

# オープン化システムにおけるデータアクセスミドルウェアの開発

山武産業システム株式会社

藤井 稔久  
Toshihisa Fujii

## キーワード

Windows - NT, OLE, DCOM, OPC, オブジェクト指向, セキュリティ

過去、オープン化と言われながらも製品ごとに通信インタフェースがそれぞれ違い、同様なアプリケーションを開発する場合にでも、個別のインタフェースを用意し、アプリケーション開発というよりも通信インタフェースの開発に費やすことが多々あった。弊社では、OPCでの規定作成に先駆けて、オープン化システム用の共通なインタフェース、ミドルウェアを用意し、DCSとの通信部分をDCSドライバとして再利用・共通化するとともに、製品系列を超えて、アプリケーション部分の再利用化・共通化を図ることができるようになった。

同時に、セキュリティなどを標準提供する機能についても共通なものを用意することができるようになった。

### Development of data access middleware in open system

In many instances in the past involving an "open system," communication interfaces were different for each product. Further development work was necessary for communication interfaces rather than applications as similar applications were developed by preparing individual interfaces. Prior to the establishment of rules by OPC, a common interface and middleware for an open system has been prepared, and it is now possible to reuse/use-in-common the communications portion of the DCS with the DCS drive. This configuration enables the reuse/common use of the application portion regardless of the product family.

At the same time, it is possible to prepare common functions which are provided as standard functions, such as security.

## 1. DCS オープン化傾向における問題点

最近のプラント制御システムのオープン化傾向にあって、弊社においてもWindows - NTなどをプラットフォームに持つシステムを提供するようになった。これらのシステムが従来の制御系LANの閉じた世界から、制御系情報LANおよび情報系LANに開放されることによって、それらのLAN上のPCにおいて制御系LANの情報を取り扱うアプリケーションの開発などの機会も増えてきた。しかし、各システム毎に提供する通信インタフェースが異なるために、アプリケーション開発の度にデータアクセス部の開発というソリューション以外の部分に多大な労力を費やしている。

さらに加えて、情報系LANの環境で用意されている様々なアプリケーションに容易に制御系LANのデータを渡せないため、本来ならばサードパーティ提供のツールで解決できる問題に対しても、そのインタフェース部の開発、も

しくはサードパーティ製品で実施できるはずのソリューションの開発に労力を費やすことになる。

## 2. データアクセスミドルウェアによるアプローチ

上記に挙げたような問題に対しては、ミドルウェアという考え方が最適な解決を提供すると考えられる。すなわち、サービスを提供する側と使用する側の間に位置し、クライアントアプリケーションの立場からは統一されたインタフェースしか存在せず、各サービス提供部分が固有に持つ属性の違いに関しては、その差異を隠蔽してしまうソフトウェアコンポーネントの提供である。本誌で紹介しているデータアクセスの場合には、各システムに共通な通信インタフェースを使用できるようなソフトウェアコンポーネントを提供することが望まれている。共通な通信インタフェースがあれば、次頁のような効果が期待できる。

#### アプリケーション開発コストの削減

従来のアプリケーション開発環境では、各アプリケーションが独自に対象データソースに対するデータアクセス機能を開発している。共通のインタフェースを提供することによって、その部分の作業から開放されるとともに、新規アプリケーションの開発時にも蓄積されたノウハウを用いてデータアクセスを行うアプリケーションが開発できるので開発コストの削減が期待できる。

#### サーバー開発コストの削減

データサーバー開発の側面から見ても、共通インタフェースの提供によって、新規データソースに対応させる場合にもインタフェース部分の開発にかかる工数が削減できる。

#### サード・パーティ・ツールとの統合

顧客からの要望が強いものの一つに Excel などのサード・パーティ・ツールからプロセス・データを読み込んで表示や加工を行いたいというものがある。提供する共通インタフェースがサード・パーティ・ツールと統合できるものであるならば、データアクセス部以外にもデータ加工部分に費やす工数の削減が期待でき、ソリューション開発に注力できる。

