

分散生産ライン管理・保守システムの オブジェクト指向技術に基づく設計と実装

楠 和 浩^{†,†††} 今 井 功[†] 大 谷 治 之[†]
中 川 路 哲 男[†] 大 島 道 隆^{††} 牛 島 和 夫^{†††}

運転状況監視と保守という2つの観点から、制御機器の1つであるPLCのオブジェクトモデル化を実施し、さらにこのモデルを分散処理環境で実行するために必要な分散オブジェクト管理機能の試作および性能評価を行った。まず、PLCのオブジェクトモデル化に際しては、PLCを、物理的なモジュールであり保守・管理の対象となる各種ハードウェアユニットと、運転状況を管理するための計算資源である内部メモリに分けたモデル化を行った。また、その際PLCに接続される入出力機器は、ネットワークに直接接続され自律動作するような機器にも対応できるようなモデルとした。一方、PLCへの実装を行う分散オブジェクト管理機能は、OMG CORBAに準拠し、CPU性能やメモリ性能が限定されるPLCへの実装が可能な設計とした。メモリ性能と応答時間性能の2つの評価の結果、設計したオブジェクトモデルおよび分散オブジェクト管理機能はPLCに対して有効であることが検証された。

A Design and Implementation of Object-oriented Distributed Manufacturing Line Monitoring and Maintenance System

KAZUHIRO KUSUNOKI,^{†,†††} ISAO IMAI,[†] HARUYUKI OHTANI,[†]
TETSUO NAKAKAWAJI,[†] MICHITAKA OHSHIMA^{††}
and KAZUO USHIJIMA^{†††}

This paper discusses the object model of PLC designed for manufacturing line monitoring and maintenance, and investigates the architecture and performance of distributed object management middleware to communicate between designed PLC objects. We designed PLC related objects from two viewpoints. We designed hardware related objects which are modeled from physical control devices of PLC for remote maintenance and management. And we designed memory related objects which are modeled from internal calculation resources in PLC for monitoring manufacturing line status. Modeling of I/O device is aimed to be applied to autonomous device that is directly connected to network. Our distributed object management middleware is based on OMG CORBA and is able to be implemented to PLC to which CPU and memory performance is restricted. We evaluated proposed PLC model and distributed object management middleware from the viewpoint of necessary memory capacity and response time. From that evaluation our PLC model and middleware is effective to distributed manufacturing line monitoring and maintenance system.

1. はじめに

製品の機能やデザインに対する要求の多様化とともに、生産形態は多品種少量生産から多品種変量生産

に変化しつつある¹⁾。個々の仕様に応じた製品を大量に生産するためには、低コストで高効率な生産ラインの構築を行うとともに、生産ラインの変更や拡張に関する柔軟性が必要になる^{2),3)}。

製品に対する機能変更やデザイン変更を速やかに生産システムに反映させ、個別仕様大量生産に対応した生産システムを構成するためには、次の条件を満足させる必要がある。

設計/製造/生産管理部門の連携強化 仕様変更にと
もなう設計データ(CAD/CAM)の変更が、生産
ラインに迅速に反映できること。その際、各部門

† 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

†† 三菱電機株式会社名古屋製作所
Nagoya Works, Mitsubishi Electric Corporation

††† 九州大学大学院システム情報科学研究科
Graduate School of Information Science and Electrical Engineering, Kyushu University

が地理的に離れている場合でも適用可能なこと。

分散型最適生産ラインの構築 生産システムを構成している特徴のあるサブシステム, または構成要素の運用上の特性を考慮した, 生産ラインの各工程に最適な制御機器および入出力機器を選択でき, それぞれが相互接続可能なこと. 制御機器としては, NC (Numerical Control) 工作機械, ロボット, PLC (Programmable Logic Controller) および入出力機器がある。

生産ライン制御プログラム開発効率の向上 生産品目の変更の際に生じる生産ラインの機器構成変更にもなる制御プログラムの変更が容易であること。

生産ライン稼働率の向上 生産ラインの生産状況がリアルタイムに把握可能なこと. また, 仮に生産ラインを構成する制御機器に障害が発生しても, 迅速に復旧できること。

これらの必要条件を満足するために, 我々は, 次の解決策を本論文で提示する. すなわち,

- オブジェクト指向技術による生産システム構成機器のモデリングと実装
- 分散オブジェクト管理機構の適用による生産ライン管理機能との連携機能の提供

の2点である。

具体的には, 運転状況監視と保守という2つの観点から PLC をオブジェクトモデル化し, さらに必ずしも性能や資源が十分でない PLC に分散オブジェクト管理機能を実装した統合システムを設計し, その試作を通して実現性と有効性を評価することを目的とする。

本論文では, 2章で解決すべき課題を述べ, 3章で PLC のオブジェクトモデルと, PLC への分散オブジェクト管理機能の実装方式について述べる. さらに4章で試作システムの性能評価結果を述べる。

2. 分散生産ライン管理・保守システムの課題と解決策

工場の生産ラインは, たとえば自動車組み立てラインでは, ドアの取付け工程や車体取付け工程など複数の工程から構成されている. それぞれの工程には, ロボットや NC 工作機械が設置され, コンベアの制御や生産ラインの状態を監視するための入出力機器が PLC によって制御されている. 各工程内および工程間では最適な制御を行うことができる PLC を選択する必要があるが, 現在の PLC は製造会社によって異なるアーキテクチャをとっており, さらに PLC 間のネットワークも独自プロトコルを採用しているために, 異機種 PLC を利用した統合制御を実現できないとい

