



## 「統合性とアクセスを Java で強化したプロセス制御 OI (オペレータ・インタフェース) の向上」

SL 社 CEO トム・ルビンスキ

SL 社 プロセス・コントロール・マネージャ プロフォール・バヤーニ 共著

トム・ルビンスキ  
CEO  
SL Corporation  
Corte Madera, California



プロフォール・バヤーニ  
Process Control  
Manager  
SL Corporation  
Corte Madera, California



典型的なプロセス情報システムでは、非常に多くの連続的に変化する変数を大きなネットワーク内にある複数のクライアントで見る必要があります。オペレータ・インタフェース (OI) のユーザは、グラフィックス画面と複雑な操作を行ない、最小限の遅延で、ほぼリアルタイムに動的な情報が見られることを期待しています。こうした諸条件のもとで競合力のある制御システムは、コンパクトでスケラブルなアーキテクチャでなくてはなりません。

画面は、特定の表示に必要なデータのみを転送して、負荷を最小限にして更新するべきです。そしてユーザが新しい表示を要求したときには、新しい画面の待ち時間は長くても 2 ~ 3 秒であるべきです。この課題は、バンド幅が狭いまたは広くないそこそこのインターネット接続を経由したシステムで利用できるようにすることが目的の場合、ますますむずかしくなります。データ表示画面の更新に使える Web ベースの技術には、サーバ・ページ、ActiveX、Java の 3 つの選択肢があります。

### サーバ・ページと ActiveX によるソリューション

一般的なソリューションの 1 つは、ある情報の特定のビューがほしいという、インターネット・クライアントからのリクエストに対し、サーバに置いたプロセスに応えさせるというものです。サーバ・プロセスが HTML ページと 1 つ以上のイメージを作成し、これをクライアントに送り返します。このアプローチは、複数のユーザがいる場合、サーバへの要求が非常に過酷になります。また、それはバンド幅も強く要求され、画面を更新し続けるにはイメージ全体を定期的に転送する必要があります。

改善ソリューションとしては、サーバが要求された特定データのみを Web 間で転送するだけにし、Thin クライアント・アプリケーションを利用してクライアント側で表示を更新することです。プロセス業界では Microsoft ActiveX ソリューションを、この目的で採用しています。これは利用可能なソリューションとして十分にその目的に対応できましたが、Microsoft Windows の環境に限られていました。また、インストールの際には細かい注意を要し、結果としてエラーとなる可能性が、常にありました。

### Java によるソリューション

多くの面で、Java によるソリューションは ActiveX 技術に非常によく似ています。クライアント・マシン上の Java プログラムがグラフィック

ス画面を表示しますが、ネットワーク間ではデータ更新だけを転送すればよいのです。

当初は、プロセス・アプリケーションで Java を使うことを懸念した開発者が多くいましたが、それには確かな理由がありました。Netscape と IE のバージョン 3.x ブラウザは制約があり、バグの多い java1.0 のみをサポートしていました。こうした初期バージョンを使うには、深刻な制約がありました。

しかしながら、今やバージョン 4 以上のブラウザでは、Java 1.1 が大変よくサポートされています。そしてサン・マイクロシステムズ社は、JDK のさらにより新しいバージョン (1.3 以上) をリリースし続けており、実世界のアプリケーションは、この環境で提供することができます。その結果、Java による Thin クライアントが、大量の動的プロセス・データへのアクセスと表示の手段として、ActiveX ソリューションに対するもう一つの選択肢になってきています。Java には、Java で書いたアプリケーションが、特に重要な Windows, UNIX, Mac を含め、ほとんどすべてのプラットフォームで走るといふ、特異な利点があります。

さらに、Java 言語はオブジェクト指向プログラミング・スタイルをサポートし、多くのプログラマにとっては C++ よりずっと使うのが簡単です。シンタックスは追いややすく、C++ のカジュアル・ユーザにとっては複雑すぎるインタフェース構造も一種の多重継承を実現するうまい方法を提供しています。オブジェクト指向アーキテクチャをサポートする Java 言語の適性が、Java をとても価値あるものにしていました。オペレータ・インタフェースの性質は非常にオブジェクト指向であり、大規模 OI アプリケーションの開発、保守、拡張には、よいツールと言語のサポートが必要です。

### ブラウザ・ベースの Java アプレット

Java 仮想マシンを持つソフトウェア・プラットフォームならどれでも、Java アプリケーションを実行できます。ローカルなイントラネットのサポートに対しては、Java アプリケーションはお客様のシステムで走っている他のプログラムと非常に似通っています。

しかしながら、インターネット・アクセスには、Netscape と IE の現行のブラウザ群で提供している Java サポートは大変良くなっています。実際、ブラウザのフレームワーク内で Java アプレットとしてアプリケーションを採用することには、重大な利点があります。どのマシンのどこにあるブラウザでも特定の Web アドレスにある HTML ペ