これらの背景から、弊社にて取り扱うシステム製品の通信インタフェースを共通化するためのミドルウェアを定義、開発した。

本ミドルウェアで提供するデータアクセスサーバーは、基幹技術に Microsoft 社の OLE ( Object Linking and Embedding ) を採用して、COM ( Component Object Model ) と呼ばれるバイナリレベルのオブジェクト指向モデルによるサービスの提供をクライアント / サーバー構成にて実現している。OLE を採用したことにより、DCOM ( Distributed COM ) を用いた、ネットワークを介したクライアント / サーバー構成も実現し、OLE オートメーションのインタフェースを提供したことで、Visual Basic などのスクリプト言語からのサービス使用も可能にしてサード・パーティ・ツールとの統合も実現した。

さらに、このデータアクセスサーバーでは、基本機能提供部分はそのままに、各システムに特化するデータアクセス部分を独立した DLL ( Dynamic Linking Library ) モジュールで提供して入れ替えることによって、どのようなシステムに対してもセキュリティなどを含んだ共通機能を提供するという構造にした。こうした構造にしたことにより、実績として社内システムのみならず、競合他社の DCS にアクセスするデータアクセスサーバーも提供し、上位アプリケーション開発の工数削減に貢献した。

本ミドルウェアで提供する基本的な機能は主に「データリード」、「データライト」、「サーバー情報取得」の3点で

あり、データアクセスについては、配列一括によるアクセス、間接参照も可能である。また、列挙型、エンティティ型、パラメータ型などのように外部名と内部IDの2種類の表現方法を有するデータタイプをサポートするシステムの場合には、それらのデータをどちらの表現形式でアクセスするのも指定可能である。ただし、後に示す構造からもわかるように、本ミドルウェアで提供するデータアクセスサーバーはシステム毎に特化した機器依存部分を独立した DLL で提供するので、その特化部分での実装によってはこれらの機能が必ずしも使用可能であるとは限らない。それらのサーバー毎における機能的な差異および名前付け規約に関する情報をランタイム時に入手する手段も提供する。

### 3. OPC との対応

弊社では、Microsoft 社の技術である OLE を利用した産業システムの標準化仕様である OPC ( OLE for Process control ) のインタフェースが規定される以前から本データアクセスサーバーの開発を行い、その資産を蓄えていった。

一方、本ミドルウェアの目的は OPC の目的と同じく、異なるベンダの機器で実行される異なるアプリケーション間のインタフェースを標準化してデータ交換を容易にすることであり、それによってサーバーおよびクライアント側のインタフェースソフトの開発にかかる工数を低減させようというものである。こうした背景から、各ベンダからは各々のシステムに対応した OPC 準拠のサーバーや、OPC 準拠のクライアントアプリケーションが提供されている。

弊社でもこうした動きに対応し、さらなるオープン化への実現を目指している。そこで過去の資産も生かした上で OPC の利点も取り込むために、本データアクセスサーバーの上位に OPC プロトコルとの変換レイヤを被せることで、OPC クライアントアプリケーションからのデータアクセスも可能にし、高いオープン性を実現している。

本ミドルウェアで提供するインタフェースと OPC インタフェースの実現方法について図 1 に示す。図にも示すように、OPC インタフェースを実現するサーバーは、本ミドルウェアで提供するデータアクセスサーバーとは独立しており、その観点からは、OPC インタフェースを実現するサーバーは本ミドルウェアで提供するデータアクセスサーバーのクライアントであると言える。今回提供した OPC インタフェースは OPC 2.0 に準拠したカスタムインタフェースで、オプション機能は提供していない。また OPC のオートメーションインタフェースは実装されていないが、OPC 評議会などで提供しているオートメーションサーバーを利用すれば、オートメーションインタフェースも使用可能である。

