

時間駆動ネットワークを導入した 組み込み制御システム向け分散オブジェクト環境

唐澤 陽平 石郷岡 祐 横山 孝典 兪 明連 志田 晃一郎
武蔵工業大学

1. はじめに

組み込み制御システムは、自動車制御や FA 等で使用されており、その多くはハードリアルタイムシステムである。リアルタイムシステムの構成法には、イベントに応じて処理を実行するイベント駆動アーキテクチャと、周期的に処理を実行する時間駆動アーキテクチャがある [1]。ハードリアルタイムシステムには、デジタル制御が一定のサンプル周期による処理を基本としていることや、最悪の反応時間を予測しやすく、時間保証が容易なため、周期的に処理を実行する時間駆動が望ましい。

近年、リアルタイムシステムの分散化により、CORBA [2] などの分散オブジェクトがリアルタイムシステム分野へ適用されつつある。CORBA のような分散オブジェクトを用いることで、イベント駆動アーキテクチャに基づく分散システムを容易に実現できる。しかし、CORBA は遠隔手続き呼び出しに基づくメッセージパッシングを採用しており、ネストされたオブジェクト間通信による遅延やジッタを避けることは難しく、時間駆動アーキテクチャには適していない。

我々は、時間駆動に基づいた分散処理を行うために、CAN(Controller Area Network) に対応した時間駆動分散オブジェクト環境 [3] を開発したが、自動車における X-by-Wire 制御のように、よりジッタの少ないシステムを実現するには、FlexRay 等の時間駆動ネットワークの導入が必要である。そこで本研究の目的は、時間駆動ネットワークにも対応させた分散オブジェクト環境を開発することである。

2. 時間駆動分散オブジェクトアーキテクチャ

我々が提案する時間駆動分散オブジェクトアーキテクチャを図 1 に示す。

コンピュータ A, B にはそれぞれオブジェクト A, B が配置されている。また、位置透過性を実現するために、オブジェクト A のコピーであるレプリカ A をコンピュータ B に配置する。オブジェクト B のメソッド update() の計算にオブジェクト A の属性値が必要な時は、オブジェクト B のメソッド get() を呼び出し、レプリカ A から属性値を得ることで分散処理を実現する。また、オブ

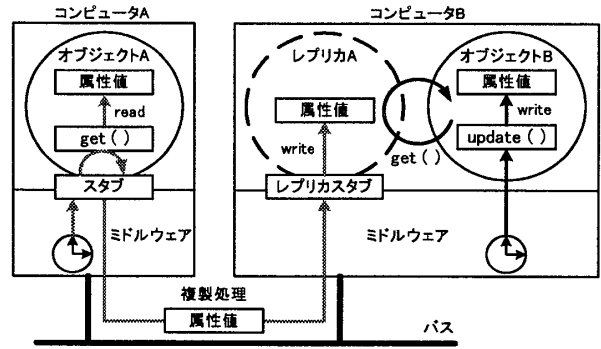


図 1 時間駆動分散オブジェクトアーキテクチャ

ジェクト本体とレプリカの属性値の一貫性を保つために、複製処理を周期的に実行する。オブジェクト側にスタブ、レプリカ側にレプリカスタブを配置する。スタブはオブジェクト A のメソッド get() を呼び出して属性値を取得し、送信メッセージにまとめる機能、レプリカスタブは受信メッセージから属性値を取得して、レプリカ A のメソッド set() を呼び出して属性値を格納する機能を提供する。

オブジェクト B のメソッド get() の処理はネットワーク通信を含まず、ローカルオブジェクトのメソッド呼び出しと同じため、実行時間の予測が容易となる。よって、時間駆動分散オブジェクトアーキテクチャは、ハードリアルタイムシステムに適した分散処理を実現することができる。

3. 分散オブジェクト環境

本研究で提案する分散オブジェクト環境を図 2 に示す。ECU(Electronic Control Unit) とは、自動車向けの組み込みコンピュータのことである。ミドルウェアには、複

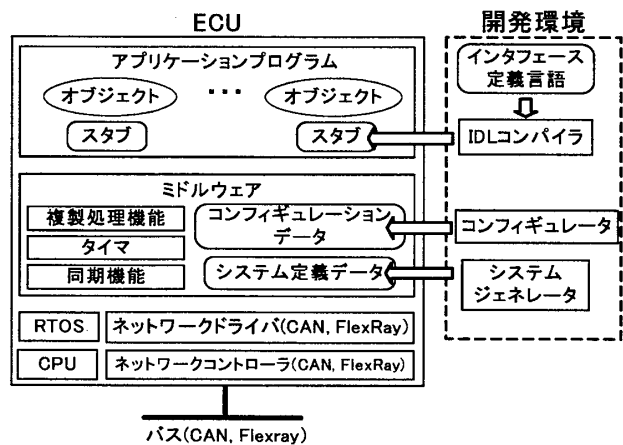


図 2 分散オブジェクト環境

A Distributed Object Computing Environment for Embedded Control Systems with Time-Triggered Networks
Yohei Karasawa, Tasuku Ishigooka, Takanori Yokoyama, Yoo Myungryun, Koichiro Shida

