

サプライチェーンからみたプラント情報化の考え方

Java Thin クライアントによって プラントフロアを超えるHMI —MES, ERP/SCMレベルに広がるプロセス 情報のビジュアル化—

SL ジャパン 石田 貴久

1. はじめに

今日のプロセス製造オペレーションは、先例のないチャレンジと向かい合っている。インターネットが、サプライヤから顧客への情報の流れを大きく変えようとしているからだ。サプライチェーンは、よりグローバルかつ複雑になり、原材料の流れはグローバル化したサプライチェーンによって加速化され、効率的でカスタマイズ可能なプロセス制御が必要とされている。

結果、全社規模でリアルタイムなプロセスデータの流れを見て統合することが要求されている。製造実行システム (MES) がプラントフロアと ERP/SCM のつなぎ役となり、生産ユニットの稼働状況と、計画および顧客注文に対する充足状況を示す、プラント情報のグラフィックスとトレンド解析を提供しな

くてはならない。

たとえば、共通のプラントタンクのビューも、プラントフロアのオペレータ、MES レベルのプロセスエンジニアや管理者、そして ERP/SCM レベルの生産計画エンジニアそれぞれが必要とするデータに応じて表示できるグラフィックス画面が要求される (図 1)。

2. プラントフロアを超える HMI の課題

プラントフロアを超えて、プロセス情報を MES, ERP/SCM レベルで共有するための HMI で要求されるのは、プラットフォームに依存しないインターネット/イントラネット環境における Web クライアントである。

Windows に限定されず、プラットフォームに依存しない Web クライアントとして、Java 技術とその開発環境の向上により、またハードウェア性能が加速するにつれて、Java がプロセス制御における HMI として選択されるようになってきた。また、Web クライアント間におけるデータ交換には、インターネットとの親和性が高く、異なる機器装置間におけるデータ通信技術である XML が標準的に採用されつつある。

