

オブジェクト指向設計によるリアルタイムソフトウェア開発手法の提案

3K-3

株式会社 東芝, システム・ソフトウェア技術研究所
西村 朋子, 五十嵐 真悟, 小山 徳章

1. はじめに

近年,リアルタイムシステムの需要は急増し,リアルタイムソフトウェアの生産性の向上が急務となっている。しかしこの分野のソフトウェア(以下ソフト)は,性能の要求が厳しい,並列性を含んでいるなどの要因があり,従来の設計手法の適用が困難な部分があった。

一方,オブジェクト指向設計は,実世界における並列性も包含してモデル化できるのでリアルタイムソフト開発に有効であると思われる。しかし現状のオブジェクト指向設計法では性能重視のリアルタイムソフト開発のための具体的な設計指針がないという問題点がある。

我々は本稿で,オブジェクト指向設計法を用いたリアルタイムソフト開発手法を提案する。この開発手法では,特に性能の予測と検討を重視している。本手法を支援する開発支援環境は,今後構築予定である。

2. 背景

2.1. リアルタイムソフト開発

リアルタイムソフト開発は,与えられる性能仕様が非常にシビアで,要求された性能が満足できないと,ソフトの設計やコーディングを変更せざるを得ない場合も多い。そのため,設計の途中あるいは設計終了後,コーディング終了後に,性能の検証(予測または実測)が必要となる。

性能の検証は,静的検証と動的検証の2種類に大別できると考えられる。前者は,プロセス代数,形式的仕様記述などに代表されるもので,デッドロック,飢餓状態の検出等を行う。後者は,シミュレーションに代表されるもので,特定の状況下における個々のタイミングを確認し,性能仕様の実現性を検討するものである。

リアルタイムソフト開発には両者が必要であるが,本稿では特に動的検証に注目する。しかしシミュレーションを用いても,非決定的な外部イベントの発生や並列性による同期等も考慮しなければならないため,すべての状況を網羅することは事実上不可能である。そのため"検証"ではなく"検討"するための手段としてシミュレーションを用いる。

2.2. オブジェクト指向設計法

リアルタイムソフト設計には,並列性の概念が必須であるが,オブジェクト指向では各オブジェクトが並行動作すると考えることも可能であり,親和性がよい。

既存のオブジェクト指向設計法は,Booch法とOMT法が有名である。この2つの設計法のリアルタイムソフト開発への適用について検討する。

(1)Booch法

Boochが提案した設計法(以下Booch法)[1]は,オブジェクト指向設計全体を包括する設計法としてまとめられ,分析より設計に重点をおいている。

Booch法ではオブジェクトの並列性を明示的に扱っており,オブジェクトをActive(独自に制御のthreadを持つもの),Blocking,Sequentialに分類する。しかしそれをその後の設計にどのように利用するかという指針については触れていない。

また,タイミングチャート(Timing Diagram)を記述する点も,リアルタイムソフトの開発には有効であると思われる。しかし,これもドキュメントとするだけで,その活用についての具体的な指針はない。

(2) Object Modeling Technique(OMT)法

OMT法[2]は,分析から設計まで一貫したオブジェクトモデルで記述でき,他の分析設計法との親和性もよいとされている。

OMT法の中のシステム設計の段階で,オブジェクトの並列性を検討すべきであり並列タスクにすべきであると述べられているが,その検討法は具体的ではない。

また,文献[2]中では,リアルタイムシステム設計は複雑であり,厳しい性能要求を満たすために非論理的な設計が必要となるので取り扱わないと記されている。

2.3. 本研究の目的

本研究では,リアルタイムソフト開発に適用するためにBooch法,OMT法に不足している部分を補足することを目的とする。特にシミュレーションによる性能仕様の検討を重視している。

性能仕様のうちシミュレーションにより検討できるのは,個々のタイミングに関するものである。タイミングに関する仕様(以下,タイミング仕様)は,

- ・ 周期性 「〇〇周期で××する」
- ・ 応答性 「××から〇〇後に△△する」

の2種類のどちらか,またはこの組み合わせで表現できる。

3. リアルタイムソフトウェア開発手法の提案

3.1. 概要

本稿で,オブジェクト指向設計によるリアルタイムソフト開発手法を提案する。これは特にシミュレーションによる性能検討に重点をおく。

図1に本稿で提案するリアルタイムソフト開発手法の支援環境を示す。

3.2. ポイント

Booch法,OMT法に補足したポイントを以下に詳述する。

3.2.1. Activeオブジェクト

設計中に並列性を明示的に扱うために,Activeオブジェクトを導入する。これはBooch法におけるActiveオブジェクトと同義で,内部に制御のthreadを持っているものである。Activeでないオブジェクトは,Passiveオブジェクトと呼ぶこととし,Activeオブジェクトからアクセスされる。Activeオブジェクトはプロセスとして実現される。

