

産業用データベース・ミドルウェアの開発

1W-3

— 基本構想とシステム構成 —

島川 博光[†]竹垣 盛一[†][†]三菱電機(株)産業システム研究所

1 まえがき

近年ではソフトウェアの開発にかかるコストがハードウェア開発にかかるコストを上回ろうとしている。特にプラント・システムでは実時間性が重視されるので、ソフトウェアが個々のプラントに依存する部分が多くなる。産業用データベース・ミドルウェア「プロセス・データベース」は、プラント制御システムを開発するさいに必要となるデータベース機能を抽出し、これを標準ライブラリや標準サーバとして実現することによって、ソフトウェア開発の効率化を図るものである。

本論文では、プロセス・データベースの開発上の基本思想と構成要素について述べる。さらに、本ミドルウェアを設計する上での特徴を述べる。

2 プラント・システムでの要求と構成

プラント・システムでは、実時間を重視したタスクと機能性や柔軟性を重視したタスクが存在する。プラント制御に必要なデータ獲得・コマンド送出といった運転業務におけるタスクは前者に当り、記録データの検索・統計や生産計画立案などの保守・管理業務におけるタスクは後者に当る。

これらタスクを両立させるために、大規模プラント・システムでは制御システムと情報システムが階層的・分散的に配置される。制御システムでは実時間が重視され、情報システムでは機能性や柔軟性が重視される。

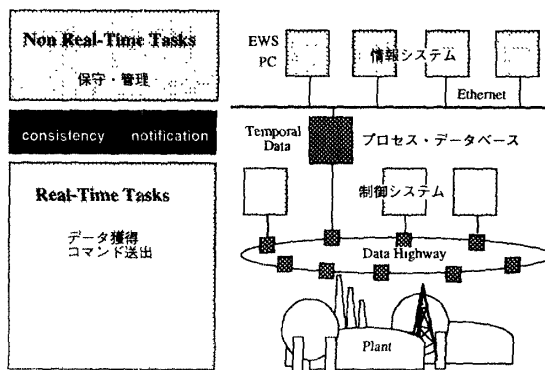


図 1: プラントの構成

産業用データベース・ミドルウェア「プロセス・データベース」は、実時間でデータを獲得し制御システムや情報システムに提供するデータベースである。本ミドル

Development of Middleware for Industrial Database Systems -
Concepts and Components -
Hiromitsu Shimakawa, Morikazu Takegaki
Mitsubishi Electric Corp.

ウェアでは、実時間トランザクションと非実時間トランザクションを切りわけた。前者の実時間性を実時間 OS 上に構築したデータベース・サーバにより保証し、後者を UNIX や Windows 上で実行するようにしている。実時間性を重視する制御システム間で取り交わされるデータを、豊富なソフトウェア資産をもちオープン性に優れた UNIX や Windows 上の情報システムで利用できれば、情報処理機能を容易に構築できる。

3 基本思想

産業用データベースミドルウェア「プロセス・データベース」の開発にあたっては以下の点を重視している。

マルチ・プラットフォームからの利用可能性 実時間で獲得されたデータはさまざまなプラットフォーム上で利用可能であることを期待されている。UNIX EWS は現場における監視・制御で使われ、品質管理や経営管理などの高度な意思決定や知識利用を伴う作業は Windows 環境で行なわれる。特に後者は近年の OLE 環境の普及とともに、部品化システムやコンポーネントウェアを実用化する環境として注目されている。我々はデータベースのサーバ機能を実時間 OS 上で開発し、これと通信可能なクライアント環境を UNIX と Windows の双方に提供することによって、マルチ・プラットフォームからの利用を可能としている。

適材敵所によるモデル採用 実時間データ処理と機能性・柔軟性を重視したデータ処理ではおのずとモデリングは異なる。我々は、実時間処理に対しては単純化した関係データモデルとデータ鮮度に基づく処理モデルを採用する。また我々は機能性を重視する問題として、物体の移動を伴う製造プラントの制御と、実時間で獲得されたデータを Windows 上の OLE 環境で柔軟かつ効率的に使える OPC(OLE for Process Control) 仕様に準拠した環境の提供に取り組んでいる。前者のために時間/場所/物体モデルを提案し、後者のためにデータベースからのデータの流れをデータ共用関係を利用して効率的に各部品に転送する Switched Dataway を提案する。

4 構成要素

プロセス・データベースは、実時間トランザクション管理機構 tact、実時間データ収集・提供機能と能動機構を備えた RTDS、移動する物体の追跡とそれにとまなうビュー生成を可能にする RTVS、Windows 上の複数の部品間でプロセス・データを効率的に利用できる OPC-X からなる。上記の思想を反映させるべく、各構成要素は次のような特性を意識して設計されている。

