C-022

アスペクト指向による時間駆動分散処理の実現

An Aspect-Oriented Implementation of Time-Triggered Distributed Objects

齋藤　亘弘†　　横山　孝典†　　兪　明連†

Nobuhiro Saito　Takanori Yokoyama　Myungryun Yoo

1. はじめに

近年，自動車や産業システムなど，組み込み制御システムを内蔵した機器が増大している．要求される機能も年々増加し，複数の組み込みコンピュータをネットワークで接続した分散型の組み込み制御システムも増えている．

分散型組み込みシステムでは，異なるコンピュータ上のソフトウェアが互いに連携しながら処理を実行する．分散システムの開発には，分散オブジェクト技術が広く利用されている．しかし，分散処理を意識せずに開発したソフトウェアを分散オブジェクト環境に対応させるには，もとのソースコードを書き換える必要がある．

ソースコードを直接書き換えずに，分散処理を実現する手法として Gal らの研究がある[1]．この研究では，アスペクト指向プログラミングを用い，CORBA によるリモートメソッド呼び出し処理をアスペクトで記述する．これにより，分散処理を意識せずに，開発したプログラムを書き換えることなく分散化を実現できる．しかし，CORBA による分散処理環境を対象しており，従来と同じミドルウェアを必要とする．

本研究では，組み込み制御システムを対象に，ミドルウェアを用いることなく，アスペクト指向プログラミングにより分散処理を実現する手法を提案する．具体的には，ハードリアルタイムシステムに適した時間駆動分散オブジェクト[2] を対象とに，その分散処理機能をアスペクトで実現する．これによりミドルウェアの開発や移植の工数も減らすことができる．

2. 時間駆動分散オブジェクト

時間駆動分散オブジェクトは自動車等の組み込み制御システムを対象に提案されたもので，周期的に処理を行うオブジェクトの集合でシステムを構成する異なるノード間でやり取りを行う場合は，レプリカオブジェクトを設けることでリアルタイム性のある分散処理を可能とする．

時間駆動分散オブジェクトの構成を図 1 に示す．各オブジェクトは，周期的に update メソッドを実行して処理を行う．update メソッドは必要により他のオブジェクトの get メソッドを呼び出してその値を取得し，自身の値を算出し，属性 value に記憶する． この例では，オブジェクト B はオブジェクト A の値を参照して，自身の値を更新する．位置透過性を実現するため，オブジェクト B があるノード 2 にオブジェクト A のコピーであるレプリカオブジェクトを配置する．ミドルウェアにより周期的にオリジナルオブジェクト A の値をレプリカオブジェクトにコピーすることで，

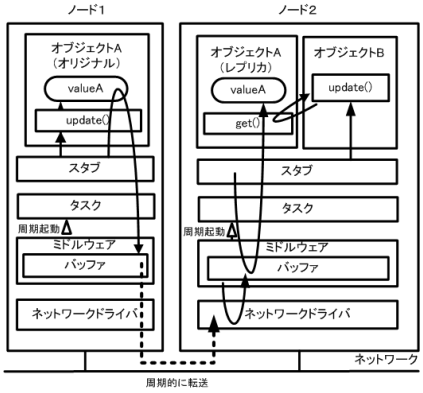

図 1　時間駆動分散オブジェクトアーキテクチャ

複製処理を実現する．オブジェクト B は update をネットワークを介さず，自身のノード上にあるレプリカオブジェクトの get メソッドを呼び出すことで，オブジェクト A の値を参照できる．

時間駆動分散オブジェクトミドルウェアはオリジナルオブジェクトのデータを周期的に送信を行う送信機能，ネットワークドライバからデータを取り出し，バッファへ格納する受信機能がある．アプリケーション中のオブジェクトとミドルウェアはスタブを介して接続される．また，スタブは同一周期のオリジナルオブジェクトが複数ある場合，それらのデータを一つのパケットにまとめて送信するためのパック処理を行なう．受信側のスタブは，パック処理されたデータから必要なデータを取り出すアンパック処理を行い，レプリカオブジェクトのデータ値を更新する．

3. アスペクト指向プログラミングの導入

時間駆動分散オブジェクト環境のミドルウェアが持つ機能をアスペクトで記述することにより，分散処理を実現する．アスペクトによる時間駆動分散オブジェクトアーキテクチャを図 2 に示す．アスペクトにより記述する機能は，パック処理機能，アンパック処理機能，送信処理機能，受信処理機能の 4 つである．ネットワークは自動車制御分野で広く用いられている CAN(Controller Area Network)[3] を用いる．CAN は優先度に基づくリアルタイムネットワークで，送信することができるデータサイズは最大 8 バイトである．