しかしながら、プロセス情報システムのグラフィックス画面では、連続的に変化する多数の変数をほぼリアルタイムに動的なグラフィックス画面で見ることが要求

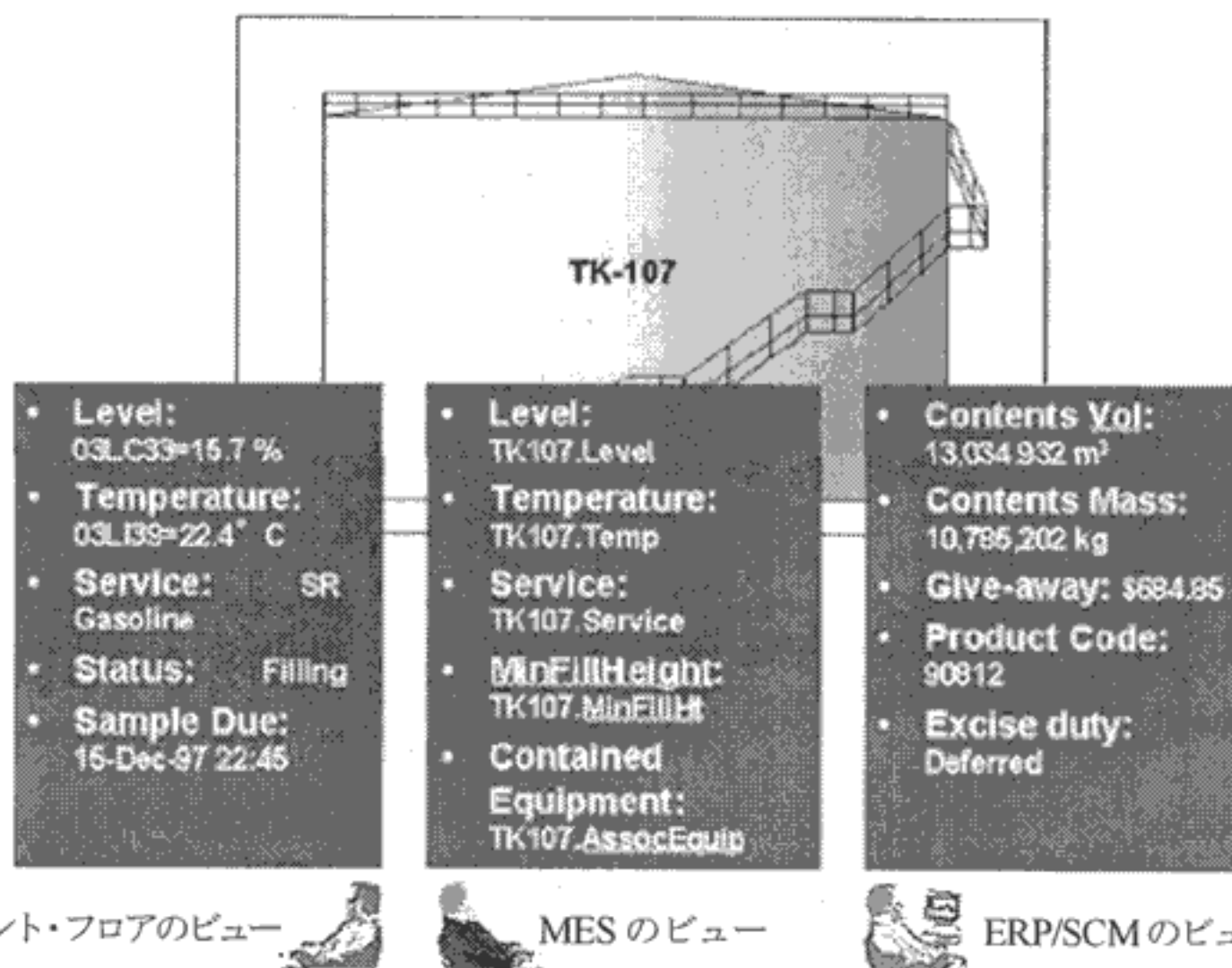


図1 タンクの異なるビュー:必要となるデータに応じたグラフィックス画面表示が可能

される。よって画面は表示が求められたデータのみを転送し、負荷を最小限にして更新時間を長くても2~3秒に留めなくてはならない。この課題は、インターネット環境における情報の共有においてはさらに深刻な課題となり、サーバが要求された特定データのみを転送するThinクライアント技術が不可欠条件となる。

JavaクライアントによるHMIには、こうしたグラフィックス性能の問題をはじめ、カスタマイズ性、既存のC++/ActiveXなどとの部品共有または統合性、新技術採用に対するリスク懸念、Java開発者の不足などの課題がある。

また、Javaコード生成などの特殊な技法を使ってスケーラビリティも達成しなくてはならない。システムのパワーが増強するについて、アプリケーションへの要求も増大し、特に競合上においてはシステムの継続的な機能拡張が避けられない。

そして、プラットフォームに依存しないスケーラブルな「グラフィックス部品」は、プロセス情報を共有するという目的から、従来のC++/ActiveX監視制御システムを再利用できることが望まれる。つまりプラントフロア、MES、ERP/SCMそれぞれのレベルの用途に応じてグラフィックス部品のコードを自動的に生成でき、インターネット/イントラネットの環境においてもリアルタイム性を維持できるパフォーマンスが必須なのである。

3. 「SL-GMS J/Developer」ソリューション

以上の課題に応え、プラントフロアを超える新世代のHMIを容易に構築できる開発環境を提供し、インターネット/イントラネット環境におけるグラフィックス・データ表示の共有を現実的なものにしたのが、「SL-GMS J/Developer」ソリューションである。

SL-GMS J/Developerでは、既存のC++/ActiveXシステムと共通して使用できるSL-GMSDrawエディタでカスタマイズ可能なグラフィックス部品を作成、Javaプログラミングをしないで迅速に画面をプロトタイプングすることができる。さらに、SL-GMS J/DeveloperのJavaコードジェネレータがコンパクトで高速なJavaアプレットを生成、それぞれの現場のニーズに合わせたスケーラビリティを実現し、他に類を見ないJava Thinクライアント技術によって、インターネット/インターネット環境におけるパフォーマンスを発揮する。

4. 「SL-GMS」Thinクライアント技術

SL-GMS J/Developerでは、最もコンパクトで効率的な高速Javaコードを生成するJava Thinクライアント技術を誇っている。ほとんどのグラフィック・ソリューションでは、さまざまなグラフィックス画面を表現するための汎用クラスをダウンロードするか、またはグラフィックス・インタプリタ・エンジン全体とグラフィック・メタファイルをダウンロードしてインタプリタを実行。どちらのケースも、カレントに表示しているグラフィックス画面では必要のないロジックや無関係のクラスを要する「重い」クライアントになってしまうのが現実である。

