

大競争時代を迎えて、企業の競争優位確保には情報システムが重要になっている。情報システムの効率的な開発や運用管理がポイントである。オープンシステムのもつ自由度は長所であると同時に短所ともなっており、効率的なシステム開発を阻害している。この自由度の制限と、WWW技術をベースにしたWebtopベースの軽量クライアント活用のような新たな情報技術活用が不可欠である。インターネット時代の時計の進みは、それ以前と比べると3倍速くなっていると言われ、システム開発スピードも3倍にする必要が叫ばれている。アプリケーションフレームワークによるソフトウェア部品の活用などにより、できるだけ作らないシステム開発も重要である。

Information systems are an important factor in ensuring that enterprises remain competitive in the global competition era. They must be able to be developed rapidly and maintained easily. Openness, the most important characteristic of open systems such as UNIX, often prevents people from developing such systems rapidly and efficiently. It is therefore necessary to restrict, to some extent, openness in system integration using open systems.

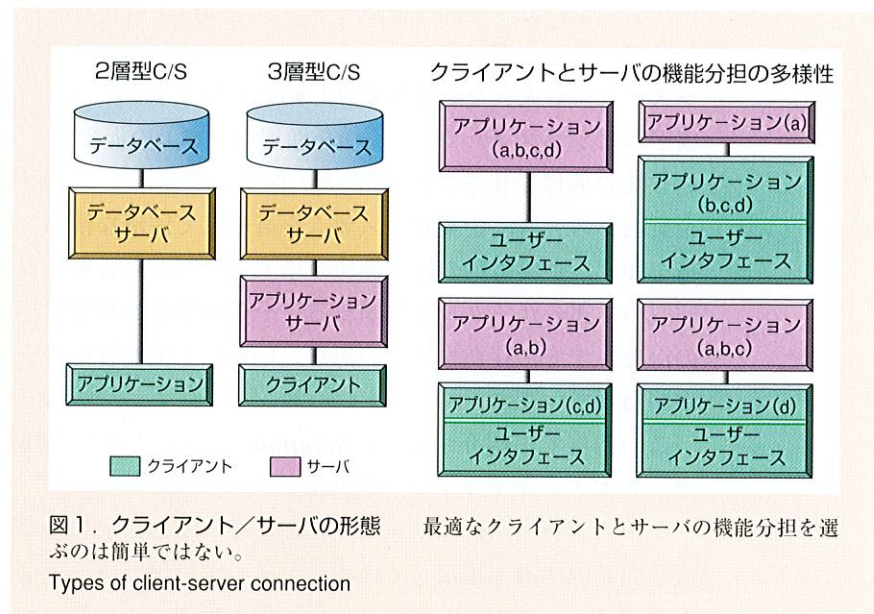
On the other hand, it is also necessary to apply new technologies such as Webtop-based thin-client to the integration. In the Internet era, times passes three times faster than previously. This means that systems must be developed three times faster as well. Development periods could be shortened through minimizing programming work by increasing software reusability with framework technology.

システムインテグレーションを巡る技術的問題

オープンシステムによる情報システムの発展の陰には、新たに発生した深刻な問題もある。

第一の問題は、オープンシステムによるシステムインテグレーションの特長とされる“組合せ型システム構築”の難しさである。全世界規模の企業間競争が激化している今日、情報システムを従来以上に迅速に構築することが求められている。世界に流通する優秀なコンポーネント(ミドルウェアなど)の組合せ型システム構築傾向が強まっている。

しかし、必要なコンポーネント選択や、それらの組合せ機能確認や性能検証は容易ではない。市場に流通するコンポーネントは、供給側の思惑で次々と変化していると言えなくもなく、顧客が必要とする基本機能は古今そう変わるものではない。新機能は、そうした基本機能の上に追



加されるべきものであるが、肝心の基本機能はおろそかになり目新しい機能だけが突出する傾向にある。コンポーネント組合せのシステム構築で、基本機能の思いがけない欠落に悩まされる。コンポーネント組合せの構造やインタフェース標準が乏しく、組合せに際して不具合が頻発し

て困り果てることも多い。第二の問題は、クライアント/サーバ(C/S)型システム構築の難しさである。90年代に入りパソコン(PC)やLANの能力が向上し、PCをクライアントとするC/Sシステムが企業システムに適用され始めた(図1)。その多くは2層型C/Sと呼ばれるも

Javaとは？

Java^(注1)は、1995年に発表されたオブジェクト指向プログラミング言語である。構文はC++に似ているが、C++のもつ問題を克服するために、次のようなくふうが見られる。