(1) パック処理

パック処理は，オリジナルオブジェクトのデータの更新直後に行うのが最適である．図 3 にパック処理アスペクトの例を示す．データ 2 バイトのこの例では，オブジェクト A(obA1) の属性 value の値を 1 バイトずつ読みだしてバッファに格納する．そこで，オリジナルオブジェクトの

†東京都市大学 Tokyo City University

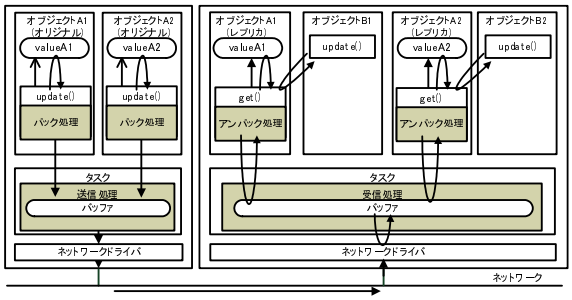

図 2 アスペクトによる時間駆動分散オブジェクトアーキテクチャ

```
before():execution(void obA1_update()){
  buf[0]=(UH)obA1_value;
  buf[1]+=(UH)obA1_value<<8;
}
```

図 3 パック処理アスペクト

```
before():execution(UH rep_obA1_get()){
  rep_obA1_value=buf[0];
  rep_obA1_value+=buf[1]<<8;
}
```

図 4 アンパック処理アスペクト

```
after():execution(void task A2(VP_INT)){
    ER err;
    err=CAN_Data_Send(buf_p,4)
}
```

図 5 送信処理アスペクト

update メソッドの実行後をポイントカットとして処理を織り込む.

(2) アンパック処理

アンパック処理は，レプリカオブジェクトのデータ値が読み出される前に行うのが最適である．図 4 にアンパック処理アスペクトの例を示す．アスペクトでは，レプリカオブジェクトの get メソッドが実行される前をポイントカットとして処理を織り込む.

(3) 送信処理

図 5 に送信処理アスペクトの例を示す．ネットワークドライバの送信処理 (CAN_Data_Send) を呼び出してバッファ中の 4 バイトのデータを送信する．送信処理は，オリジナルオブジェクトの update メソッドの処理を含んだタスクの実行後をポイントカットとして処理を織り込む．同一周期のオリジナルオブジェクトが複数存在し，かつそれぞれのオブジェクトが異なるタスクで実行される場合は，最後に処理するタスクに対し送信処理を織り込む.

(4) 受信処理

図 6 に受信処理アスペクトの例を示す．この例では，ネットワークドライバの受信処理 (CAN_Data_Recieve) を呼び出して受信したデータをバッファに格納する．受信処理は，レプリカオブジェクトの get メソッドを呼び出す処理を含むタスクが実行される前をポイントカットとし処理を織り込む.

```
before():execution(void taskB1(VP_INT)){
  ER err;
  err=CAN_Data_Recieve(0,100);
}
```

図 6 受信処理アスペクト

表 1 実行環境

ターゲットボード	H8S/2638
ネットワーク	CAN
OS	TOPPERS jsp カーネル 1.4
プログラミング言語	C 言語
アスペクト指向言語	AspeCt-oriented C コンパイラ v0.8

処理＼手法	ミドルウェアによる手法	提案手法
パック処理	59	24
アンパック処理	67	21

4. 評　　価

時間駆動分散オブジェクトモデルに従って実装したアプリケーションオブジェクトに対し，アスペクト指向プログラミングを用いて複製処理機能を開発，評価用組み込みコンピュータ上に実装した．表 1 に実装環境を示す.

本研究による実装と時間駆動分散オブジェクト環境の処理時間を評価するため，パック処理，アンパック処理を実行するために必要な命令サイクル数を測定した．測定の結果を表 3 に示す．汎用的な処理を行うミドルウェアよりも，必要となる機能に特化した処理をアスペクトで記述することにより実行効率を上げることができた.

5. ま と め

分散処理機能をアスペクト記述することで，ミドルウェアを用いずに時間駆動分散オブジェクトを実現した．本手法を用いれば，分散処理を意識せずに開発したソースコードを修正せずに，分散処理機能を追加することができる．また，ミドルウェアの開発や移植が不要になるとともに，システムジェネレータ，コンフィギュレータ等のツールも不要になり，効率的にソフトウェア開発を行うことができる.

参 考 文 献

1) Andreas Gal，Olaf Spinczyk，Wolfgang Schroder-Preikschat：On Aspect-Orientation in Distributed Real-time Dependable Systems ，Object-Oriented Real-Time Dependable Systems (WORDS 2002), pp.1530-1443
2) 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, Nov, 2007.
3) BOSCH, CAN Specification, Version 2.0, Robert Bobert Bosch GmbH, 1991.