている。

- 2) Distributed iNternet application Architecture (DNA).....マイクロソフト社からの標準であり、以下に挙げる二つの標準と同じくオブジェクト指向プログラミングモデルに基づいている。DNA はマイクロソフト社の最近の進化である Component Object Model (COM) : Windows 2000 では COM + の開発から生まれた。NET はインターネットおよびその他のネットワークの間で広く分散するアプリケーションの協調を支援する目的で、現在マイクロソフト社が力を入れて展開している方向性である。
- 3) Java 2 Enterprise Edition (J2EE).....この標準は Sun Microsystems 社から出された。この標準に沿った製品は非常に多く、BEA Systems 社からの WebLogic Server がこの製品である。
- 4) CORBA Component Model (CCM).....Object Mangement Group (OMG) から提出されたオープンな標準である。OMG はオブジェクト指向技術を標準化しようとする多くの組織を纏め上げる目的で作られたグループである。CORBA(Common Object Request Broker Architecture) は何年にも渡り発展してきて、異なるレベルの仕様が提唱されている。CCM は、特にトランザクション処理を目指しているが、筆者の知る限りでは CCM を実装した主要ベンダーは現時点ではないようだ。

2 番目のグループに挙げられているものが Message Queuing Middleware (MQM) である。MQM を使用してお互いに通信しているということを除けば、アプリケーションは全く独立である。必要なことは通信上のメッセージの形式と内容だけである。さらに、MQM は通信する要素を全く非同期に操作することを許している。例えば、送信側のアプリケーションがメッセージを投げる時に、受信側のアプリケーションが実行されていなければならない、という必要がない。ミドルウェアが必要とする中間の記憶域を用意するからである。

メッセージ・キューイングには様々な標準があるが、最も知られていて共通な標準は IBM 社の MQSeries である。IBM 社のシステムは当然として、MQSeries は Unisys 社を含む多くの他社のシステムに実装されている。

最後のグループは下位レベルの技術を含み、単純な RPC (Remote Procedure Call) などがそのグループに属す。ある範囲では、このような下位レベルの技術はミドルウェアとは見なされていないが、分散システムの構築に使うことができるし、また実際に使われている。このような技術は正しい環境で使われると効果的である。

アプリケーション・サーバー.....アプリケーション・サーバーはアプリケーション実行環境製品の一つとしてよく位置付けられる。アプリケーション・サーバーは、ほとんどが Web ベースのアプリケーションを構築するためのツール類や機能を提供する。例えば、BEA Systems 社の WebLogic Server (WLS) や Unisys 社の Internet Commerce Enabler (ICE, 旧 Cool ICE) 等がそうである。

実行環境統合.....いくつかの異なるミドルウェアの標準があることや、標準として認め

られているミドルウェアに準拠しないアプリケーションがあるために 統合を目的とする、異なる標準を接続することができる技術および製品への要求がある。このような製品には、Unisys 社の Transaction Integrator や BEA Systems 社の eLINK がある。

他の技術分野でも同様であろうが、どのような分類にも上手く適合しないものがあるというのは常である。これはミドルウェアにもあてはまるのであって、ミドルウェアというラベルがつけられている製品ではあっても、上記の分類にきちっと適合しないものは存在する。しかしながら、本稿のミドルウェアの紹介という目的からすると、この分類で十分であろう。

5. ミドルウェアの使用 (Using middleware)

アプリケーション開発およびビジネス・チャネルの伸展の中で発生する問題の解決に、どのようにミドルウェアが役立っているのだろうか。初めに、新規のアプリケーションの構築を考えてみよう。ミドルウェアは、アプリケーションの実行環境を提供する基礎として使うことができる。