このように、現在では、本ミドルウェアは OPC 仕様のインタフェースも提供できるようになっている。しかし、そ

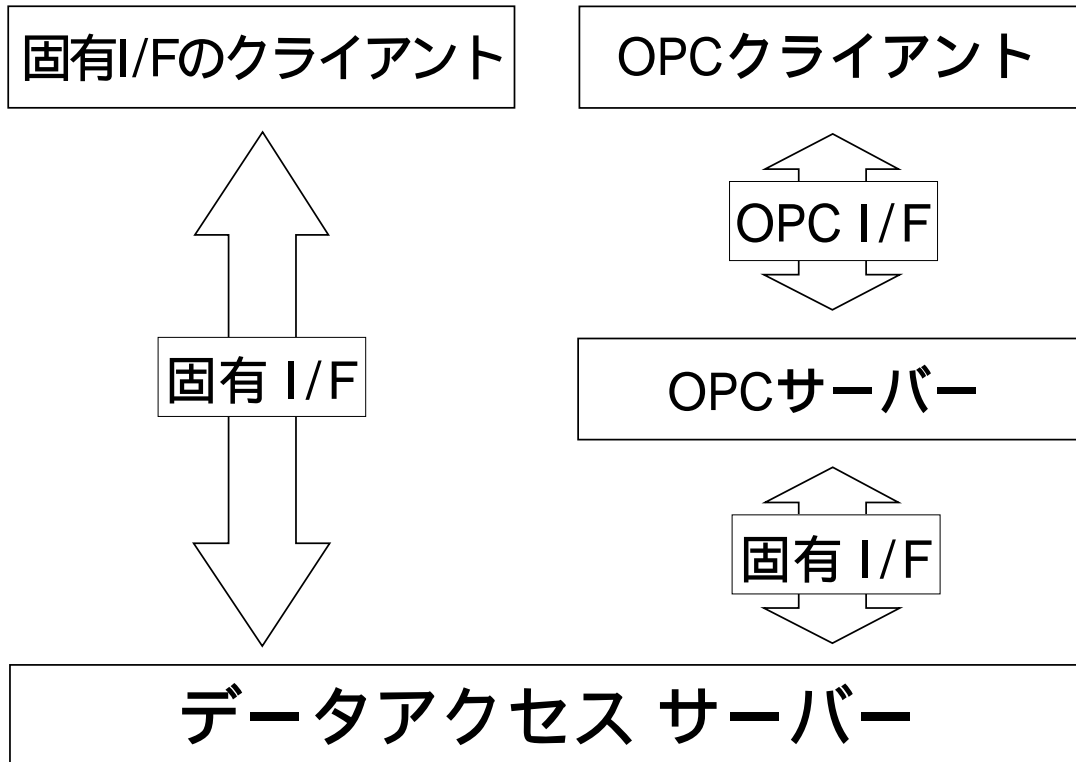


図.1 データアクセスミドルウェアの利用形態

れによって本ミドルウェアの提供するインタフェースが消えるとも考えていない。その理由は、本ミドルウェアとOPCの考えるデータサーバーの構造の違いに依っている。OPC仕様のデータサーバーでは内部にデータ・キャッシュを持ち、そのキャッシュ内容が一定の更新周期で更新される。通常はクライアントからのリード要求に対しては、このキャッシュ内のデータが返る。すなわちキャッシュがRTDB ( Realtime Database ) の役目を成している。よってリードアクセスのスループットは良いが、たまにしかアクセスしないデータも一定周期で更新されるために、余分な負荷がシステム内にかかる場合もある。本ミドルウェア仕様のデータサーバーにもデータ・キャッシュが存在するが、その用途はOPC仕様とは異なり、あるポイントへのアクセスが同時期に複数の要求があった場合に実デバイスへのアクセス回数を低減させる目的に使用される。この用途では、通常のスループットはOPC仕様と比較して早くないが、不必要なアクセスを無くすることができる。これらの違いには其々の長所と短所が存在し、システムまたは上位アプリケーションの特性から使い分けられるものと考えられる。本ミドルウェア仕様でのキャッシュの使い方を図. 2 に、OPC仕様でのキャッシュの使い方を図. 3 に示す。

もう一つ、ここで挙げる両者の仕様の違いとして、本ミドルウェアで提供するインタフェースには接続先システムの属性を返すものが存在する。ここでの属性とは、本ミドルウェアが提供するアクセス手段として配列一括アクセス

