情報制御システムにおける制御系保護を狙いにした異種データの マルチモード・フィルタリング方式

正員 河野克己 ((株)日立製作所システム開発研究所)
 非会員 鮫島茂稔 ((株)日立製作所システム開発研究所)
 非会員 足達芳昭 ((株)日立製作所情報制御システム事業部)
 正員 薦田憲久 (大阪大学工学研究科)

A multi-mode filtering technique to assure real-time data sharing in information and control system

Katsumi Kawano member, Shigetoshi Sameshima non member, Yoshiaki Adachi, non member (Hitachi, ltd.),
 Norihisa Komoda, member (Osaka University)

We propose a multi-modal information filtering technique for attaining dependable and free access to heterogeneous data in decentralized information and control systems. It takes into consideration the difference in the quality of the information on control schedule, management, data for real-time control, and so on. The architecture extends the conventional model called data field (DF). In order to prevent the turbulence from an information system affecting a real-time control system, the developed architecture and technique enable to limit the multicasting range of a message respectively on DF, Multi-Cast Group (MCG) and Mode, and a DF can provide plurality of network segments as the information sharing media. Moreover, It is shown that the information filtering technique can also treat the conventional online test method systematically. The developed architecture has been applied to various real industrial production systems and its effectiveness been verified.

キーワード: 情報制御システム、自律分散システム、情報フィルタリング、オンライン拡張

Keywords: information and control system, autonomous decentralized system, information filtering, on-line expansion

1. 緒言

交通や産業分野の情報制御システムでは、ユーザーニーズの変化やネットワーク技術の発展に伴い制御系のオープン化、分散化や情報系の対象範囲拡大が進行している。変化するユーザーニーズに即応するためには、運用計画や実績管理を司る情報系と機器やプロセスの制御系との緊密な連携が必要であり、また変化に応じてシステムを段階的に既設部を停止することなく拡張する必要がある。情報系と制御系の緊密な連携や無停止での拡張には、計画、管理、保守、制御データといったシステム内に分散して存在する複数種類のデータの共有を、如何に制御系のリアルタイム性を損なうことなく行うかが一つの課題となっている。

これまで、システムの信頼性、拡張性、保守性向上を狙いにした自律分散概念の提案[1]と、この概念に基づきシステムを構成するサブシステム間でのネットワークを介したデータ共有を同報通信を応用して行う方式が提

案されている。交通や産業分野の情報制御システムでの実用化[2]-[5]や、その通信プロトコルの業界標準化[6][7]の報告もある。これらの中で、オンラインでのテスト支援を目的とした従来方式[8]-[10]では、サブシステムごとにオンラインとテストという二つのモードを持たせ、このモードに応じてサブシステムが受信するデータの透過性を制御するものである。これは、サブシステムをオンライン系とテスト系の二つのグループに分けグループ間のデータ共有をテスト系がオンライン系へ悪影響を与えぬよう部分的に制約した方式であるが、グループ数が増えた場合についての考慮はされていない。一方、他の従来方式[11]ではデータごとにその同報通信する範囲をサブシステムのグループに限定する方式が提案されている。これは、各サブシステムでの同報データの受信処理負荷を減らす為の方式であるが、グループ間でのデータ共有については考慮されていない。

本論では、上記従来方式を拡張して二つ以上の複数モードを持つ複数のグループに対するフィルタリング則を導出し、情報制御システム内の質の異なるデータを

グループ間でフィルタリングする方式を提案する。更に提案方式の実装方式を示し、これまで個別に議論されているモードとグループを組み合わせることによってフィルタリングがより確実に行えることを示す。最後に提案方式をFA(Factory Automation)システムに適用した例を用いてその有効性を示す。

2. 自律分散システム

<2.1 放送型通信>

システムの拡張性向上等を狙いに同報通信を応用した放送型通信[7]と呼ばれる方式を用いた自律分散システムが提案されている[2]。従来の自律分散システムアーキテクチャは、図1に示すように、自律したサブシステム、各サブシステムを管理するシステムオブジェクト、アプリケーション処理を行うアプリケーションオブジェクト、サブシステム間のデータ授受を実現するデータフィールド(DF)、からなるモデルで定義出来る。尚、図1のような自律分散制御システムでは、各サブシステムは各々自らの制御範囲を持ち、自らの対象範囲との制御入出力を行う。