図 1 はこの環境を示したものである。

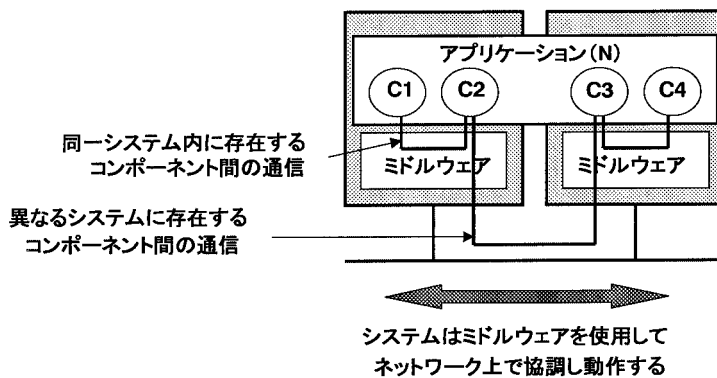


図 1 新規のアプリケーションでのミドルウェアの使用

図 1 には二つのシステムが示されている。アプリケーション(“アプリケーション(N)”)は両システムに渡って配置された四つのコンポーネント(C1~C4)を実装している。コンポーネントは、様々なアプリケーションの機能、代表的なものではトランザクション処理、を協同して実行する。C1とC2はミドルウェアを介して同一システム内で通信し、C3とC4はもう一つのシステム内で通信している。またC2は別のシステム内のC3と通信をしている。C2とC3の通信はネットワークを介さなければならない。すべてのケースにおいて、コンポーネントはシステムにローカルなミドルウェアとインターフェースを行うだけであって、ミドルウェアが他のコンポーネントの検索と接続の開始を担当する。この時、必要であればネットワークを介する。すべてのコンポーネントは、アプリケーションの変更なしに再配置が可能である。例えば、C2はC3とC4が配置されているシス

テムに移動可能である。

図1で示したミドルウェアは**アプリケーション実行環境**に分類されるであろう。代表的なものとしてはトランザクション処理サーバーが挙げられる。多くのケースで、追加の補助的なミドルウェアの必要性があるだろう。例えば Open Group DTP の実装は即時に「要求/応答」する通信に使われるだろう。また、メッセージを後で処理するために、他のシステムにメッセージを書き込むというような、異なった通信を必要とするアプリケーションもあるだろう。このようなアプリケーションにはメッセージ・キューイングが理想的な提案である。例えば、ATM（現金自動受払機）は口座の残高紹介や現金払い出しには、即座に応答することが要求される。しかし、取引一覧表等の文書を後で郵送したり、電子メールで送付してもらうような場合には、要求メッセージが即座に処理される必要がない。要求メッセージは文書印刷を実行するシステムに置かれるだけでよい。

図1ではミドルウェアを使用し、新規のアプリケーションを実施する方法を示しているといえる。この種の技術は今日のアプリケーション開発の好ましいアプローチである。異なる技術は、各々の技術の中で「コンポーネント」という言葉が使われるに連れて、「コンポーネント」により明確な意味を与えている。少なくともコンポーネントのいくつかは再利用可能であろうという理由から、アプリケーション開発がより早くできるという期待が上がってくる。しかし、ことは思ったようには単純ではないにも関わらず、アプリケーションをコンポーネントとして開発することは多くの魅力がある。

しかしながら、前に議論したように、企業はある異なったモデルを使用した既存のIT環境を持っているのが一般的である。このような企業は新しいアプリケーションを実現しようと望むかもしれないが、問題は、どうすれば既存のアプリケーションに新しい機能を追加し、新しいチャネルを開けるかである。複数の既存のアプリケーションからデータとサービスを統合しながら、新しい機能を開発することが要求される。以上のように、既存のアプリケーションを新しいモデルで実行できるように書き換えることは、非常に大きなリスクを伴う。ミドルウェアは統合をどのように支援できるのだろうか。