- (1) オブジェクトの寿命 C++, Javaともに、オブジェクト生成はプログラム責任で行う。一方、不要になったオブジェクト削除は、C++ではプログラム責任で行わなければならないが、Javaでは自動的に行われる。Javaではオブジェクトの寿命はプログラムが意識する必要がない。C++では不要なオブジェクト削除忘れにより、メモリが食いつぶされることがある。

- (2) ポインタ型 ポインタ型は問題発生の元凶である。ポインタ変数の値を誤り、メモリ破壊を招くことがある。ポインタが、プログラムの可読性を損なうこともよく知られている。Javaにはポインタ型は存在しない。

Javaは、ネットワークを意識した言語である。アプレット(小さいアプリケーションで、WWWページで実行される)実行に必要なクラスライブラリは、ネットワークから実行に先立ちダウンロードされるので、アプリケーションの事前インストールが不要である。ネットワーク経由でダウンロードされるプログラムの場合、どの計算機で実行されるのかをプログラム作成時に予測できないことが多い。したがって、Javaでは、プログラムを、バイトコー

ドとよばれる仮想的な計算機(Virtual Machine)の命令コードに変換・実行する方式を採用している。バイトコードでは性能が不十分な場合、JIT(Just-In-Time compiler)と呼ばれるコンパイラで、実行の直前に機械語にコンパイルする方式を採用している。

これらの性質から、Javaはクライアント側の記述言語として期待されていたが、最近ではオブジェクト指向プログラム言語としての長所、プログラム可搬性の高さなどから、サーバ側の記述言語としての期待が高まっている。サーバで要求されるクラスライブラリの標準化などが精力的に行われているほか、最近では、Javaプログラムを最適化された機械語に変換する、いわゆる普通の最適化コンパイラに注目が集まり始めている。

のであり、クライアントに搭載したプログラムがSQL^(注2)を発行し、データベース(DB)サーバに検索を依頼し、検索結果をクライアントで加工する形態である。グループウェアや情報検索サービスが中心であった。DBのリアルタイム更新に伴う企業基幹システムでは、中間にアプリケーションサーバを介在させて負荷調整をはかり、クライアントとの情報交信を効率化する3階層C/Sの実用が試みられた。クライアントとサーバが一体として目的の仕事を実行するのがC/S方式であり、クライアントとサーバのそれぞれにどのような機能を分担させるかが重要である。しかし、機能分担の多様性(図1)にもかかわらず、そのための有効なガイドが存在しておらず、C/S形態のこれ以上の普及を阻害している。また、現場部門導入のEUC(End User Computing)用ソフトウェアとIS(Information Systems)部門配布の基幹処理用ソフトウェアが混

在すると、“クライアント資源実状把握が困難”、“ソフトウェア更新作業が煩雑”など、システム運用管理の問題も深刻になった。

C SolutionTMの特長技術

このような問題の解決を旨とし、システムインテグレーション体系“C SolutionTM”を提供する(図2)。その特長技術は次のとおりである。

- (1) インターネット技術、分散オブジェクト技術、システムセキュリティ技術、ネットワークシステム管理技術を融合し、UNIX^(注3)とWindows[®](注4)の混在に適合する共通プラットフォームを提供。
- (2) アプリケーションプログラム開発のための高度なソフトウェア部品技術、“アプリケーションフレームワーク技術”を実用化。
- (3) それらを活用した迅速なシステム開発向け開発方法論と開発

支援ツールおよびプロジェクト管理技術を提供。

第一フェーズで、“Webtopコンピューティング”アーキテクチャに沿うプラットフォーム機能とその上に構築される応用層のためのソフトウェア部品、開発方法論を実現し、第二フェーズで、分散オブジェクト技術によるスケーラビリティ(拡張性)を付加する。

Webtopコンピューティングは、ブラウザを主体としたシンプルクライアントをもち、処理主体をアプリケーションサーバに移してDBサーバとの仲介を担わせる、3層型C/Sの整理されたアーキテクチャである。クライアントへの事前のアプリケーション配備は行わず、実行時に、必要なプログラムがクライアント環境にダウンロードされる。アプリケーション資産のサーバでの集中管理により、TCO(Total Cost of Operation : システム運用コスト)改善を図る。また、従来、応用層で

(注1) Javaは、Sun Microsystems社の商標。

(注2) SQL(Structured Query Language)