このシステムでのアプリケーションオブジェクト間のデータ授受の流れを以下となる。

- (1) 送信: システムオブジェクトは、アプリケーションオブジェクトが出力するデータ(図1では data1)を PDU (Protocol Data Unit) に、また該データの識別子である TCD(Transaction CoDe)の値(図1では TCD1)を PH (Protocol Header) に、各々設定し DF を介して該メッセージを各サブシステムへ同報通信する。
- (2) 受信: 各サブシステムのシステムオブジェクトは DF からメッセージを受け取り、該メッセージ PH の TCD の値から自サブシステムで必要なデータと判断した場合にはその PDU のデータをアプリケーションオブジェクトに渡す。

<2.2 放送型通信範囲の限定>

拡張時のサブシステムのテストやサブシステム間の効率の良いデータ授受を狙いに、前節で示した放送型通信の通信範囲を限定する以下の方式が提案されている。

(1) オンラインテスト [8], [9], [10]

本方式は、サブシステム拡張時の事前テストとして、既に稼動しているサブシステムが出力するデータを DF を介して共有し、拡張サブシステムのアプリケーションオブジェクトのテストをすることを目的にしている。図1に示す如く、システムオブジェクトにオンラインとテストという2種類のサブシステムのモードを管理させる。システムオブジェクトは、メッセージを DF に送る際に自サブシステムのモード値を PH のテストフラグ(test flag)に設定する。メッセージの受信は表1のルールによって行われる。即ち、システムオブジェクトはテスト中サブシステムの出力データとオンライン稼動中のサブシステムのデータを test flag により識別し、自サブシステムのモードが、①オンラインの場合はオ

ンラインと設定(test flag=0)されているメッセージのみを受け取り、②テストの場合はオンラインとテストのどちらの設定(test flag=0 or 1)であっても受け取る。

本方式を用いると、テストをするサブシステムでは稼動しているサブシステムからオンライン・リアルタイムでデータを受け取ることが出来、このデータを使つてのテストが可能となる。一方、システムオブジェクトはテスト中サブシステムの出力データがオンライン稼動中のサブシステムへ渡ることを抑止できる。

図1. 放送型通信とテストデータの抑止

Fig. 1. Broadcasting and suppression of test data

Table 1. On-line test rule

表1. オンラインテストルール

		受信メッセージのモード	
自サブシステムのモード		On-line	Test
	On-line	0	1
	Test	0	0

0: 受信
1: 抑止

(2) マルチキャストグループ [11]

図2に示すように一つの DF 内に複数のマルチキャストグループ(MCG: Multi-Cast Group)を構成させてデータ授受を行う他の従来方式が提案されている。DF を介して授受されるメッセージの PH には TCD, Test flag の他に MCG 番号が追加されている。TCD は一つの DF 内でユ

図2. マルチキャストグループ

Fig. 2. Multicast group

ニークに決められる。

この方式では、2.1節の(1)(2)で述べた同報通信の範囲が MCG の範囲内に限定され、アプリケーションオブジェクト間のデータ授受は各 MCG に属するサブシステム間で行われる。図 2 の例では二つのサブシステムが MCG1 に、別の一つのサブシステムが MCG2 に登録されており、MCG2 に属するサブシステムの送信メッセージは MCG1 に属するサブシステムでは受信できない。

<2.3 問題点>

(1) 統一システムモデル

従来の2方式は、各々別の目的に対して個別に提案されてきた。何れも同報通信範囲の限定を扱っているが両者の関係についての言及はない。以下の(2)(3)で述べるサブシステムの状態やデータの種別という二つの要因が組み合わされた場合についての考慮はない。従って、これらの方式は、情報系と制御系の緊密な連携に要求されている複数種類のデータを状況に応じてリアルタイムに共有する統一的手段として十分ではない。

(2) サブシステムの状態への対応

オンラインテスト方式は、サブシステムのオンライン状態またはテスト状態、の二つのモードに応じて授受するデータを分離したり部分的に透過にしたりするものである。しかし、考慮しているモードは 2 種類しかない。たとえ同じアプリケーションオブジェクトから成るサブシステムであっても、例えば緊急時、通常運用時、保守時といった様々な状態を取り得るが、これらへの対応は十分ではない。

(3) データの種別による対応

マルチキャストグループ方式は、複数のサブシステムグループ毎に共有するデータを分離する方式である。即ちデータの種別に応じて同報の範囲を限定する方式である。これは、例えば制御、監視あるいは計画といったアプリケーションごとに用いるデータのリアルタイム性の度合いに応じてグループ分けするものであるが、グループ間でデータを共有する場合についての考慮がなされていない。