図1で示した構造は、ミドルウェアの標準に準拠していない他のシステムを包含することができる。図2はどのようにこれらの他のシステムを包含するかを示している。

図2は、前に図1で示したように、コンポーネントC1~C4を使いアプリケーション(N)を実行する二つのシステムと、アプリケーション(E1)とアプリケーション(E2)を実行する新しい二つのシステムを示している。この二つのアプリケーションは全くミドルウェアを使って書かれていないか、いくつかのミドルウェアを使っているかも知れない。いくつかのケースでは、アプリケーションをミドルウェアを使う環境内で実行するよう変換できる。また他のケースでは、図に示したようにアプリケーション・ゲートウェイを通してアクセスできる。このようなゲートウェイはミドルウェアからコンポーネントとして認識されるので、他のコンポーネントから呼び出すことも可能である。その機能は要求をアプリケーションに理解できる形式に変換することであり、アプリケーションを、ミドルウェアの標準にしたがって書かれたコンポーネントを持っているように、見せかける。このプロセスは**ラッピング**あるいは**カプセル化**として知られている。

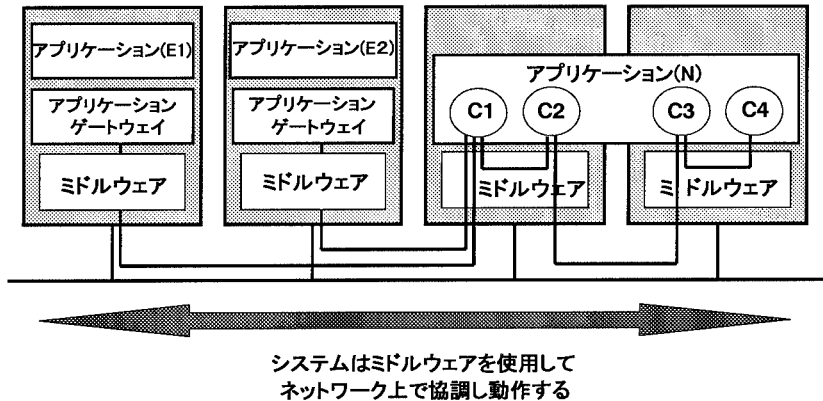


図 2 アプリケーション統合におけるミドルウェアの使用

ゲートウェイは、この章の前で議論したように、**実行環境統合**から出てきた典型的な技術であり製品である。例えば、Unisys 社の Transaction Integrator はこのような変換を行い、様々なミドルウェアが Unisys の OS 2200 および MCP/AS 環境下で実行するアプリケーションにアクセスできるようにする。Transaction Integrator は、実行環境が限定されたアプリケーションへのアクセスを提供するものである。その他のゲートウェイプロダクトには BEA Systems 社の eLink も含まれる。eLink は異なる環境への様々なゲートウェイを提供する。同様にマイクロソフトも異なったアプリケーション環境への多くのゲートウェイ製品を提供している。

図 2 は異なった環境で実行されるアプリケーションが、ミドルウェアおよびゲートウェイを使用して、どのように相互に動作するかを示している。次のステップでは、どのようにミドルウェアが新しいアクセスチャネルの追加を助けていて、アクセスチャネルを通して新しいサービスを提供するために、どのように追加アプリケーション・ロジックが既存のシステムの統合に使われるかを提示する。図 3 はこれがどのように達成されるのかを示している。

図 3 の右側には図 1 および図 2 と同様にアプリケーション (N) を実行する二つのシステムがある。ここにおけるミドルウェアをある特定の技術を支援する“ミドルウェア 1”と呼ぶことにする。図 2 のもう二つのアプリケーションも図 3 に存在するが、アプリケーション (E1) はミドルウェアを使わないという点が多少変わっている。例えば、Unisys の TIP あるいは COMS を使っているかもしれない。アプリケーション (E2) はミドルウェア 1 と異なるミドルウェア 2 を使用していることとする。例えば、ミドルウェア 1 がマイクロソフト社の COM+ であり、ミドルウェア 2 が Unisys 社の Open/DTP である。