そこでSL-GMS J/Developerでは、特定のグラフィック・オブジェクトならびに画面に絞った動的な描画と更新を処理できる最適化されたコンパクトなJavaクラスを生成する。これにより、最もThinなクライアントの作成による性能無比なダウンロード速度を実現している(図2)。

図2は、SL-GMS Javaコードジェネレータが生成する、最もコンパクトで高速なJavaアプレットの例を示している。図には「補助加熱器アプレット Only 25K」、「核反応炉 Only 22K」、「カード・ラック Only 19K」の3つのアプレットのスクリーンショットが示されている。補助加熱器アプレットは複雑なプロセスフローを示し、核反応炉は円形のパネルとデータ表示を含み、カード・ラックは縦長のデータテーブルを示している。

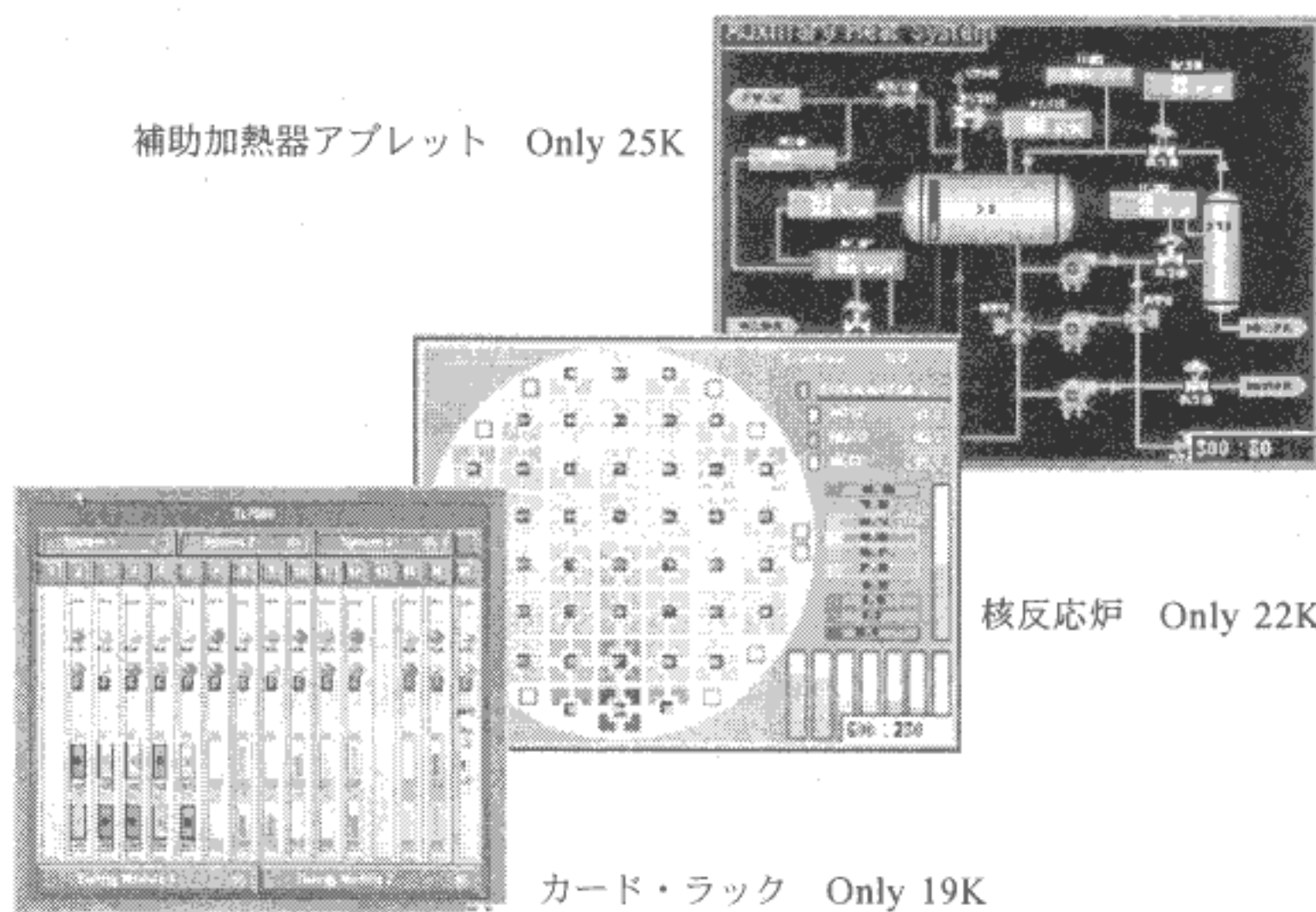


図2 「SL-GMS Javaコードジェネレータ」が生成する、最もコンパクトで高速なJavaアプレット

5. カスタマイズ可能なグラフィックス, 迅速なプロトタイピング

Java グラフィックス・ライブラリを拡張するためにコードを書く代わりに, SL-GMS J/Developer ではSL-GMSDraw エディタを使ったグラフィックス表現と動的振る舞いの定義ができる完全な柔軟性を開発者に提供している。よって, 開発者はダイナミック・グラフィックスを実装するための詳細の Java コーディングから解放される。

幅広い動的振る舞いを持つSL-GMSDrawでは, リアルタイムデータの変化に応じてどのようにグラフィックス表現が更新されるかを定義することができる。そしてエディタ内で動的振る舞いをプレビューできるため, Java コードを1語も書くことなく, ダイナミック・グラフィックス画面のプロトタイプを迅速に作成し, テストすることが可能である。これにより, グラフィカル・オブジェクトと画面を迅速にプロトタイピングし, アプリケーション・プログラミング作業を開始する前に, さまざまなユーザインタフェースの選択肢を試すことができる。

開発者は, ポンプやバルブを表現するアイコンのような動的オブジェクトや, カードとラックのような機器装置の状態を表示する詳細な画面を SL-GMSDraw で作成する。SL-GMSDraw は, ダイナミック・オブジェクトを階層的に作成できるため, すでに定義されたオブジェクトで合成したアイコンをつくるのが可能だ。そして, プロセス概観, ダイナミック・オブジェクト, 定義されたアイコンのインスタンスで構成される画面を作成し, リアルタイムプロセスを自在に表現することができる。