関係データベース(RDB)とよばれるデータベース内の情報を取り扱うための言語。

(注3) UNIXは、The Open Groupの米国およびその他の国における登録商標。

(注4) Windowsは、Microsoft社の商標。

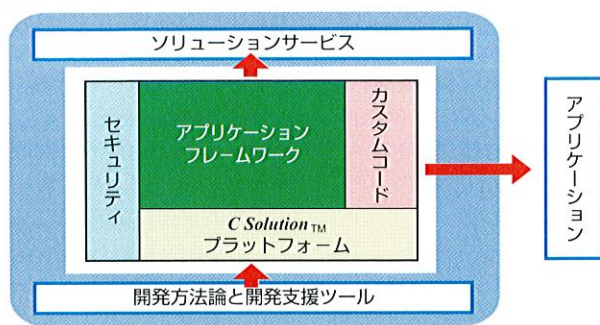


図2. C Solution™の構成 C Solution™には、ソフトウェアだけではなく、開発・プロジェクト管理方法論、ソリューションサービスなども含まれている。
Configuration of C Solution™

くふうを余儀なくされていた多数のクライアントとサーバの対話通信制御を基本機能として提供して、アプリケーション開発の容易化と性能バランス確保を実現している。

C Solution™では、オープン時代のシステム構築フレームワーク(標準枠組み)を確立した。プラットフォームが本来具備すべき基本機能を見定め、それを実現するコンポーネント群を選定、組合せ検証により、安定したプラットフォームに仕立て上げている(C Solution™プラットフォーム)。コンポーネント選定では、当社自製品に加え、全世界から優秀なコンポーネント製品を選んだ。また、印刷や画面の処理定義容易化も、地道であるが必須(す)なテーマとして実現機能の筆頭に挙げた。ブラウザの長所であり欠点でもある操作の自由度も、基幹処理のための制約機構を設けて整合性を確保している。また、大量データのいろいろな価値観からの分析活用には、新しい考えに基づくバッチ処理が不可欠であり、ジョブフロー制御機構として実装した。

独立に開発されたコンポーネント群の間に均一の強度のセキュリティを実現することにも注意を払った。

今後主流となると期待される個人認証機構とディレクトリサービス機能を中心に、セキュリティ機能を実現した。

アプリケーションシステムの迅速、かつ確実な開発のためには、安定したプラットフォームと、アプリケーションを構成する検証済みのソフトウェア部品が必要である。また、それらを活用し、開発・検証の積重ねで開発が確実に進展し、第三者にその進展が可視化される開発方法論とプロジェクト管理方法論が備わるべきである。C Solution™は、Javaを活用したソフトウェア部品、アプリケーションフレームワークやインクリメンタル開発を現実する方法論により、迅速な開発(RAD: Rapid Application Development)を支援している。

C Solution™が目ざすサービス

組合せ型システム構築能力をもち、顧客の事業環境と条件を認識し、投資効果が見えるシステムを、早く確実に実現するサービスの需要が増大している。当社は、C Solution™に基づくRAD技法を駆使するスキルをもつシステムエンジニアを育

成/資格認定して配備することにより、この需要にこたえるシステムインテグレーションサービス事業を展開する。

今後の技術動向

今後のシステムインテグレーションを支える技術は、分散オブジェクト技術とコンポーネント技術である。

分散オブジェクト技術は、プログラムを構成するオブジェクトの複数サーバへの分散を可能とし、スケーラビリティを実現する。分散オブジェクト技術では、CORBA^(注5)とDCOM^(注6)が二大注目株であるが、マルチプラットフォーム対応性から、C Solution™ではCORBAを最有力候補として対応する。

コンポーネント技術では、Javaで記述されたソフトウェア部品の流通を可能とするEnterprise JavaBeans (EJB)と呼ばれる技術に注目している。EJBはオープンな技術であり、Beansと呼ばれるソフト部品を作成するための標準と、Beansを実行させるためのContainerと呼ばれる実行環境の標準からなっている。

C Solution™では、すでにアプリケーションコンポーネントをサポートしているが、今後はアプリケーションフレームワークのEJB対応を行う。



調 重俊
SHIRABE Shigetoshi

情報通信・制御システム事業本部 S1技術開発センター 参事、C Solution™開発プロジェクトリーダー。コンピュータ基本ソフトウェアの開発に従事。情報処理学会会員。S1 Technology Center

(注5) CORBA (Common Object Request Broker Architecture)

CORBAは、分散オブジェクト技術の仕様を定めている。その仕様は、OMG(Object Management Group)と呼ばれる団体が決定され、その結果は一般に公開されている。

(注6) DCOM

Microsoft社の分散オブジェクトモデル。UNIXでは利用できない。