図 3 には前に挙げた図 1 および図 2 にはない“アクセスと統合”と名付けられた新しい要素を含んでいる。“アクセスと統合”は一般的には**アクセス/統合サーバー**と呼ぶ別のシステムに実装するであろう。その機能は、一方では Web ブラウザ、ワークステーション、WAP 電話、他システムなどのアクセスチャネルへの接続であり、もう一方では二つの間のギャップを橋渡ししてアプリケーションに接続することである。単純ではあるが、その

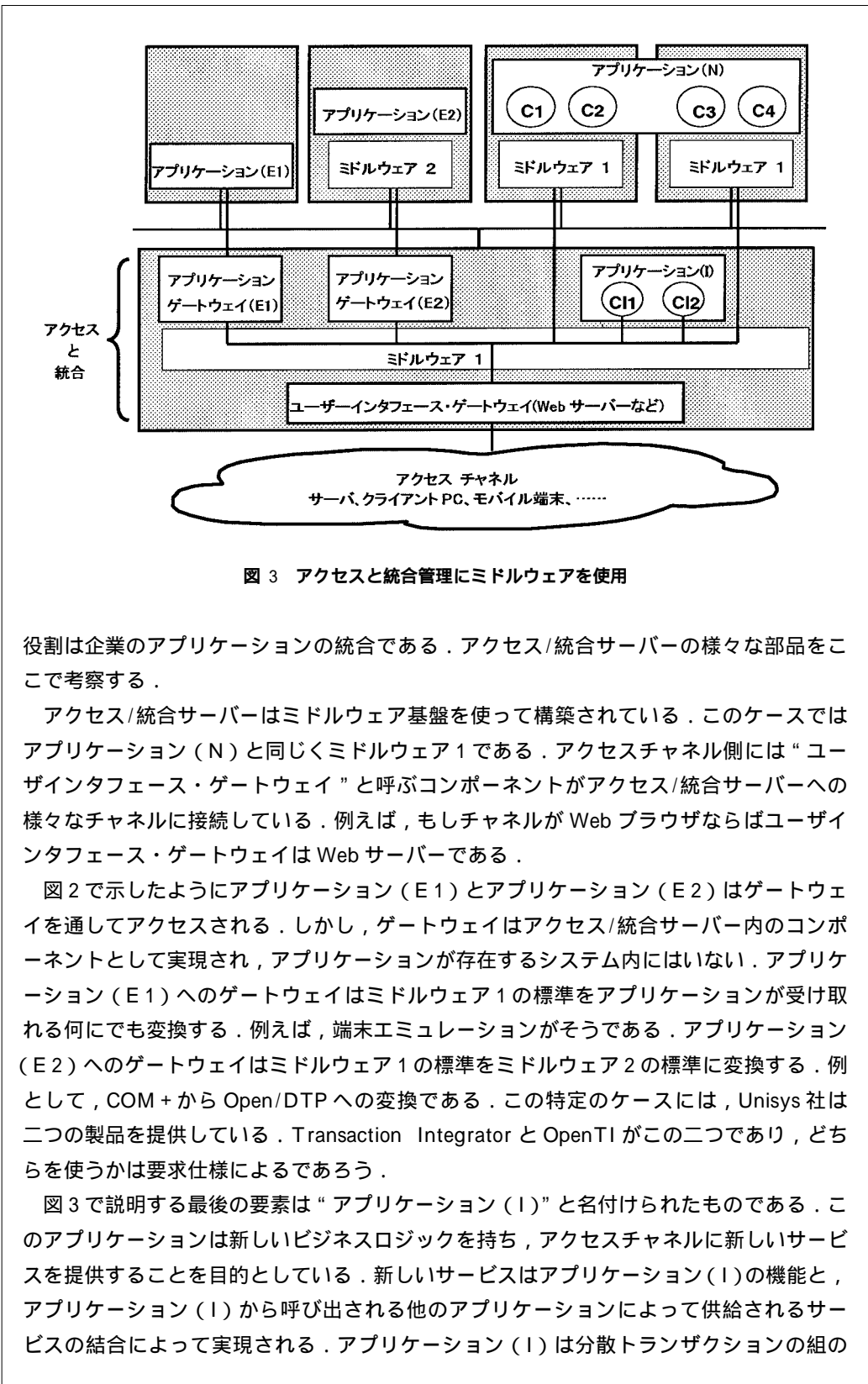


図 3 アクセスと統合管理にミドルウェアを使用

役割は企業のアプリケーションの統合である。アクセス/統合サーバーの様々な部品をここで考察する。

アクセス/統合サーバーはミドルウェア基盤を使って構築されている。このケースではアプリケーション(N)と同じくミドルウェア1である。アクセスチャンネル側には“ユーザーインターフェース・ゲートウェイ”と呼ぶコンポーネントがアクセス/統合サーバーへの様々なチャンネルに接続している。例えば、もしチャンネルがWebブラウザならばユーザーインターフェース・ゲートウェイはWebサーバーである。

図2で示したようにアプリケーション(E1)とアプリケーション(E2)はゲートウェイを通してアクセスされる。しかし、ゲートウェイはアクセス/統合サーバー内のコンポーネントとして実現され、アプリケーションが存在するシステム内にはない。アプリケーション(E1)へのゲートウェイはミドルウェア1の標準をアプリケーションが受け取れる何にでも変換する。例えば、端末エミュレーションがそうである。アプリケーション(E2)へのゲートウェイはミドルウェア1の標準をミドルウェア2の標準に変換する。例として、COM+からOpen/DTPへの変換である。この特定のケースには、Unisys社は二つの製品を提供している。Transaction IntegratorとOpenTIがこの二つであり、どちらを使うかは要求仕様によるであろう。