SL-GMS の強力なオブジェクト指向アーキテクチャは, ダイナミック・グラフィックス・オブジェクトの再利用によって市場投入時間の短縮に貢献してきた。さらに, DCS/PLC 分野のお客様は, それぞれのプロセスプラントのルック&フィールにおける完全な柔軟性をエンドユーザに提供できるカスタマイズ・エディタ・ソリューションを選択されてきた。

6. 高速な Java ランタイム・エンジン

SL-GMS J/Developer のシンプルでかつパワフル, そして拡張可能なデータソースへのインタフェースは, インターネット環境における Java アプリケー

ションまたはJavaアプレットを膨大なデータで動的に更新できる柔軟性を開発者に提供している。

データソース・クラスは, ダイナミック・グラフィックスを駆動する変数が含まれる変数テーブルを提供している。入って来るデータを管理し, どのグラフィックスがデータに依存するかを判断し, 個々のグラフィカル・オブジェクトに対してメソッドを呼び出してその外観を更新する代わりに, ユーザは特定のデータ変数が変更されたことだけを示すだけで, そのデータに依存するグラフィカル・オブジェクトの動きは自動的に更新される。また, XML をデータソースとして使えるクラスも提供している。

SL-GMS Java ランタイム・ライブラリは, 実行時における動的に表示するグラフィックスのインスタンスをサポートしており, データまたはユーザインタラクションによって画面領域にオブジェクトを表示させることが可能である。そのほか, ズーム/パン, レイヤリング, オブジェクトの選択, プルダウン・メニューなどのオブジェクト操作など, 豊富な機能をサポートしている。

7. Java, C++, ActiveXで統一されたソリューション

Java プラットフォームが多くの分野で幅広く採用され始めている一方で, ミッションクリティカルなシステムでは, パフォーマンスや機能面, 従来システムサポートなどの理由で, 多種のプラットフォームが使用されているのが現状である。

SL-GMS 製品では, UNIX/Linux/Motif, Windows/MFC, ActiveX 環境などの幅広いプラットフォームをサポートし, 共通のSL-GMSDrawエディタで作成したグラフィックスモデルを活用できる。SL-GMS のダイナミック・グラフィックスは, これらのプラットフォームとグラフィック・フレームワークにおいて優れた移植性と再利用性を実現, 共通のルック&フィールを推進し, これらインタフェースへの開発投資を有効なものとする。

SL-GMS の統一されたソリューションにより, C++/ActiveX で構築する監視制御システムのオペレーションインタフェースを, Java Thin クライアントによるプラントフロアを超えたHMIへと拡張していくことが可能になる(図3)。