う問題がある。

また, 生産ライン稼働中に状況を監視するシステムでは, 生産ライン監視アプリケーションと PLC とがネットワークで接続された分散システムとなる. PLC が生産ライン監視アプリケーションに公開しているインタフェースは, PLC が, その内部で情報を保持するメモリの物理的な構成を直接公開したものである. さらに, PLC に接続される入出力機器は, それらが接続されている PLC の接点のメモリ番地を通してしか操作できない. したがって, 生産ライン監視アプリケーションは PLC の内部構造の変化や入出力機器の接続先変更の影響を直接受けることになり, 生産ラインの変更や拡張に対する柔軟性が低くなるとともに, アプリケーションの保守性も悪くなるという問題がある。

これを解決するために, オブジェクト指向技術を適用する. 特に, 制御機器は一般に現在の状態と機器制御命令を備えているため, オブジェクト指向技術との親和性が高く, さらにオブジェクト指向技術を適用することでソフトウェアの生産性向上も期待できる。

これまでに, オブジェクト指向技術を用いることによって, 機種依存部分をオブジェクト内に隠蔽し, 機種非依存のインタフェースを提供し, さらにそれらのオブジェクトをネットワーク経由で連携させる分散オブジェクト管理技術を適用して異機種間相互接続性を確保した統合生産システムを構築する研究が行われている^{4),5)}。

これらの研究は, 適応対象が NC 工作機械に限定されている. 生産ラインを構成する重要な機器である PLC のオブジェクトモデルはこれまで提案されていない。

また, 生産ラインの生産効率向上のためには, 制御機器の保守を効率良く実施し, 仮に障害が発生した場合でも, 短時間で障害原因の解析と, 機器の復旧を実行する必要がある。

そのためには, NC 工作機械や PLC などの制御機器, および PLC が制御している入出力機器の現在の状態を監視できる機能だけでなく, 障害発生時のイベント通知機能や, ハードウェアの連続運転時間などの保守に必要な情報を検索する機能が必要である. しかし, これまでの研究⁶⁾では, 保守を実行するための機能と, 生産ライン稼働中に生産状況を監視するための機能との統合方式は提案されていない。

本論文では, これらの問題を解決する PLC のオブジェクトモデルを提案する。

一方, 生産ラインは複数の制御機器をネットワークで接続した分散システムである. したがって, 機器単

体のオブジェクトモデルと同時に、分散環境でのオブジェクト間管理機能が必要である。これまでの分散オブジェクト管理機能は、主に WS (Work Station), PC (Personal Computer) あるいは、汎用機で実行される、いわゆる情報系システムを対象に開発されているため、一般的に高性能な計算資源（特に、CPU 性能、メモリ性能）を必要とする。したがって、生産システムに対して分散オブジェクト技術を適用したこれまでの研究でも、対象制御機器は、一般的に高性能な計算資源を持つ NC 工作機械に限られている。

本論文では、生産ラインに必要な分散オブジェクト管理機能を定義するとともに、処理速度を考慮した実装方法を提案する。

3. 設計および実装方法

本章では、まず 3.1 節で生産ライン稼働中の生産状況管理とともに、機器の保守・管理が実現できる PLC のオブジェクトモデルを提案し、次に 3.2 節で、生産ラインに必要な分散オブジェクト管理機能の定義と、各装置における実装方式を示す。さらに 3.3 節で、分散環境で動作する各オブジェクト（3.1 節で提案する PLC オブジェクト）を分散環境で一意に識別するためのオブジェクト識別子の定義方法について述べる。

3.1 PLC のオブジェクトモデル化

これまでの生産システムでは、入出力機器の制御または状態監視は、各入出力機器が接続された接点に対応した入出力メモリ番地を指定して実行されている。制御プログラム（シーケンス制御プログラム）は制御専用言語（ラダー言語など）で作成する。作成した制御プログラムは、PLC 起動時に自動的に PLC 内で実行され、指定された入出力メモリ番地の状態に応じた制御動作（入出力メモリの操作）を繰り返し実行する。一方、生産ラインの稼働状況や生産状況を監視する生産ライン監視プログラムは、制御プログラムとは独立に、LAN またはシリアル回線で接続された計算機上で動作する。生産ライン監視は、PLC 内の入出力メモリ番号を指定することによって実現されている。

したがって、生産品目の変更時に生産ラインの機器構成を変更し入出力機器の接続先が変更となった場合には、シーケンス制御プログラムだけでなく、生産ライン監視プログラムも変更する必要がある。

しかしながら、各入出力機器が PLC のどの接点に接続されているかは、PLC 内でのみ必要な物理情報である。生産ライン監視プログラムが、入出力メモリ番地という物理情報を意識せずに、抽象度の高い論理名を使用して状態監視を実行できれば、ソフトウェア

の保守・管理が非常に容易になる。

オブジェクト指向技術を採用することによって、上記の問題も解決可能である。

PLC をオブジェクトモデル化するにあたっては、次の設計方針を設定した。

- (1) 生産ライン稼働中の生産状況管理とともに、機器の保守・管理が実現できるオブジェクトモデルとする。その際、生産状況管理を実施するアプリケーションと、機器保守を実施するアプリケーションとで、取得できる情報の種類や、情報書き込み制限が異なっても対処しやすいモデルとする。
- (2) 入出力機器は、現状では PLC がすべて管理する形態をとっているが、たとえば LONWORKS⁷⁾ のように、ネットワークに直接接続され自律動作可能な入出力機器も出現している。したがって、入出力機器と PLC の機能を分離し、自律動作可能な入出力機器がネットワーク接続により動作する形態の生産システムにおいても、利用可能なオブジェクトモデルとする。