図3で説明する最後の要素は“アプリケーション(I)”と名付けられたものである。このアプリケーションは新しいビジネスロジックを持ち、アクセスチャンネルに新しいサービスを提供することを目的としている。新しいサービスはアプリケーション(I)の機能と、アプリケーション(I)から呼び出される他のアプリケーションによって供給されるサービスの結合によって実現される。アプリケーション(I)は分散トランザクションの組の

開始地点となる。図示されるように、アプリケーション (1) はミドルウェア 1 の標準内のコンポーネント (CI 1 および CI 2) として実現されている。アプリケーション (N) もまたミドルウェア 1 を使用するので、CI 1 と CI 2 はミドルウェア 1 を通じてアプリケーション (N) と通信する。アプリケーション E 1 および E 2 は図に示したようにアプリケーション・ゲートウェイを通してアクセスする。

アクセス/統合サーバーはミドルウェア・プラットフォームに加え様々なゲートウェイを使って構築できる。ミドルウェア・プラットフォームはアプリケーションの環境にもよるが、**実行環境統合**に分類されるミドルウェア製品か、またはあつらえて構築した製品である。さらに、アプリケーションサーバーは、通常は Web ブラウザや WAP 電話のようなインターネット技術といった機能に合わせてデザインされている。これらの製品はミドルウェアを基礎とし、様々なセキュリティ機能、開発援助やその他のサービスを提供する。Unisys 社の Internet Commerce Enabler や BEA Systems 社の WebLogic Server (WLS) がその製品の例である。

多くの会社がミドルウェア環境に**付加価値製品** (Value added products) と呼ばれるものを提供している。彼等は業界を水平に横断し、あるいは市場を垂直に貫いて、特定のビジネスの問題を解決する製品をパッケージ化し販売している。そのために、それらのソフトウェア会社は要求に合うように、ミドルウェアのフレームワークの中にアプリケーション・コンポーネントを開発している。図 3 におけるアプリケーション (1) がそれである。