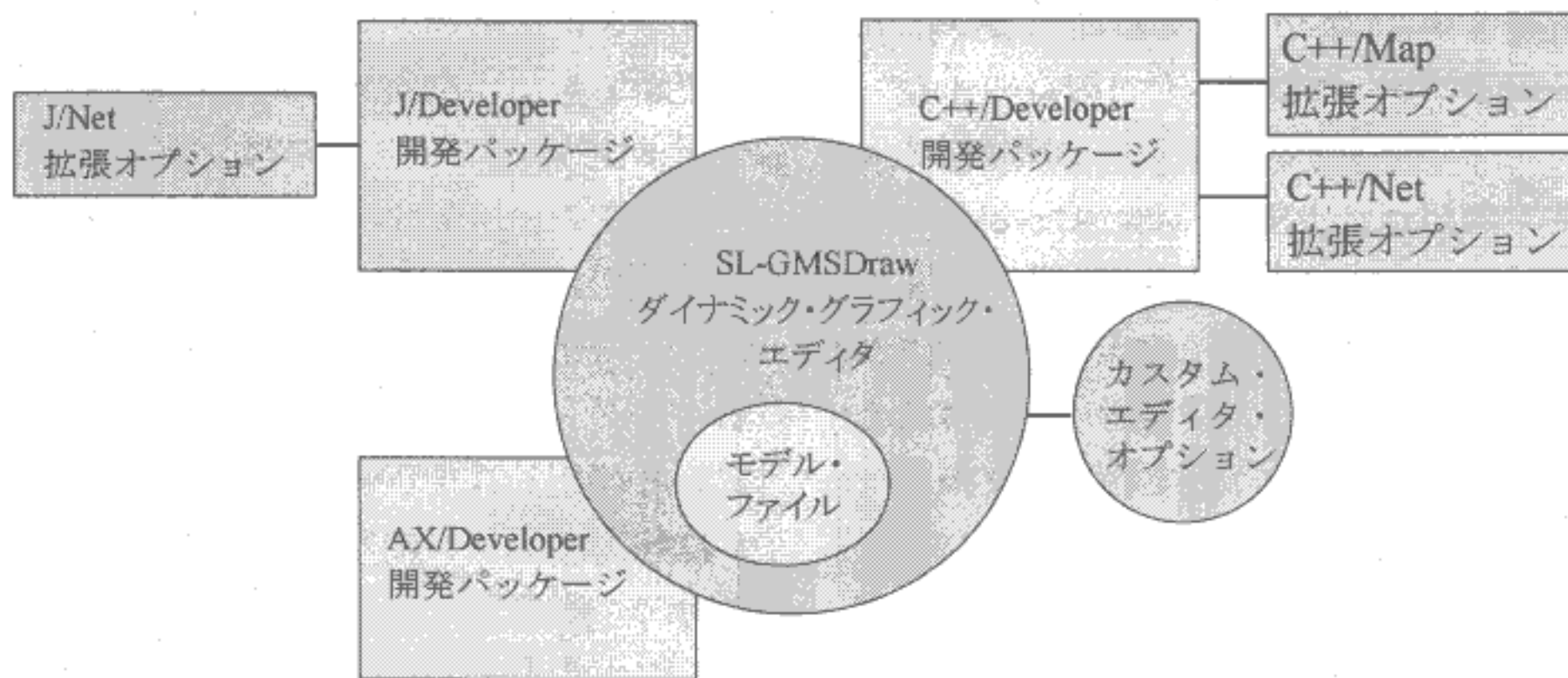


図3 「SL-GMS」で統一されたグラフィックスソリューション

8. 今後の展望：無線へ広がるJava ThinクライアントHMI

Java Thin クライアントによってプラントフロアを超え、MES、ERP/SCM レベルに広がるプロセス情報の共有は、今後さらに無線へと広がっていく。最近、高解像度のハンドヘルド端末が市場に出て話題を呼んだ。Personal Java をサポートしている SL-GMS J/Developer では、そのコンパクトな Thin クライアント技術により、SL-GMS のダイナミック・プ

ロセス制御画面を表示し、現実的なパフォーマンスを発揮できる。

注) 本文に記載されている製品名は、それぞれ各社の商標または登録商標である。

イシダ・タカヒサ
 (株)SL ジャパン 営業部
 〒107-0062・東京都港区南青山3-1-30
 電話(03)3423-6051
 E-mail : info@sl-j.co.jp