製処理機能, タイマ, 同期実行機能がある。また, システムに依存する部分はその依存部の内容に応じたデータとする。ミドルウェアは各データを参照し処理を実行する。データは開発環境からそれぞれ生成される。

4. ミドルウェア

ミドルウェアは以下の機能を持つ。

- (1) 複製処理機能: オブジェクトの属性をレプリカへ周期的にコピーする。
- (2) タイマ: 異なる周期を混在させることができる。
- (3) 同期機能: 各 ECU の処理の同期を合わせる。

各機能は次のように動作する。(1) は, 各スタブを呼び出し属性値をネットワークへ送信する。そして受信後, レプリカスタブによりレプリカへコピーされる。これは, 一定の周期で行われる。(2) は, RTOS の周期ハンドラ, ハードウェアタイマ及び FlexRay のタイマがミドルウェアのタイマをカウントし, 複製処理や属性値の更新に周期的に実行要求を出す。(3) は, タイマからの周期的な実行要求により, ネットワーク上に信号を流し, 各 ECU のタイマを合わせる。同期機能は CAN を使用した場合のみ実行される。FlexRay の場合はそれ自身が同期機能を有している。

5. 開発環境

5.1 IDL コンパイラ

アプリケーションに依存する部分は, IDL(Interface Definition Language)[2] により記述されたオブジェクトのインタフェース定義言語から, IDL コンパイラによって自動生成する。インタフェース定義には複製処理で送受信される属性のみ宣言する。レプリカの構造体, レプリカのインスタンス宣言, スタブ, スタブに記述されるメソッド get(), レプリカスタブ, レプリカスタブに記述されるメソッド set() 等が生成される。

5.2 コンフィギュレータ

GUI を用いてシステム構成に依存する情報をコンフィギュレータに入力することで, コンフィギュレーションデータが生成される。入力する情報は, システム全体の ECU の数, 同期信号の周期, 複製処理の周期及びデータサイズ, オブジェクトとレプリカの配置情報等である。生成されるコンフィギュレーションデータには, 入力したパラメータ等が含まれる。

5.3 システムジェネレータ

GUI を用いてシステム環境に依存する情報をシステムジェネレータに入力することで, システム定義データが生成される。入力する情報は, ECU で使用するリアルタイム OS, 使用するリアルタイムネットワーク等である。生成されるシステム定義データには, 使用する OS の宣言, 選択したネットワークの宣言などの宣言定義が含まれる。

6. 評価

ミドルウェアの性能を評価するために, 表 1 に示す実験環境で, 送信複製処理と受信複製処理の時間を測定した。その結果を表 2 に示す。表中の時間の単位はすべて μ sec である。

表 1 実験環境

番号	ECU	OS	ネットワーク
環境 1	H8S/2638(20MHz)	μ ITRON	CAN
環境 2	V850/IA1(50MHz)	なし	FlexRay

表 2 複製処理の測定結果 (単位: μ sec)

複製処理の種類	平均時間	最大時間	最小時間	ジッタ (差)
環境 1 送信	380.8	384.0	380.8	3.2
環境 1 受信	419.2	422.4	419.2	3.2
環境 2 送信	43.20	43.28	43.20	0.08
環境 2 受信	47.04	47.04	47.04	0.00

環境 2 の受信複製処理のジッタが 0.00μ sec となっているが, この環境におけるハードウェアタイマの 1 カウントが 0.08μ sec 以下であるため, ジッタが測定に現れなかったためと考察する。

環境 1 のジッタは CAN コントローラによって発生するジッタと比較して非常に小さいため, CAN を使用するパワートレイン制御に適応可能であると考えられる。環境 2 のジッタは 0.08μ sec 以下であり, CPU のクロック数が 50MHz であるため 4 命令に相当する。この程度のジッタなら, FlexRay を使用する X-by-Wire 制御に適応可能と考える。

7. おわりに

CAN 及び FlexRay に対応したミドルウェア及び, 開発環境を開発した。これにより, 分散化した組み込み制御システムを容易に開発することができる。

今後の計画として, 時間駆動とイベント駆動が混在したシステムに対応可能なミドルウェア及び, 開発環境の開発を考えている。

参考文献

- 1) H.Kopetz, Should Responsive Systems be Event-Triggered or Time-Triggered? IEICE Trans. Inf. & Syst., Vol.E76-D, No.11, pp.1325-1332, 1993.
- 2) Object Management Group, Common Object Request Broker Architecture: Core Specification, Version 3.0.3, 2004
- 3) 石郷岡祐, 横山孝典: 組み込み制御システム向け時間駆動分散オブジェクト環境, 情報処理学会論文誌, Vol.48, NO.9, pp. 2936-2945, 2007.