設計方針に従ってモデル化した結果を図 1 に示す。ここで記述方法は OMT 法⁸⁾に基づいている。

各クラスの詳細は以下のとおりである。

【PLC クラス】 本モデルでは、機器の保守・管理の側面から定義が必要なハードウェア関連クラスと、生産状況管理に必要なメモリ関連クラスを定義する。本クラスは、ハードウェア関連クラスとメモリ関連クラスを統合するために用意したクラスである。本クラスは、PLC を構成するすべてのオブジェクトを管理するとともに、PLC の起動および停止機能を提供する。

【ハードウェアクラス】 機器の保守・管理はハードウェア単位で実施される。本クラスは、保守・管理の観点から各種ハードウェアユニットに共通に必要な属性を管理するために必要なクラスである。具体的には、このクラスでは、設置日や稼働開始日などの保守情報とともに、論理名を属性として保持する。

【各種ハードウェアユニットクラス】 PLC は、図 2 に示すように、ベースユニットに複数のハードウェアユニットを装着して必要な機能を満足させる。ここでは、CPU ユニット、電源ユニット、通信ユニット、入力ユニット、出力ユニットの各ハードウェアユニットに対応するクラスを総称してハードウェアユニットクラスと呼ぶ。本クラスは、ハードウェアの製造会社名や型名などの機器識別情報

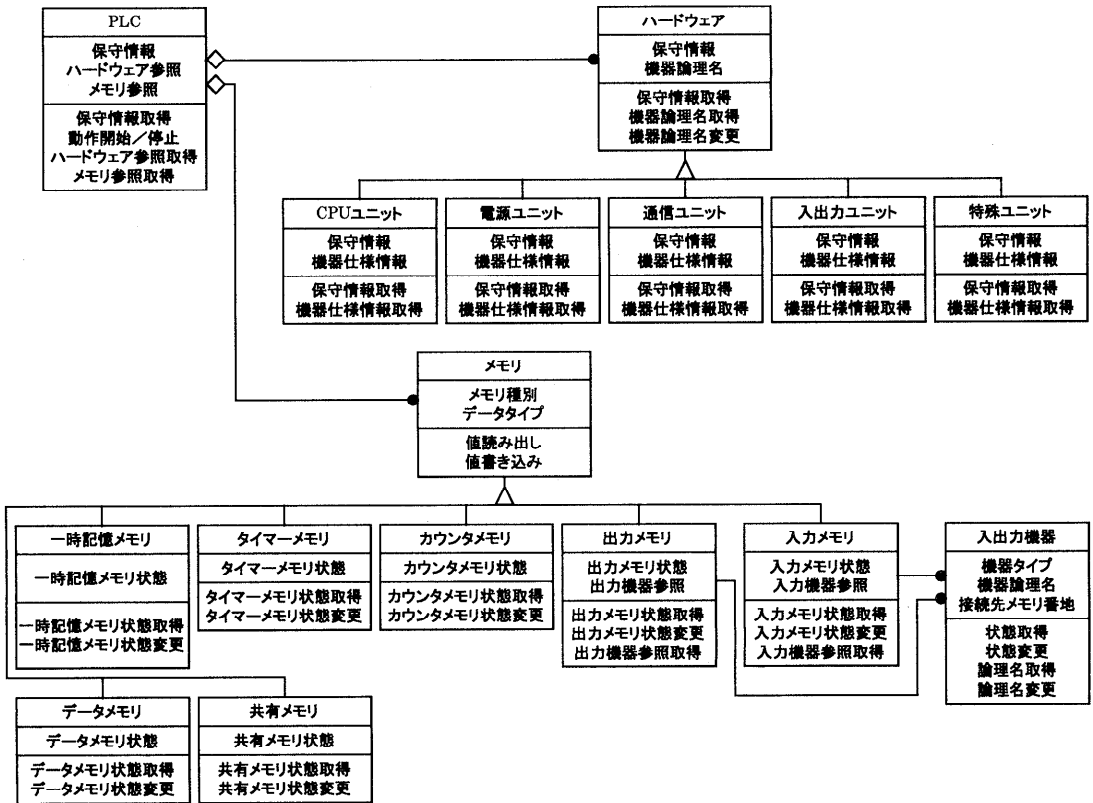


図1 PLCのオブジェクト構成図
Fig.1 Object model of PLC.

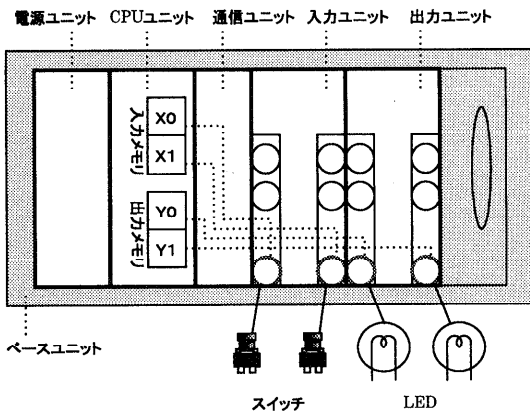


図2 PLCの構成例
Fig.2 Example of PLC modules.