代表的な例がワークフロー管理アプリケーションであり、必要に応じたデータ変換機能を備え、B2B の協同作業のような特定の問題解決を目指すソリューションである。さらに、例えば XML を使用する特別なチャネルへのゲートウェイが開発されている。有名な SAP および Siebel へのゲートウェイも同様である。完成されたソリューションはアプリケーション実行環境あるいはアプリケーション・サーバーに加え、アプリケーション環境統合、付加価値アプリケーション、ゲートウェイなどを含むものになるのであろう。結果として図 3 に示した形態を取るであろう。

いくつかの場合には、ベンダーは全ての部品を含んだ完全なソリューションを提供するであろう。これは BEA Systems 社の製品の提供方法や、IBM 社が MQSeries を MQSeries integrator のような製品と共に提供することに代表される。ベンダーは、他のベンダーが開発した部品を統合して、さらに部品を開発する。これは、多くのベンダーがソリューションの部品を提供し、さらに他のベンダー達がそれを統合して行く、ということの意味している。

ミドルウェアのような動きの速い技術においては、企業はお互いの製品を統合したり、企業合併、企業買収を進めたりと、結果として通常的に Web を介しての関係にシフトしている。前にも述べたように BEA Systems 社や IBM 社は、このような範疇にある多くの会社の中でも抜き出ている。その他、webMethods、BowStreet、Vitria および Bluestone (現在は HP 社の子会社) などなど多くの企業がある。Unisys の Internet Commerce Enabler (ICE) は Unisys のパートナー企業である英国の ICS 社の開発した製品であり、い

くつかの特定のアプリケーションに使用されてきた。

まとめると、以上の議論は、ミドルウェアはIT組織に面している問題の解決を助ける、つまり新しいアプリケーション機能およびアクセス・チャンネルをできるだけ効果的に提供する方法、である、ということを示すためのものであった。新しいアプリケーションは、**アプリケーション実行環境**製品のの一つを使用して、ミドルウェアというインフラの中のコンポーネントとして開発されるものであろう。ゲートウェイは、**アプリケーション環境統合**の一つを使用して、既存システムとの統合を支援する。そして新しいチャンネルの追加および既存アプリケーションの拡張は、アプリケーション・サーバー製品を基礎に、**アクセス/統合サーバー**のコンセプトを使って実現することができる。最後に、特定のビジネスの問題を解決することを狙った包括的な付加価値製品が数多く生まれてきている。

6. ユニシスの戦略とサーバー (Unisys strategy with its servers)

ユニシスのサーバー戦略は、現在最も要求の強い大量トランザクション処理を照準にしたハイエンドのシステムを提供することである。ClearPathシステムは、このような現状を踏まえ、ユニシスの2200シリーズおよびAシリーズの開発で培ってきた長い間の経験をもとに開発された。一昨年の2000年にはユニシスはCellular Multi Processing (CMP)を開発した。CMPはコンピュータ・システムを構築するための**アーキテクチャ**であり、いわゆる製品の名称ではない。最初のCMPアーキテクチャを実装したシステムはES7000シリーズとして2000年に市場にリリースされた。さらにCMPを基盤とした最新のClearPathサーバーである、CS7802システム(2200シリーズ系のハイエンド機)、およびCS7101システム(Aシリーズ系の中/小型機)を昨年の2001年に提供を開始した。将来、ユニシスから提供されるサーバーは、全てCMPアーキテクチャを基本として行く予定であり、本年2002年に提供予定のAシリーズのハイエンド機であるCS7801システムもCMPアーキテクチャを基盤としている。

これらのシステムは、本稿の中で述べてきたビジネス世界の様々な変化および不確実性に、対応できなければならない。ミドルウェアは、本稿で説明しようとしたように非常に重要な役割を果たす。したがって、ClearPathシステムおよび他のサーバーにおいて、主要なミドルウェアを支援し、お互いに協調してシステムが動作し、ミドルウェアを使用しないで書かれたシステムを統合することは基本的なことである。