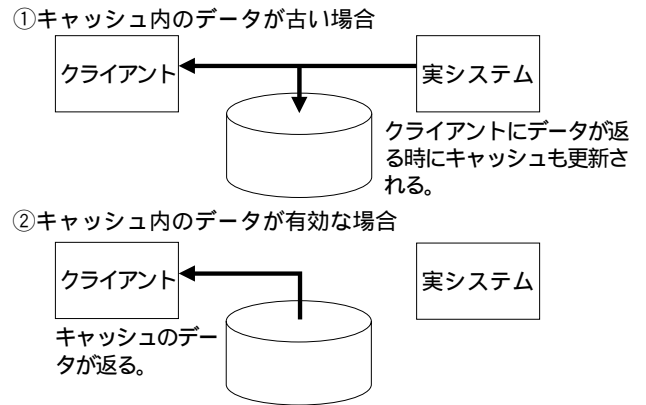


図.2 固定I/Fでのキャッシュ利用方法

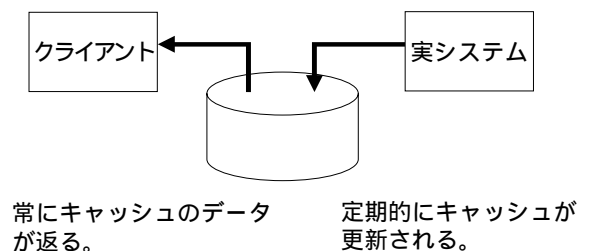


図.3 OPC I/Fでのキャッシュ利用方法

や間接参照アクセスをシステムの機器依存部分がサポートしているかといった情報の他に、システム毎の名前付け規約の差異を吸収できるようにするための情報を提供する。たとえばシステム毎のタグなどの外部名に使われるポイントとパラメータを区別する識別子の文字列や、配列を表現する場合のインデックスを指定する文字列がこれにあたる。これらをデータサーバーから動的に取得することにより、何の設定変更も行わずに各種システムと接続可能な単一のアプリケーションの開発も可能になる。OPC評議会では、名前付け規約に対する問題の対処は現在も検討中であり、独自の仕様を定義して、ネームスペース変換部分を各システム毎に提供させる可能性もある。

#### 4. セキュリティ

本ミドルウェアで標準的に提供するセキュリティ機能について述べる。

セキュリティと言っても様々な観点があり、制御系LANや情報系LANを扱うようなネットワークに関しても、不正侵入検知、監査、認証・識別、情報漏洩、情報改ざんなどがあり、それらについてプログラムにより対処する問題や、ソーシャルセキュリティのような運用により対処する問題がある。

セキュリティという議論で取り上げられるこれらの様々な観点の内、本ミドルウェアでは現段階としてログオンアカウントを利用した認証機能と、書き込み許可リストによる誤操作による改ざん防止機能を提供している。

前者はOLEの偽装機能を利用して、クライアントアプリケーションを実行しているユーザーのログオンアカウントから、ユーザーのアクセスレベルを「プログラム」、「エンジニア」、「スーパーバイザ」、「オペレータ」、「ビューオンリー」、「アクセス不可」の6段階に分けて、権限を抑制するというものである。

後者は、データアクセスサーバー側に、そのサーバーから書き込みが可能なポイント、または書き込みを禁止するポイントのリストを定義し、そこで許された範囲のポイントのみに書き込み処理を制限するものである。このリスト定義にはワイルドカードも使用可能である。

また標準的な機能ではないが、先程から述べられているように、データアクセスサーバーはシステムの機器依存なアクセス部分を個別DLLで提供するために、各システムの状態に促したセキュリティ機能を付加することも可能である。たとえば、データアクセスサーバーの搭載されるノード上にアクセスレベルキーが存在している場合には、その設定内容と上記のログオンアカウントから割り当てられたアクセスレベルを比較して、権限の低い方を実際のアクセスレベルにするといった実装も現実に行われた。