のほか、稼働開始日時や総稼働時間、さらにシステムROMのバージョン番号などの保守情報を保持する。

【メモリクラス】 PLCはシーケンス制御を実行するために、内部に計算資源である用途別メモリを内蔵している。生産システム稼働中の生産状況は、

各用途別メモリに反映される。本クラスは、各用途別メモリクラスの親クラスであり、メモリ種別とデータタイプ(ビットタイプメモリ、バイトタイプメモリ)を属性として保持している。

【用途別メモリクラス】 一時記憶メモリ、タイマーメモリ、カウンタメモリ、データメモリ、共有メモリ、入力メモリ、および出力メモリクラスを総称して、ここでは用途別メモリクラスと呼ぶ。PLC内のシーケンス制御プログラムは、入出力を含む各メモリの状態を検査して制御を実行する。その際、シーケンス入出力ユニットに直結された入出力機器は、入出力メモリを通して制御される。

【入出力機器クラス】 現状の制御システムでは、プログラムは、入出力機器が接続された入出力メモリ番地で、各入出力機器を認識している。たとえば、図2に示すように、あるスイッチが入力メモリの0番地(一般的には、“X0”と記述される)に接続されている場合には、制御プログラムは、このスイッチの状態を取得するために、番地X0を指定して状態値を読み込む。本クラスは、入出力機器がPLCに接続されている場合には、物理

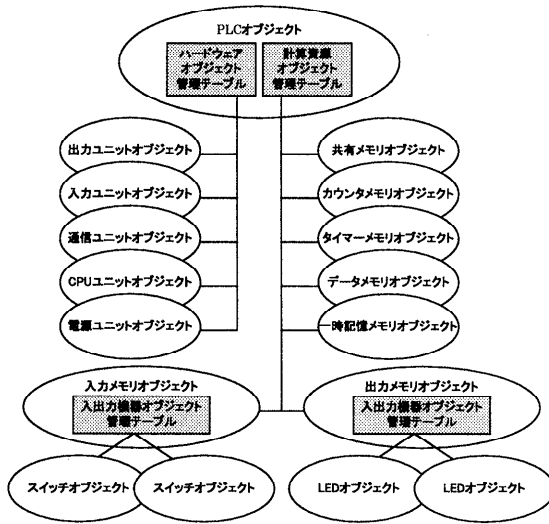


図3 オブジェクト構成例

Fig. 3 Example of object configuration.

情報であるメモリ番地ではなく、論理名で認識できるようにするために設計されたクラスである。また、自律動作可能な入出力機器がネットワーク接続により動作する形態の生産システムにおいても利用可能なモデルとするために、先に定義した本クラス以外のクラスとは疎な関係として定義する。本クラスから生成されたオブジェクトは入出力メモリオブジェクトによって管理され、接続先入出力メモリ番地と、論理名を属性情報として保持する。

入力機器として2つのスイッチが、出力機器として2つの発光ダイオード(LED)が接続された場合(図2)を例として、本論文で提案するオブジェクト構成を、図3に示す。

まず、PLCクラスより生成されたPLCオブジェクトが存在する。このオブジェクトは、各種ハードウェアユニットクラスより生成された複数のオブジェクトと、用途別メモリクラスより生成された複数のオブジェクトを管理する。ここでは入力機器が2つ、出力機器が2つと仮定しているため、入力ユニット、出力ユニットは、それぞれ1つで対応可能である。したがって、入力ユニットオブジェクト、出力ユニットオブジェクトは、ともに1つだけ存在する。

入出力機器クラスから生成されたオブジェクトは、システムの構成に依存して個数が変化する。図3では、2つのスイッチに対応して2つのスイッチオブジェクトが、2つのLEDに対応して2つのLEDオブジェクトが入出力機器クラスから生成される。

たとえば、2つのスイッチがそれぞれX0, X1というメモリ番地に接続されている場合を仮定する。スイッチオブジェクトは、内部属性(接続先接点)値として、それぞれX0またはX1を保持する。また、入力メモリオブジェクトの入出力機器オブジェクト管理テーブルでは、X0およびX1と、それぞれに対応するスイッチオブジェクトの対応関係を保持する。

メモリ番地X0に接続されたスイッチが、メモリ番地X3に接続先が変更になった場合には、スイッチオブジェクトの内部属性(接続先接点)の値をX0から、X3に変更し、入力メモリオブジェクトの入出力機器オブジェクト管理テーブルの対応場所を変更する。

つまり、本オブジェクトモデルでは、機器が接続されたメモリ番地をオブジェクト内に隠蔽しており、生産ライン監視プログラムからは入出力機器を論理名で識別するため柔軟性の高いシステム構築が可能になる。

3.2 制御機器への分散オブジェクト管理の実現

制御機器に分散オブジェクト管理機能を実現するにあたり、次の設計方針を設定した。

- (1) 生産ラインの各工程で、高品質の製品を高い生産効率で安全に生産するためには、各工程に最も適した制御機器を使用する必要がある。その場合、必ずしも同一機種あるいは同一会社の製品が使用されるとは限らない。したがって、複数の会社の製品間で連携制御可能な分散オブジェクト管理機能を実装する(開放性)。
- (2) 制御機器は一般にCPU性能およびメモリ性能が限定される。また、2次記憶装置を装備していないものも多い。したがって、提供機能を限定して実行モジュールサイズを小さくする(コンパクト性)。
- (3) 統合管理装置あるいは遠隔保守・管理装置からのアクセスに対する実時間応答性能は、各PLC内での制御動作(シーケンス制御プログラムの実行速度)に比べると厳しくないが、一般のOA系の遠隔アクセスと比較すれば高速性が要求される。さらに、今後、オブジェクト指向技術に基づく分散制御システムが実現される可能性もあり、高速実行が可能な処理構造とする(高速処理性)。
- (4) クライアント(統合制御管理装置)/サーバオブジェクト(制御機器)間の通信は、異機種間接続可能なプロトコルを採用し、さらにWWWブラウザをクライアントとするシステム形態の実現を考慮したプロトコルを採用する(拡張性)。それぞれの設計方針に対する解決策の概要を以下に