NT/Windows 2000/Windows XPあるいはUnixの下で稼働するES7000および他のサーバーは使用可能なミドルウェア製品の全てを支援する。一方、ClearPathシステムに関してのUnisysのミドルウェア戦略は、OS2200およびMCP/ASオペレーティング・システムの下で実行するキーとなる一揃えの技術を開発あるいは取得することである。ミドルウェア技術はミドルウェアを取り巻く世界の状況の中で日々変化しており、留まることがない。したがって新しい製品の出現は、古い製品に取って代わり、古い製品は消えて行くことになる。Unisysのミドルウェアのコア製品群は次の製品を包含する。

- ・ Open/DTP: Open GroupのDistributed Transaction Processing (分散トランザクション処理) 標準を実現

- ・ MQSeries
- ・ Java : Java Virtual Machine (JVM)
- ・ Web アプリケーション・サーバー環境
- ・ 低レベルの接続機能

その他のオペレーティング・システムの下で利用可能な技術もさらに存在する．一例として Windows NT および Windows 2000 の下で動作するマイクロソフト COM および関連技術が挙げられる．他の例としては BEA Systems 社の Tuxedo や WebLogic 製品があり，NT/Windows 2000 および UNIX の下で利用できる．これらの技術を OS 2200 および MCP/AS 環境で支援する一つの方法はネイティブ・モードで実行できるように移植することである．しかし，移植は複雑な作業であり，ソフトウェアのライセンス取得という理由からも不可能に近い．ライセンスという問題がたとえ解決しても移植には時間がかかるため，製品を使おうとしても製品化との間に時間的な遅延が発生するのは避けられない．代替案は OS 2200 および MCP/AS の下で利用可能な核となるミドルウェアに接続する統合技術を開発することである．

このアプローチにしたがうことにより，いくつかは本稿において前にも記述しているが，多くのアプリケーション環境統合を開発する結果となっている．必要なソフトウェアとミドルウェアを Unisys が選択，統合しさらにテストを実施したのちに製品をパッケージの形にして提供する．良く調整されテストされたミドルウェアのセットを持つことにより，ミドルウェアによって提供される様々な選択肢を通して，開発者を導くことを特別に目的としたツールを Unisys が製作できるようにしている．

7. 参考情報

ミドルウェアに関する文献は，カンファレンスの数と同様に非常に多い．ここではいくつかの Web ベースの情報を提供しよう．

Unisys のソフトウェア製品の情報は次のサイトから入手できる．

http://www.unisys.com/products/clearpath_servers/

加えて，ClearPath システムのミドルウェア戦略と製品に関する二つの技術文献が利用可能である．一つは OS 2200 システムに関するものであり，もう一方は MCP/AS システムに関するものである．これらの技術文献では利用可能なミドルウェア製品の背景を模索し，情報を提供することとともに，数多くのミドルウェアの使用例を提供している．

BEA Systems 社の製品データおよびその他の情報は次のサイトから得ることができる．

<http://www.bea.com>

マイクロソフト社のコンポーネント技術に関する情報は次のサイトから得られる．

<http://www.microsoft.com/com/>

Java に関する情報は次のサイトから得られる．

<http://java.sun.com>

CORBA に関する情報は

<http://www.omg.org>

IBM 社の MQSeries に関しては

<http://www.ibm.com/software/ts/mqseries>

から得ることができる。

最後に、ClearPath システムに興味を持たれた方には、Web サイト Unisys ClearPath eCommunity がミドルウェアを含む情報、Web 放送および様々な ClearPath システムのトピックスを提供している。この Web サイトを利用するには次のサイトをアクセスしてメンバー登録されたい。

<http://ecomunity.unisys.com>

* 1 Chris Britton 著 “IT Architecture and Middleware: Strategies for Building Large, Integrated Systems” Addison Wesley 社 2001 年発行，ページ 91.