3. データのフィルタリング

<3.1 考え方とシステムアーキテクチャ>

DF を介して共有するデータをサブシステム間で分離、あるいは部分的に透過にする機能をデータフィルタリングと呼ぶことにする。以下では、このデータフィルタリングという観点から拡張マルチキャストグループ XMG(eXtended Multicast Group)を提案しデータの分離と部分的な透過の考え方を示す。更に、この XMG を用いて DF が複数存在する場合も含めた全体システムモデルを示す。

(1) XMG

一つの DF は複数の XMG から成る。一つの XMG は、サブシステムの状態とデータの種別、の二つの要因の組

み合わせで定める。即ち図 3 に示すように、DF を介したデータ授受に必要なリアルタイム性の度合いを(サブシステムの状態、データの種別)の組み合わせで捉えて、同程度リアルタイム処理を必要とするデータをグループ化する。ここで、XMG の値は m 種類の状態 x と n 種類のデータ種別 y とを用いて XMG_{xy} で表している。この XMG は従来の Test flag を用いた方式を複数モードへと一般化し、更に従来の MCG をデータの種別に対応させて新たに定義したものである。XMG 間のデータは必要とするリアルタイム性がより高いグループから低いグループに対してのみ透過とし、逆向きに流れるデータは抑止する。

図3. 拡張マルチキャストグループとマルチモードフィルタリング

Fig. 3. Multi-mode filtering by extended multicast group

図4. 全体システムモデル

Fig. 4. Total system model

(2) 統一システムモデル

上記(1)の考え方に基づき、XMGを導入したときのシステムアーキテクチャを図4に示す。このアーキテクチャは、対象とするシステム全体が共有するデータの集合をドメイン Domain と名づけ、一つの Domain が複数の DF から成る場合も考慮している。ここで、一つの DF 内は、ユニークな値を持つ TCD により識別できるデータの集まりであり、

該データは複数の XMG により分離して管理する。DF が異なると、例えば TCD や XMG が同一であっても異なるデータとして扱う。異なる DF 間でのデータ共有は、図 4 に示すゲートウェイに相当するサブシステムのシステムオブジェクトが、ある DF のデータを別の DF に渡す際に DF、XMG、TCD 値の変換することにより実現する。

＜3.2 マルチモード・フィルタリング＞

(1) サブシステムのモード

上記の考えを用いて、従来のオンラインテスト方式を拡張し、 m 種類のモードに対する DF 内フィルタリング方式を提案する。従来の $\text{Test flag} = 0 \text{ or } 1$ に代わって m 種類のモードとしてサブシステムモード x とモード u を定義する。モード値は $0, 1, 2, \dots, m-1$ を取りうるものとし、システムオブジェクトが管理する。ここで m は自然数である。システムオブジェクトはメッセージを送信する際に自サブシステムのモード値をメッセージモード値に設定する。即ち u は x の値をサブシステムの送信メッセージに付加したものである。システムオブジェクトは以下の判断に基づきデータフィルタリングを行う。

ステップ1: もし自サブシステムの $x < \text{受信メッセージの } u$ であれば該メッセージは廃棄する。

ステップ2: もし $x \geq u$ であれば受け取る。

尚、受け取った後の TCD を用いたデータ授受は 2.1 節で示した手順と同様である。

サブシステムモード値、メッセージモード値とフィルタリング則の関係を、表1で示したオンラインテストルールに相当する Decision matrix を D で表すと、 D は次式で導かれる。

$$D = x^T * u. \quad (1)$$

ここで、記号 x, u は $x = [0, 1, \dots, m-1]$ 、 $u = [0, 1, \dots, m-1]$ のベクトル、 T は転置を表す。また記号* は以下の演算を表す。

$$\begin{aligned} \text{if } x_i < u_j, \text{ then } x_i * u_j &= 0, \\ \text{if } x_i \geq u_j, \text{ then } x_i * u_j &= 1 \end{aligned} \quad (2)$$

例として $m=3$ の場合を考えると

$$D = \begin{vmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{vmatrix}$$

となる。ここで $x = [0, 1]$ 、 $u = [0, 1]$ とし上記演算をすると、表1と等価となる D が導かれ、提案方式が従来のオンラインテスト方式を拡張し一般化したものであることが分かる。

(2) データのモード

各 XMG の n 種類のデータ種別を示す値を $y, y = 0, 1, 2, \dots, n-1$ とする。XMG の値は上記 x に加えて y との組み合わせで MCG_{xy} で表すこととし、メッセージの PH に書かれた y の値、即ちメッセージのデータの種別を v とすると、上記(1)と同様に以下のフィルタリング則が定められる。