示す。

まず、(1) の設計条件を満足するために、OMG (Object Management Group) が規約化したデファクト標準である CORBA (Common Object Request Broker Architecture)⁹⁾ に準拠した分散オブジェクト管理機能 (ORB: Object Request Broker) を実現した。次に、この分散オブジェクト管理機能の適用対象である制御システムの特性を考慮し、CORBA 仕様のうち、静的なオブジェクト生成機能とメソッド解決機能だけを実装することで、設計条件 (2) を解決した。さらに、メソッド処理は、IDL コンパイラが生成する汎用処理プログラムを使用せずに最適化を行い、ORB およびオブジェクトを 1 プロセスで実現してプロセス間スイッチのオーバーヘッドを削減する方法により、設計条件 (3) を解決した。また、設計条件 (4) については、CORBA2.0 で規定されている ORB 間相互接続プロトコルである IIOP (Internet Inter-ORB Protocol) を採用して解決した。

ここで、設計条件 (2) および (3) につき、具体的な解決策を以下に述べる。

CORBA 仕様では、オブジェクトの生成方法やオブジェクトに対するメソッド要求方法が広範囲のアプリケーションに対応可能のように多数定義されている。我々は、制御システムへの適用として特性を考慮し、次の機能に限定して実装することで設計方針 (2) を満足した。つまり、制御システムにおいては、ネットワークに接続される機器の台数や、それぞれの機器に接続される入出力機器の個数、さらに各制御機器で実行されるアプリケーションは、制御システム設計時に決定され、通常、生産ライン稼働中に変更されることは少ない。また、クライアント (統合制御管理装置) 側も、オブジェクトが提供するメソッドを動的に検索することはない。したがって、CORBA 仕様のうち、静的なオブジェクト生成機能と、静的なメソッド解決機能だけを実装することで分散オブジェクト管理機能の実行モジュールサイズを制限した。

設計条件 (3) の具体的な解決策を示すために、まず、図 4 に制御機器への分散オブジェクト管理のソフトウェアアーキテクチャを示す。

オブジェクトは基本オブジェクト群と、応用オブジェクトの 2 種類に分類され、処理方法が異なる。

基本オブジェクト群は、3.1 節で述べた PLC オブジェクトである。これらのオブジェクトは、PLC で共通なオブジェクトである。一方、応用オブジェクトは、システム依存の情報処理機能を提供するオブジェクトである。たとえば、センサの ON/OFF 回数の時

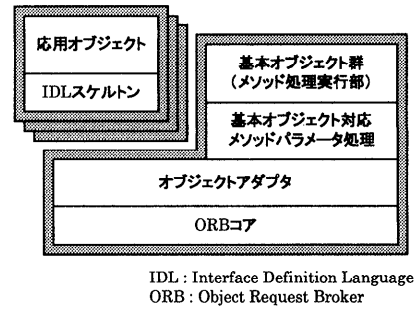


図 4 制御機器分散オブジェクト管理のアーキテクチャ構造
Fig. 4 Software architecture of distributed object management for controller.

間平均処理を提供するオブジェクトである。

基本オブジェクト群が提供するメソッド処理のための各機能単位の概要を示す。

【ORB コア】 クライアント/サーバオブジェクト間の通信プロトコル処理を実行する。

【オブジェクトアダプタ】 オブジェクトに対してネットワーク内で一意な識別子を生成および付与するとともに、クライアントからのメソッド要求を適切なオブジェクトに振り分ける処理を実行する。

【基本オブジェクト対応メソッドパラメータ処理】メソッド処理を実行するための、機器アーキテクチャに依存しないパラメータの再組立てや、応答を送信するための送信メッセージの作成を実行する。汎用的な分散オブジェクト管理機能では、インタフェースを定義する IDL (Interface Definition Language) によって記述されたオブジェクト定義から、IDL コンパイラによって自動的に処理プログラムが生成されるが、本実装では、以下の理由によりメソッドごとに個別に処理プログラムを実装した。

- IDL コンパイラから生成される処理プログラムは、汎用的な処理を実行するので、一般的にモジュールサイズが大きい。
- オブジェクトとそれらが提供するメソッドが決定されており、かつメソッド総数が少ないので、個別対応可能である。
- いくつかのメソッド処理の共通部分をまとめる処理プログラムの最適化 (高速化および軽量化) を図ることが可能である。

【基本オブジェクト群】 実際のメソッド処理を実行する。

基本オブジェクト群が提供するメソッド処理は、処理速度を向上させるため、すべて 1 プロセスで実行する。

応用オブジェクト群の処理は、汎用の分散オブジェクト管理機能と同様に IDL コンパイラを使用した処理形態をとる。

3.3 分散環境におけるオブジェクト識別子の定義

3.2 節で設計方針(4)を満たすために、CORBA2.0 で規定されている ORB 間相互接続プロトコル IIOP を採用すると述べた。IIOP を使用した分散オブジェクト環境では、オブジェクトの識別は、IOR (Interoperable Object Reference) という識別子で実現する。IOR は、図 5 の構造をしている。図 5 において、網掛け部分はデータの型を示している。