オープン化傾向が進むに連れてセキュリティに対する関心も高まり、今後の検討課題としても残されたセキュリティ観点からの機能の拡張が考えられ、漏洩に対する通信データの暗号化、侵入検知に対するファイアウォールのような機能、監査に対するログ記録などが挙げられる。基幹技術にOLEを用いたことにより、すでに通信データの暗号化も理論的には使用可能であるし、Windows 2000で装備されるCOM+で提供するセキュリティ機能の利用なども期待できる。

#### 5. 機器依存部分の分離

すでに何回か取り上げているが、本ミドルウェアで提供するデータアクセスサーバーの構造は標準的な共通機能を提供する共通部分のサーバーと、システムの機器依存なDLLの部分から構成されている。そしてDLLを入れ替えることによって、各システムに対応するデータアクセスサーバーを実現する。別の観点から言えば、DLLで実現する部分は機器依存なアクセスをする必要最小限の範囲に絞っており、ここを実現するだけで、OPCサポートやセキュリティなども含めて今までに記述してきたすべての機能を実現することができる(図. 4)。よって、固有のシステムに対するデータアクセスサーバーを開発する工数が飛躍的に削減できる。また、この構成は保守の観点からも利点が多く、共通部分のサーバーと機器依存なDLLを独立して管理でき、不具合修正が発生したとしても互いに影響を及ぼして両者を再ビルドする必要もない。

DLLを実装する場合、その範囲は大きく宣言部分と制御部分の2つに分けられる。宣言部分とは対象となるシステムおよびシステムへのデータアクセスに関する属性を定義するものである。システムの属性にはシステム内の名前付け規約に関する部分などがあり、データアクセスの属性とはサポートするデータアクセスサービスの種類や、チューニングに関するもの、たとえば上位アプリケーションから多数のポイントへのアクセス要求が来た場合に、実システムへのアクセス要求をどれくらいに分割して行うかなどの情報が含まれる。

制御部分は実システムへの実際のアクセスに関するもので、先に述べた外部名と内部IDの変換、実システムとの通信チャンネルの接続と切断、リード、ライトなどを実装する。これらの処理はオブジェクト指向に基づいて実現されており、機器依存な部分を隠蔽化することによってDLLへの分離が実現できている(図. 5)。開発者が理解するオブジェクトクラスは主に2つある。一つは実システムとの通信チャンネルを表現したクラスで、前述の実装するべき機能を各々のメソッドとして提供する。開発者はこれらのメソッドを実装することで制御部分を開発できる。この実装内容