本文は、米国 Control Solutions 誌 February 2001 号に掲載された記事の抜き刷りを和訳したものです。



## コントロール・ソフトウェア・フォーラム

ージにアクセスでき、ブラウザ組み込みの Java VM を使って自動的に Java アプレットを実行できるからです。ユーザのシステムにダウンロードするものは何もありません。アプレットは Swing コントロールや Java 2D グラフィックスなどの、より新しい Java 機能を利用できるように設計することもできます。しかし、ユーザはその特定のブラウザ用の特別な Java プラグインをダウンロードする必要があるかもしれません。

### 性能の考慮

Java アプリケーションはどれも、同等の C++ ネイティブ・アプリケーションより遅くなりますが、それは、単に Java の実装のインタープリタの性質によるものです。性能の問題が、大規模システムへの Java 採用を遅らせていましたが、Java が向上し、ハードウェア性能が加速するにつれて、これはめまぐるしく変化しています。

Java は今にコンパイルされるようになります。Java が C++ のようにコンパイル・コードを生成するように処理できない理由は何もありません。今は便宜上、インタープリタです -- マシン毎にコンパイラを作る必要はありません。しかしながら、Java で書いたアプリケーションが増えてくるにつれ、最適化の要求が増大し、それはコンパイルによってのみ達成できるものです。

### Java で構築されたサードパーティのサポートの選択

プロセス・アプリケーション用の OI 構築時には、解決すべき問題がたくさんあります。その用途に特定して設計され、最新の Web 機能や Java ベースの機能を盛り込んだ最新の技術を使ったサードパーティ製品を利用することは意味があることです。たとえば、良く設計されたサードパーティ開発システムの多くでは、次のような基本機能を提供しています：

1. グラフィックス画面の表示と関連するデータ変数を管理するランタイム・システム
2. グラフィックスおよびそれに関連する動的振る舞いの構築ができるグラフィックス・エディタ
3. SVG、XML などの Web に共通するファイル形式に対する互換性に加え、最適化に使える独自のファイル形式

他の機能の多くはさほど目立ってはいませんが、以下は現行の Java 実行環境では極めて重要です：

4. Java コード生成などの特別な技法を使って達成するスケラビリティ。システムのパワーが増強するにつれて、アプリケーションへの要求も増大します。システムでそれができるなら、お客様のアプリケーションにさらに詳細仕様を乗せようとするのではないのでしょうか。たとえば、お客様の競合会社がお客様と競ってさらなる機能を実装するでしょう。成功するには、幅広い最適ソリューションを提供しなくてはなりません。一寸の余地もないのです。
5. ユーザ・インタラクションを扱うルーチンのあるステート指向アーキテクチャ。これは多くの場合、有限ステート

ト・マシン・アーキテクチャの使用と、有効なステート階層をサポートする API が関係してきます。

### 未来に向けての開発

Java の成熟と共に、開発ツールとライブラリが簡単に利用できるようになりつつあり、Java の Thin クライアントが、プロセス・グラフィックス OI への採用において、真剣に考慮すべきオプションになりつつあります。お客様の製品やサービスの構築を支援するのにどのようなツールや製品を使うかを決められる際には、あるいはお客様が社内でシステムを構築されることを考えておられるなら、大規模な制御システムの OI 構築が複雑な仕事であることを認識してください。それは単に OI のグラフィックス部分だけではありません。統合、性能、実行に関する問題がたくさんあります。インターネットは、サプライヤからカスタマへの情報の流れ方を劇的に変えてしまいました。サプライ・チェーンはグローバルにかつますます複雑になりつつあります。原材料の流れはグローバル化したサプライ・チェーンで加速され、より効率的でカスタム化可能なプロセス・コントロールが求められています。

これにはリアルタイムの情報の流れ、全社規模の統合と視覚化が必要になります。MES (製造実行システム) はプラントの現場と ERP/SCM (Enterprise Resource Planning/Supply Chain Management) を結びものとなり、プラントのグラフィックスとトレンド解析を混合したものを提供し、生産ユニットの稼働状況と顧客注文の完了ステータスを報告するものとなりました。

Java 環境では、将来を念頭に置いて開発することができます。システムをアプレットまたはアプリケーションとしても機能するように、あるいは AWT (Abstract Window toolkit) または Swing コントロールでも動くように開発することで、複数の採用シナリオを提供することができます。また、オブジェクト指向およびステート指向のアプローチをアプリケーション・アーキテクチャに利用し、C++ でも Java でも等しく実装できるモジュールを、クラスおよびステート・クラスとして設計することにより、アプリケーションの背後にあるロジックへの投資を守ることもできます。そうしてのみ、お客様の OI に、Java とそれ以降の未来が約束されます。

### 著者について

トム・ルピンスキは 1984 年に SL 社を設立し、現在同社の社長兼 CEO であり、SL 社のグラフィカル・モデリング・ソリューション (SL-GMS) ソフトウェアの開発に携わってきました。ルピンスキはカリフォルニア技術大学 (バサデナ) で学びました。

プロフォル・バヤーニは SL 社のプロセス・コントロール・マネージャであり、SL-GMS 技術の方向付けを担当しています。バヤーニはプロセス・オートメーションで 15 年の経験があり、1984 年にバンガロア大学を卒業し、工学学士号を得ました。

無断転写転載禁止