クライアントからのメソッド要求の結果、サーバ側では IOR 内のオブジェクトキーと、メソッド名およびそのパラメータを受信する。オブジェクトアダプタは、指定されたオブジェクトキーから実際にメソッド処理を実行するオブジェクトを検索する。

オブジェクトキーの採番方法は、サーバオブジェクトを管理するオブジェクトアダプタで任意に決定する。大規模システム構築用 PLC では、2000 を超える入出力機器の制御を実行する。つまり、3.1 節で述べた入出力機器クラスを基に生成するオブジェクトが 2000 を超える。したがって、対象となるオブジェクトの検索を高速化する必要がある。

汎用の ORB では、オブジェクトキーは、たとえば、生成された順番に整数値を割り当てるなどの方法を行っている。これに対して、制御機器では、3.1 節で示したように、生成されるクラスが明確に規定されていることを利用して、オブジェクトキーを構造化することが可能である。

本実装では、次の採番方法を採用した。

オブジェクトキーの型は、OCTET 列であり、その長さは任意であるが、文字列を基準とした検索は時間を要するため、32 ビット整数と見なしたキーを採用する。FA 機器のように ORB 上で実行されるオブジェクトの種類があらかじめ分かっている場合には、各クラスに対応して上位 16 ビットを使用してクラスごと

に識別番号を与える。さらに、入出力機器のように、FA システムに依存して数の変化するオブジェクトに対応するため、下位 16 ビットにオブジェクト生成順に識別番号を付与する。

これにより、入出力機器数に依存しないオブジェクト検索を実現することができる。

4. システム評価

4.1 メモリ性能

ここでは、我々が設計した分散オブジェクト管理機能とオブジェクトモデル化を行った PLC オブジェクトが、メモリ資源が十分でない環境に実装可能かどうかを試作システムでの必要メモリ容量を基に評価する。

まず、分散オブジェクト管理機能 (ORB) の実行に必要なメモリ容量 (図 4 の ORB コアとオブジェクトアダプタが使用するメモリ容量) は、228 KB である。

さらに、システム規模に応じた入出力機器の増加に対する必要メモリ容量の変化を表 1 に示す。表 1 で示す必要メモリ容量は、分散オブジェクト管理 (ORB) と、3.1 節で示した PLC 対応オブジェクトを含めたものである。

システム規模が大きくなると、入出力機器が増加する。それにともない 3.1 節で示したオブジェクトモデルでは、入出力メモリオブジェクトおよび入出力機器オブジェクトの総数が増加する。さらに、入出力接点が増大することで、シーケンス制御を実施するのに必要な計算資源が増加し、結果として入出力メモリ以外のメモリオブジェクトも増加する。

ここでは、三菱電機製汎用シーケンサ (PLC と同意) の仕様を参照し、入出力点数が 512 以下の場合には、同社製汎用シーケンサ MELSEC-A2SCPU に準じたメモリオブジェクトを、入出力点数が 512 を超える場合には、MELSEC-A3UCPU に準じたメモリオブジェクトを作成して測定を実施した。

また、必要なメモリ容量は、クラスの属性内容に依存する。本論文で述べた PLC オブジェクトに関して最もメモリ容量に影響するのは、入出力機器オブジェクトや、各メモリが属性として持つコメントおよび名

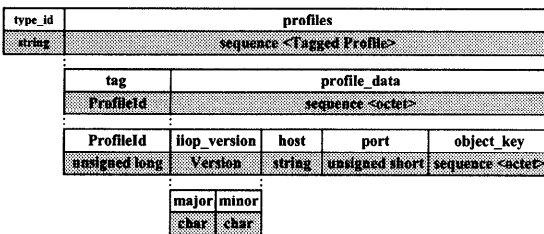


図 5 IOR のデータ構造
Fig. 5 Data structure of IOR.

表 1 メモリ性能
Table 1 Memory performance.

入出力点数	必要メモリ容量 [KB]
128	564
256	652
512	836
1024	1904
2048	2640

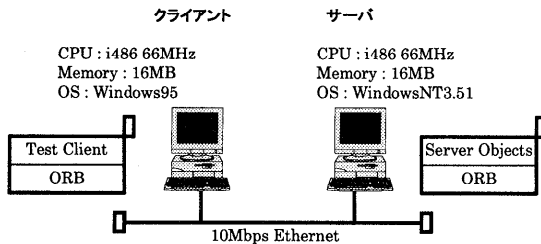


図6 性能測定環境

Fig. 6 Performance evaluation environment.

前などの文字列である。ここでは、すべて 20 文字の配列で測定を行った。

たとえば、EPSON 社製カード PC による PLC の実現を考えた場合、使用できるメモリ容量は最低でも 4 MB 用意されている。今回我々が実装を行ったシステムでは、入出力点数が 2048 点でも 2.6 MB で実現可能であり、メモリ資源が十分でない PLC などの制御機器にも十分実装が可能である。

4.2 応答性能評価

図 6 に、応答性能測定環境を示す。

今回の実装では、ORB は、PLC 内に実装せず、PLC とネットワークで接続したパソコンに実装している。しかしながら、我々は PLC 内への ORB の実装を目標としている。PLC 内に ORB を実装した場合には、PLC の内部メモリに対するアクセスは、内部バスを通して実行される。内部メモリアクセスにかかる処理時間は、クライアント/サーバ間の ORB 処理時間に比べて無視できるほど小さいと考えられる。したがって、この測定においては、実際の入出力機器へのアクセスを実施していない。