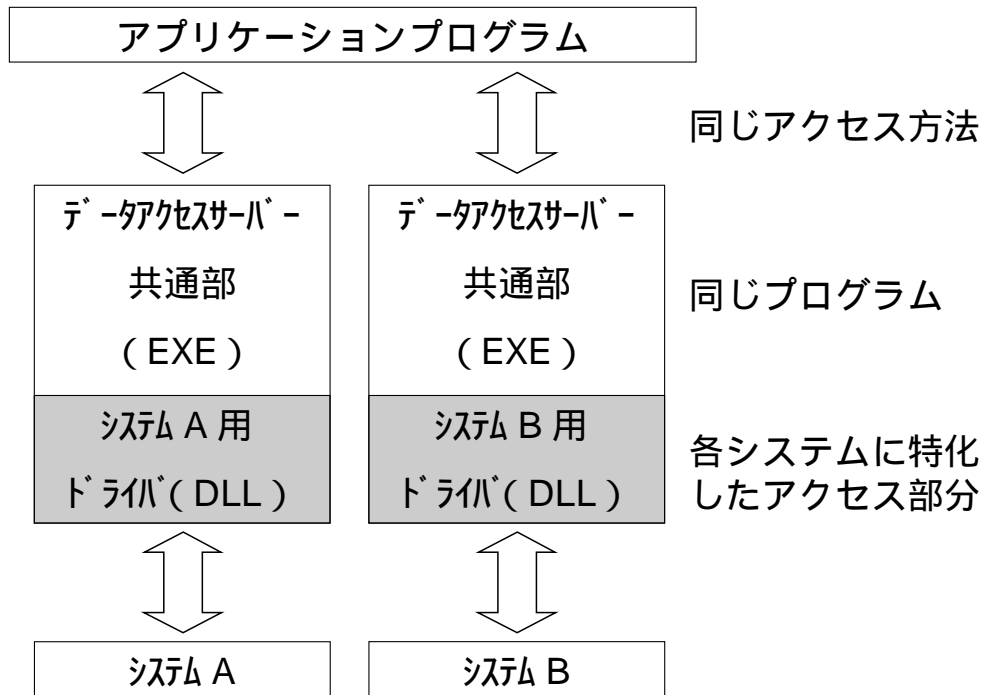


図.4 データアクセスサーバーの構成

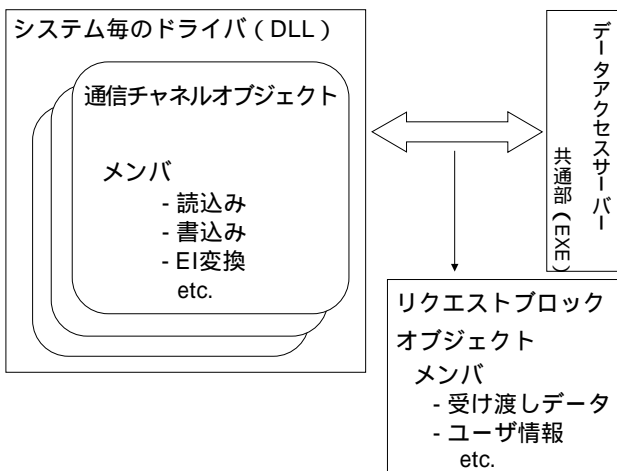


図.5 共通部分とシステム機器依存部分のI/F

として、たとえば、セキュリティの項で触れたアクセスレベルキーのサポートなどを作り込めば、各システムの特長を生かした機能も実現できる。もう一つは共通部分のサーバーとシステム固有のDLLとの間でのやり取りに必要なすべての情報を隠蔽するオブジェクトのクラスで、OLEオートメーションで使われている VARIANT や SAFEARRAY などを用いて表現されるアクセスデータの外に、クライアントユーザーのアクセスレベルなどの情報が受け渡しされる。

加えて、本ミドルウェアで提供するデータアクセスサーバーの開発支援のために、開発者用キットとでも呼べるような環境を用意している。具体的には上記 DLL のソース

コード、実装手順を記述した開発者用ガイド、本ミドルウェアの共通部分であるサーバー、開発した DLL をセットアップするための環境を提供するインストールパッケージ、テスト用プログラムである。ソースコードはスタブ処理を行うテンプレートの Visual C++ のプロジェクト環境になっており、実装必要な部分にコメントがついて検索可能になっている。

さらに、パフォーマンス測定キットとして、パフォーマンス測定プログラム、測定テスト仕様書、表計算ソフトのファイルによる実測結果記入シートも提供し、各システム用のデータアクセスサーバーができる限り似たような条件の元でパフォーマンス比較ができるような仕組みを構築している。

## 6. まとめと今後の課題

各システムで共通なインタフェースを提供するデータアクセス・ミドルウェアを定義、開発した。これによってアプリケーション開発の工数削減を実現した。またデータアクセスサーバーの構造も、システム毎の機器依存部分を個別の DLL で実現したことで実装部分を最小化でき、開発支援環境を提供することによって、その開発工数を削減した。

今後の検討課題としてはセキュリティ機能の充実が挙げられると共に、アプリケーション開発側からは実機システムの存在しないオフライン環境でアプリケーション開発を実現する機能の提供が要求されている。

また、社内インタフェースの標準化という視点からすると、データアクセスのみならず、その他のサービスを共通したインタフェースで提供するミドルウェアの開発が望まれるが、これには、今後のソフトウェア業界の標準化動向を常にトラッキングして、検討していく必要がある。

### 参考文献

「OPC技術概要書 Version2.0」日本OPC協議会技術部会

\* 本文中および図中に登場する製品名は、一般に各発売元の商標または登録商標です。

### 著者所属

開発2部 藤井 稔久