ステップ1: もし自サブシステムの $y < \text{受信メッセージの } v$ 、であれば該メッセージは廃棄する。

ステップ2: もし $y \geq v$ 、であれば受け取る。

マルチモード・フィルタリングは、以上(1)(2)で各々示したステップ1とステップ2により行う。

図5に3種類のモード値 x が各々 i, j, k ($i < j < k$) であり、メッセージの種類 y が p, q ($p < q$) である MCG が存在する場合のデータの流れを示す。グループの中で最もリアルタイム性処理が要求される XMGip で発生したデータ、即ち XMGip に属するサブシステムが出力したデータは他の全ての XMG で透過的に流れ、各 XMG に属するサブシステムで共有することが出来る。逆に XMGip へは他の XMG で発生したデータは全て抑止される。XMGjp と XMGiq へは XMGip で発生したデータのみ透過的に流れ、XMGkp へは XMGip と XMGjp の二つの XMG で発生したデータが流されている。

図5. マルチモード フィルタリング例

Fig. 5. An example of multi-mode filtering

＜3.3 ミドルソフトウェアの実装＞

3.2 節で示した方式を LAN(Local Area Network)に接続された計算機システムを対象にした実装例を示す。一つの DF は一つまたは複数のネットワークセグメントに対応させ、サブシステムは計算機に対応させる。DF 内での同報通信には UDP/IP(User Data gram Protocol /Internet Protocol)を用い、一つの XMG は一つの UDP ポートに対応させる。即ち、同報通信は OSI7層のネットワーク層以下の層で行わせる。提案方式の処理を行うシステムオブジェクトは、その上位層のミドルソフト(以下 NX と呼ぶ[12])として実装する。

(1) API (Application Program Interface)と通信処理

アプリケーションプログラムがDFを介してデータの授受を行うために必要なミドルソフトとの間の API と API ごとのミドルソフトでの通信処理を以下に示す。

① attach(DF, XMG, TCD); 自サブシステムが属する DF, XMG と受信するデータの TCD を登録する。NX は DF, XMG, TCD ならびにどのアプリケーションオブジェクトが該 TCD のデータを必要とするか、を自らの管理テーブルに格納する。

② put(TCD, data); データを該データに対応する TCD を付加して送信する。NX は PH に該当する DF, XMG, TCD の値を設定し、data を PDU に設定して、自サブシステムが属する XMG に対応する UDP ポートから同報通信する。

③ get(TCD, data); 当該 TCD に対応するデータを受信する。NX は受信したメッセージの XMG 値から 3.2 節のステップ 1~4 の処理により該メッセージのフィルタリングを行い、廃棄しなかったメッセージの TCD と自管理テーブルとから判断して該当するアプリケーションオブジェクトにデータを渡す。

(2) 性能解析

本実装方式によれば、各サブシステムで受け取る必要のないデータネットワーク層で廃棄することが出来、以下の解析に示すようにミドルソフトの処理負荷を減らすことが出来る。

解析のための次のパラメータを定義する。

X: LAN 上の単位時間あたりの全体トラフィック量、

χ : フィルタリング後のトラフィック量、

α : ネットワーク層での 1 トラフィック受信処理時間、

β : ミドルソフト NX での 1 トラフィック受信処理時間、

γ : ネットワーク層でのフィルタリング(廃棄)処理時間。

この時、サブシステムが LAN からメッセージを受信してアプリケーションオブジェクトにデータを渡すまでの単位時間あたりの処理負荷は、

フィルタリングを行わない場合: $Y(X) = (\alpha + \beta) \cdot X$ 、

フィルタリングした場合: $\phi(X) = \gamma \cdot (X - \chi) + (\alpha + \beta) \cdot \chi$

となる。 $\chi \leq X$ であり $\chi = X$ の時、 $\phi = Y$ となる。ネットワーク層での UDP パケットの廃棄処理 γ は α, β に比べ一般に十分小さく、また例えば LAN ボード上でハード処理する場合には $\gamma \ll 0$ である。従って、 $\phi(X) = \phi \approx (\alpha + \beta) \cdot \chi$ (一定) となり、 $\phi \leq Y(X)$ となる。以上より、提案方式の処理負荷は全体トラフィック増大の影響をほとんど受けることがなく一定に保つことが出来、従来方式より少なくすることが出来る。即ち、ある XMG 以外のデータが増大しても当該 XMG でのデータ授受のリアルタイム性保証が容易になる。