まず、図 7 に入出力機器の状態取得にかかる応答性能を示す。応答時間の単位はミリ秒である。

ここで性能評価は、

- 入力メモリまたは出力メモリを指定し、直結された入出力機器の状態を取得するもの（図 7 において、read through memory と記述）
- 入力機器オブジェクトまたは出力機器オブジェクトを指定し、その状態を取得するもの（図 7 において、read directly from device と記述）

の 4 種類の応答時間測定を実施した。

図 7 において、入出力点数に対して応答時間の増加の割合が高い 2 本のグラフは、入力メモリおよび出力メモリを指定して、直結された入出力機器の状態を取得した場合のものである。対して、傾きの少ない 2 本のグラフ（図 7 ではほとんど重なって 1 本に見える）は、入力機器および出力機器オブジェクトに対して状

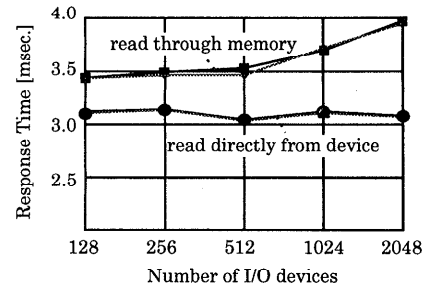


図7 入出力機器の状態取得の応答時間

Fig. 7 Response time to get I/O device status.

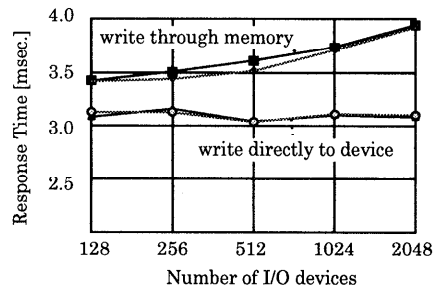


図8 入出力機器の状態変更の応答時間

Fig. 8 Response time to change I/O device status.

態取得メソッドを発行した場合の応答時間である。

次に、図 8 に入出力機器の状態変更にかかる応答性能を示す。この場合も応答時間の単位はミリ秒である。

さらに、状態取得に関する性能評価と同様に、

- 入力メモリまたは出力メモリを指定し、直結された入出力機器の状態を変更するもの（図 8 において、write through memory と記述）
- 入力機器オブジェクトまたは出力機器オブジェクトを指定し、その状態を変更するもの（図 8 において、write directly from device と記述）

の 4 種類の応答時間測定を実施した。

図 7 と同様にして、図 8 でも、入出力点数に対して応答時間の増加の割合が高い 2 本のグラフは、入力メモリおよび出力メモリを指定して、直結された入出力機器の状態を変更した場合のものであり、傾きの少ない 2 本のグラフは、入力機器および出力機器オブジェクトに対して状態変更メソッドを発行した場合の応答時間である。

これらの応答時間を見ると、入出力機器オブジェクトに対するメソッド処理は、最大 3.2 ミリ秒以内の応答性能を達成している。また、メモリオブジェクトを通して入出力機器の状態の取得または変更を行う場合でも最大 4 ミリ秒以内でメソッド処理を完了している。

生産ラインの状態監視を行うための代表的な監視

ツールの場合、最小 50 ミリ秒の一定周期で機器の状態を監視することができる¹⁰⁾。

したがって、管理ツール側で最小 50 ミリ秒周期で状態読み出しを行っても、本実装システムにおける応答時間が最大 4 ミリ秒以下であることより、監視制御に必要な性能は達成することができた。

ここで、メモリオブジェクトを通して入出力機器の状態を取得または変更する場合に、入出力点数に比例して応答時間が大きくなるのは、以下の理由による。

PLC のオブジェクトモデルにおいて、メモリは、使用用途別にクラス定義を行った。したがって、たとえば入力メモリの場合には、対象 PLC が用意する入力点数分の状態値および番地情報を属性情報として保持する。番地情報は、たとえば入力メモリの場合には、X0 から X255 (入力点数が 256 の場合) までがあり、クライアント側は、この番地を指定して対応する接点に接続された入力機器の状態を取得する。ところが、メモリの種類によっては、数字で表される番号 (上記の場合には 0 から 255) が規則性を持たない場合がある。そのため、メモリを通して入出力機器の状態を確認する場合には、入出力メモリオブジェクト内で対応する番地情報を検索するため入出力点数に比例して応答時間が増加する。それに対して、入出力機器オブジェクトに対する状態の取得または変更メソッドを発行した場合には、3.3 節で述べた IOR 内のオブジェクトキーの採番方式によって、入出力点数に依存しない処理を実現できる。

また、入出力機器の状態の取得と変更で応答性能が異なるのは、それぞれに対応するメソッドにおけるパラメータが異なるために、ネットワーク経由でのパラメータの引き渡しの際の処理が異なることが原因である。

5. おわりに

オブジェクト指向を用いた分散生産ライン管理・保守システムを提案し、評価を行った。

生産ラインの構成機器である PLC について、生産状況の監視と故障時の保守の両面に対応可能なオブジェクトモデル案を提案した。具体的には、PLC を、機器の保守・管理の側面から定義が必要なハードウェア関連クラスと、生産状況管理に必要なメモリ関連クラスを定義した。さらに、入出力メモリクラスとは独立に、入出力機器クラスを設計したことで、ネットワークに直接接続され自律動作するような機器にも対応できるモデルとなっている。