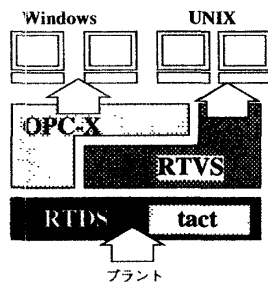


図2: プロセス・データベースの構成要素

4.1 オブジェクト指向に基づく設計

オブジェクト指向の考え方は、データベースのサーバ内ではソフトウェアの保守性向上につながり、クライアント上ではコンポーネントウェアによるシステムの安価な構築を可能にする。

4.1.1 サーバとスタブ関数

プロセス・データベース環境では、tact内の実時間トランザクション管理サーバと、RTDSとRTVSの2つのデータベース・サーバが動作している。これらはすべてサーバ・プロセスとスタブ関数から構成されており、各サーバのクライアントはこのスタブ関数を通してサーバにアクセスする。すなわち各サーバは厳密に定義されたインタフェースをもつオブジェクトとして設計されている。こうすることによって、サーバ内では思いがけない手続きが起動されることがなくなり極めて小さなコード量でサーバが実現できることとなった。

4.1.2 ソフトウェア部品

信頼できるプラント・システムを短期間で構築する上でソフトウェア部品の合成によるシステム構築は有益である。しかし、ソフトウェア部品は可搬性のために処理が非効率的である。産業応用では可搬性と効率性の両立が求められる。通知機能をもったオートメーション・サーバであるOPC-XはOLE環境に準拠して作成された部品がサーバからのデータに効率的にアクセスできるように設計されている。Switched Datawayは部品間で共用されるデータの、データベースからクライアント側の部品までの流れを管理する。

4.2 仮想化と透明化

4.2.1 スケジューリング・ポリシー

tactでは、共有メモリ上のスケジュールに対してスケジューラは生産者、トランザクション起動機構は消費者の関係にある。これらの独立性が高いためにRMA[1]、Frame Scheduling[2]など複数のスケジューリング・ポリシーにtactは対応できる。RTDSはtact上に構築されるが、RTDSからはこれらのポリシーは隠蔽され問題に適したポリシーが選択できる。

4.2.2 データベース内の仮想工場

バッチプロセスで代表されるようなロットがライン内を移動していくプラントの制御のために「時間軸上で複数の場所間を物体が移動する」という時間/場所/物体モデルを提案する。ロットや工程といったプラント構成

要素をオブジェクトとして定義した上で、このモデルに基づきデータベース内に仮想プラントを表現する。

4.2.3 異なる環境間のデータの流れ

RTDS、RTVSはUNIXまたはUNIXに準拠した実時間OS上に構築されるので、これらサーバとWindows環境とのインタフェースが必要である。Switched Datawayは、Windows上の部品からはみずからに接続されたデータ転送チャンネルに見え、UNIX上のサーバからはひとつのクライアントに見える。すなわち、Windows上の複数の部品には仮想チャンネルを提供し、UNIXの世界からはWindows上にある複数の部品を隠蔽する。

4.3 セマンティクスを生かした効率性

4.3.1 簡潔なデータ管理

RTDSでは、獲得されたデータは保持されているデータの最後尾に追加されていくという性質を利用して、データ管理手続きを簡潔化・効率化している。格納のためのバッファはリング状に構成され、排他制御は更新されている位置情報だけに対して行なわれる。RTVSでは、ロットの工程間の移動はロットを表すインスタンスへのリンクを工程に対応したインスタンス間で付け変えるだけで表現されている。

4.3.2 データ鮮度に基づく締切

実時間データ処理をすべてのトランザクションに対して保証することは事実上困難である。我々は産業応用分野では「要求されたデータが新しければ新しいほど、その処理に対する締切は厳密になる」と考え、データ鮮度に応じた厳しさの締切[3]を設定することにより、非周期に発生する能動トランザクションと実時間トランザクションとの統合を簡潔に実現している。

5 あとがき

本稿では、産業用データベース・ミドルウェアの基本思想と構成要素について述べた。本ミドルウェアの開発では、マルチプラットフォームからの利用可能性を重視し、各問題に適したモデルを採用することにより手続きの単純化・効率化を実現している。

参考文献

- [1] C.L.Liu and J.W.Layland, Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *J. ACM*, Vol.20, No.1, pp.46-61, 1973
- [2] K.Ramamritham, et.al, Efficient Scheduling Algorithms for Real-Time Multiprocessor Systems, *IEEE Trans. Parallel and Distributed Systems*, Vol.1, No.2, 1990, pp.184-194
- [3] H.Shimakawa, et.al, Active Transactions Integrated with Real-Time Transactions according to Data Freshness, *Proc.of 3rd Real-Time Technologies and Application Symposium*, 1997