また、本実装方式は受信バッファを UDP ポートごとに確保出来るため、同報通信により受け取ったデータを自サブシステム内でその種別ごとに分離して管理することが可能である。従って、各 XMG に対応するバッファ処理が相互に悪影響を与えないソフト構造とすることが出来、ある XMG のトラフィックの擾乱に対しても他の XMG でのデータ授受の信頼性を保証する実装方式となる。

4. 適用事例

<4.1 対象システム>

FA(Factory Automation)分野を対象に提案方式を適用した例を示す。本分野では、需要変動即応のため情

報系と制御系との緊密な連携が強く求められている。時々即ち、刻々変化する計画や実績、保守、制御等に関わる複数種類のデータを各サブシステムが自らの状況に応じてオンラインでリアルタイムに共有できるシステムが必要となっている。図 6 は適用した対象システムの構成である。このシステムは、大きく、①生産計画と製品レシピを管理する工程管理サーバー、②上流工程から流れてくる部品を検知する位置検知用コントローラ、③流れてきた部品の加工等を行う設備制御用コントローラ、④工程管理サーバーとコントローラ間を繋ぐネットワーク(制御系 LAN)、ならびに上位ホスト計算機や上流工程サーバー(図示せず)との間を繋ぐネットワーク(情報系 LAN)、⑤ネットワーク間を接続するゲートウェイ、とから構成される。

工程管理サーバーはゲートウェイを介して上位ホスト計算機等から入手する生産計画、レシピ情報を管理する。部品のトラッキングデータは位置検知用コントローラから、また需要変動に伴う生産計画の変更データや対応するレシピデータは工程管理サーバーから、各々設備制御用コントローラに送られる。設備制御用コントローラはこれらのデータを用いて制御を行う。

需要の変動は、生産計画の変更は勿論、製造ラインの増設や改造を伴う場合が多い。このため、このシステムで扱うデータには①製造ラインで定常的に、かつ高頻度で発生する制御データ、の他に②上位ホスト計算機等から流される生産計画変更に関するデータ、ならびに③ライン増改造時の発生する保守の為のデータ、が存在する。求められている需要変動即応のためにはこれらデータをサブシステム間でタイムリーに共有する必要がある。

図6. 対象システム構成

Fig. 6. Applied system configuration

<4.2 提案方式の適用>

第3章で述べた提案方式を図6の生産システムへ適用した例を以下に示す。サーバー、コントローラ群ならびにゲートウェイを各々サブシステムに対応させ各々に NX を実装し、また一つの LAN ネットワークセグメントを DF に対応させ UDP/IP プロトコルを実装させている。DF は、上位ホスト計算機等とゲートウェイとのデータ授受を行う為の

情報系 LAN に対応する DF2、サーバーとコントローラ間の制御系 LAN に対応する DF1、とから成る。DF1 には 4.1 節でも述べたように三つの大きなデータの流があり、各々 XMG00, XMG01, XMG11(図示せず)に対応させる。XMG00 は定常的な制御処理を行うためのトラッキングデータ、XMG01 は当該制御処理に必要な生産計画データ、製品レシピデータや制御結果の実績データ、XMG11 は保守診断処理用のテストデータ、である。XMG00 のデータはデータ量は小さいが周期的に発生し高いリアルタイム処理が要求される。XMG01 のデータ発生は過渡的であり授受のリアルタイム性はそれほど必要ないが、データ量が大きくはデータ授受を行う際の伝送負荷への影響が大きい。また、XMG11 のデータはテスト用であり、他の XMG に属するサブシステムへ渡ることのないようにせねばならない。

ここで、3.2 節で示したマルチモード・フィルタリング則の適用すると、設備制御用コントローラはトラッキングデータと生産・レシピデータを受け取ることが出来る。この時、このシステムで最もリアルタイム性が要求されるトラッキングデータの同報配信性能が他のデータの送受信によって劣化することは従来方式より少なく出来る。また、保守業務を行うサブシステムは XMG00, XMG01 で実際に利用しているデータをオンライン、リアルタイムで受け取り生産現場実環境でのテストなどに用いることが出来る。この時、XMG11 を介してのデータ授受が XMG00, XMG01 に与える影響は少ない。

<4.3 性能評価結果>

4.2 節で示した適用事例の性能評価結果を図 6 に示す。図 7 の縦軸は、図 6 のシステム構成にて XMG11 を流れるトラフィックを増大させた時の、XMG00 でのコントローラ間のデータ授受(スループット)の平均時間である。実線が提案方式を適用した場合、破線はフィルタリングをしない従来方式である。提案方式は全体のトラフィック量(横軸)が増大した時のスループットの劣化を従来方式に比