さらに、PLC のように CPU 性能やメモリ性能が必

ずしも十分でない制御機器に対する分散オブジェクト管理機能の実現方式を提案し、その試作を行った。

試作した分散オブジェクト管理機能上に提案した PLC オブジェクトを実装し、メモリ性能と応答性能の観点からシステム性能評価を行った結果、メモリ性能上でも応答性能上でも管理・保守に必要な機能を実現できることが分かった。

今後は、応用オブジェクト作成に際して開発効率を向上させるために必要な IDL コンパイラの開発や、複数の保守・管理クライアントからの要求を多重処理するためのマルチスレッド化開発を行う予定である。

参考文献

- 1) Rockwell International Corporation: Perspectives on the Future of Automation Control - A White Paper (1997).
<http://www.ab.com/events/choices/index.html>.
- 2) Wozny, M.J. and Regli, W.C. (guest editors): Computer Science in manufacturing, *Comm. ACM*, Vol.39, No.2 (1996).
- 3) Computer Integrated Manufacturing (CIM) Application Framework, Specification 1.3, SEMATECH Technology Transfer #93061697F-ENG (1996).
- 4) Whiteside, R., Pancerella, C.M. and Klevgard, P.: A CORBA-Based Manufacturing Environment, *Proc. Hawaii International Conference on System Sciences* (1997).
<http://nittany.ca.sandia.gov:8001/DistObjMan.html>.
- 5) 松家英雄: 機械工場におけるオープン化, 精密工学会誌, Vol.63, No.5, pp.633-638 (1997).
- 6) Lee, B.H.: Agent-based Remote Diagnostics of Product Populations Across the Full Product Life Cycle: An Industrial Multi-Agent System Approach In An Electronic Commerce Framework, *Proc. 7th International Workshop on Principles of Diagnosis, Technical Paper (Poster Session)* (1996).
<http://diagnostics.stanford.edu/bhl/pubs/DX96Abstract.html>.
- 7) Blomseth, R., Capolongo, W., Dolin, B. and Lund, J.: The LONWORKS Network Services (LNS) Architecture Technical Overview, Technical Report, Echelon Corporation (1997).
http://www.echelon.com/LonWorks_Networks/wpapers.htm.
- 8) J. ランボーほか (著), 羽生田栄一 (監訳): オブジェクト指向方法論 OMT—モデル化と設計, トッパン (1992).
- 9) Object Management Group: The Common

Object Request Broker: Architecture and Specification, Technical Report, Revision 2.0 (1995).

- 10) 上野雅敏: FA/PA 監視制御用ソフト「FIX」, オートメーション, Vol.41, No.7, pp.13-18 (1996).

(平成 9 年 11 月 27 日受付)

(平成 11 年 1 月 8 日採録)



楠 和浩 (正会員)

1962 年生. 1986 年九州大学工学部情報工学科卒業. 1988 年同大学大学院工学研究科情報工学専攻修士課程修了. 同年三菱電機(株)入社. 現在同社情報技術総合研究所に勤務.

1997 年より在職のまま九州大学大学院システム情報科学研究科博士課程に在学. 情報ネットワークおよび分散処理システム構築に関する研究・開発に従事. 1992 年情報処理学会第 45 回全国大会奨励賞受賞. 電子情報通信学会会員.



今井 功 (正会員)

1965 年生. 1988 年東海大学工学部電子工学科卒業. 1990 年同大学大学院工学研究科電気工学専攻修士課程修了. 同年三菱電機(株)入社. 現在同社情報技術総合研究所に勤務.

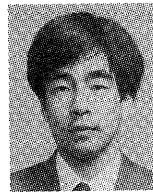
分散システム構築基盤技術の研究・開発に従事.



大谷 治之 (正会員)

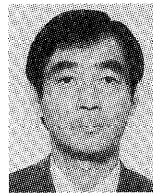
1990 年早稲田大学教育学部理学科数学専修卒業. 同年三菱電機(株)入社. 現在同社情報技術総合研究所に勤務. ユーザインタフェース, 分散システム, トランザクション処理,

オブジェクト指向技術の研究・開発に従事.



中川路哲男 (正会員)

1958 年生. 1981 年東京大学電気工学科卒業. 1983 年同大学大学院工学研究科電気工学専攻修士課程修了. 同年三菱電機(株)入社. 現在同社情報技術総合研究所に勤務. 工学博士. 情報セキュリティに関する研究・開発に従事. 1988 年 9 月情報処理学会学術奨励賞受賞. 電子情報通信学会会員.



大島 道隆

1947 年生. 1971 年東京大学工学部産業機械工学科卒業. 1973 年同大学大学院工学研究科産業機械工学専攻修士課程修了. 同年三菱電機(株)入社. 1980 年ミシガン大学計算機工学科修士課程修了. 現在同社名古屋製作所ロボットシステム部長. 制御機器のオブジェクトモデル化, 工業用ロボットにおけるプログラミング開発環境用 CCV アプリケーションの研究開発に従事. IEEE 会員.



牛島 和夫 (正会員)

1937 年生. 1961 年東京大学工学部応用物理学科(数理工学コース)卒業. 1963 年同大学大学院修士課程修了. 同年九州大学中央計数施設勤務. 1977 年九州大学工学部情報工学科教授(計算機ソフトウェア講座担当), 1996 年同大学大学院システム情報科学研究科教授, 現在に至る. 1990 年 4 月から 1994 年 3 月まで九州大学大型計算機センター長, 1996 年から九州大学大学院システム情報科学研究科長を兼務. 1991 年度情報処理学会九州支部長. 1995/96 年度本学会監事. 工学博士. 電子情報通信学会, ソフトウェア科学会, ACM, IEEE-CS 各会員.