図7. データ伝送性能の評価結果

Fig. 7. An evaluation result of data transmission performance

べ 20%程度にすることが出来た。図 7 の提案方式における劣化部分はほとんど LAN 上の伝送遅れの増大によると考えられる。

5. 結言

情報制御システムを対象に、制御系に擾乱を与えることなく、情報系との連携や既設システム稼動状態での拡張保守、を容易にするマルチモード・フィルタリング方式を提案した。提案方式は従来の自律分散システムで個別に議論されていたオンラインテスト方式と MCG 方式を拡張し統一したものであり、データの種別とサブシステムの状態とから判断して部分的なデータ透過性制御を行うものである。また提案方式のミドルソフトウェアとしての実装し、自サブシステム内で異なる種類のデータをより確実に分離する方式を示した。更に FA システムに適用した例を用いてその有効性を示した。

提案方式は、既に製品化され多くの実システムに適用されている[12]。

(平成 13 年 3 月 23 日受付、平成 13 年 8 月 21 日再受付)

文献

- [1] 森, 他: “自律分散概念の提案”, 電学論 C, 104, 12, 303, 1984
- [2] K. Mori: “Autonomous Decentralized System: Concept Data Field Architecture and Future Trends”, Proc. of ISADS'93, pp. 28-34, 1993.
- [3] 河野: “自律分散システム”, 情報処理, Vol. 36, No. 11, pp.1054-1061, 1995.
- [4] F. Kitahara, et al.: Widely Distributed Train Traffic Computer Control System and its Step-by-step Construction, Proc. of ISADS 95, pp. 93-102, 1995.
- [5] M. Oku, et al.: “Hi-Cell Architecture and the project Model for Manufacturing ADS”, Proc. of ISADS'93, pp. 398-405, 1993.
- [6] S. Shin: “Panel Discussion on Open ADS”, Proc. of ISADS 95, 1999.
- [7] 新: “ADS-net と国際標準化活動”, 計測と制御, Vol.39, NO. 3, pp. 209-215, 2000.
- [8] K. Kawano, et al.: “Autonomous Decentralized Test Technique”, Proc. of COMPSAC 89, pp. 52-57, 1989.
- [9] H. Yamamoto and K. Kawano: “On-line Software Test Techniques Based on Autonomous Decentralized System Concept”, Proc. of Workshop on Future Trends of Distributed Computing Systems, pp. 291-296, 1993.
- [10] E. Nishijima and K. Kawano: “On-line Testing for Application Software of Widely Distributed System”,

Proc. of 15th Symposium on Reliable Distributed Systems, pp. 54-63, 1996

[11] H. Wataya and K. Kawano: "The Cooperating Autonomous Decentralized System Architecture", Proc. of ISADS 95, pp. 40-47, 1995.

[12] "特集 産業社会のグローバル化と変革に対応する新情報制御システム", 日立評論, vol. 78, 1996

河野 克己 (正員)

1980 年早稲田大学理工学研究科電気工学専攻修士課程修了。同年(株)日立製作所入社。以後、同社システム開発研究所にて産業、交通、電力分野などの情報制御システムを対象にした自律分散システムの研究開発に従事。計測自動制御学会 98 年度技術賞。IEEE などの会員。

鮫嶋 茂稔 (非会員)

1993 年東京大学大学院工学系研究科修士課程修了。同年(株)日立製作所入社。同社システム開発研究所にて、自律分散システムを始め、分散制御システムのシステムアーキテクチャ、ミドルソフト開発に従事。計測自動制御学会会員。

足達 芳昭 (非会員)

1978 年愛媛大学大学院工学研究科修士課程終了。同年、(株)日立製作所に入社。現在情報制御事業部において、主に制御用ネットワーク、ネットワークミドルウェア、自律分散システム等の開発に従事。技術士(情報工学部門)

薦田 憲久 (正員)

1974 年大阪大学大学院工学研究科電気工学専攻修士課程修了。同年(株)日立製作所入社。81-82 年UCLA留学。91 年4月大阪大学工学部情報システム工学科助教授、現在、同教授。工博。情報システム工学、知識情報処理の研究に従事。計測自動制御学会 86 年度論文賞、同学会 87 年度技術賞、電気学会 98 年度論文賞、同学会 2000 年度進歩賞受賞。IEEE などの会員。