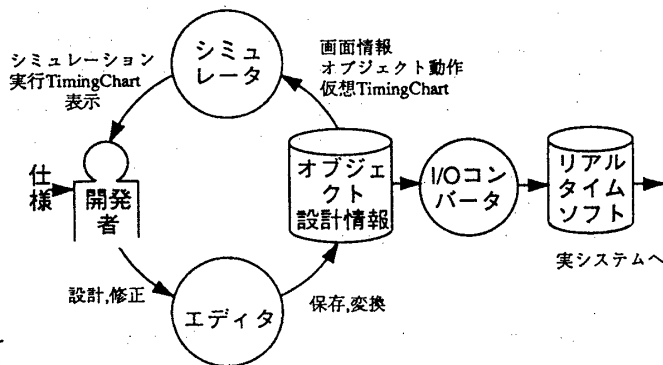


図1 システム概要図

Active/Passiveの情報は、その後の、オブジェクト間のメッセージ通信の向きと同期/非同期の決定、仮想タイミングチャートの作成、オブジェクトをプロセスとして実装するかどうかの決定などの設計作業に有効に使用される。

3.2.2. 拡張有限状態機械

オブジェクトの動作をモデル化するために、ローカル変数を持つ拡張有限状態機械(Extended Finite State Machine, 以下EFSM)を導入した。オブジェクトは内部データ及び内部状態を持つため、EFSMで無理なく表現できる。

オブジェクトへのメッセージ通信は、オブジェクトの状態に直接作用するものと、オブジェクトの状態を問い合わせるだけの通信の2種類に分類することができる。つまり、

オブジェクト=EFSM+状態問い合わせ部分と定義することができる(図2参照)。オブジェクトは並列に動作するので、遷移中あるいはアクション中に他オブジェクトが状態を問い合わせる可能性もある。ソフト全体の性能を向上させるために、状態問い合わせ部分は、EFSMが状態遷移中、アクション中にかかわらず、迅速に問い合わせに返答するものとする。

EFSMのイベントやアクション、状態問い合わせルーチンについて、それぞれ必要な処理時間を見積もる必要がある。シミュレーションにこの見積処理時間を用いる。また、この動作仕様は、シミュレーションのためのプログラム生成にも活用される。

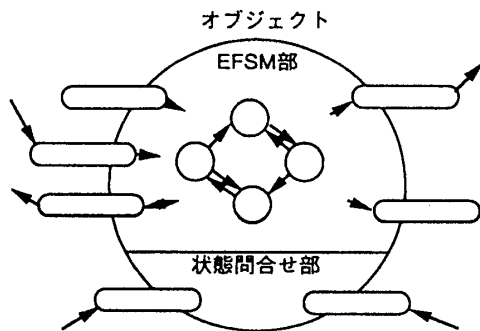


図2 オブジェクト概念図

3.2.3. 仮想タイミングチャート

シミュレーションで性能を検討するために全状況を網羅するのは事実上不可能なので、予め想定した状況下での「仮想タイミングチャート」を記述しておきシミュレーション結果との比較により検討を行う。

仮想タイミングチャートには、条件として状況と外部イベント列を与える。状況は、タイミングチャートの最初の時点での状態定義であり、各オブジェクトの状態の組合せで定義する。外部イベント列は、外部からの入力イベントの発生を時間の経過とともに定義したものである。

仮想タイミングチャートには、これらの情報を条件とし、オブジェクト間の動作を仮想的に記述する。特に、Activeオブジェクトのthreadを起動するイベントから、そのthread内のオブジェクト間のメッセージ通信に注目して記述するべきである。

3.3. 設計手順

本手法によるリアルタイムソフトの設計手順を述べる。

(1) I/O画面の定義

シミュレーションのため、リアルタイムな入出力(I/O)を画面に何らかの形式で表示する。

(2) オブジェクト設計

(2-1) オブジェクトの抽出

(2-2) オブジェクト間の関係の定義

(2-3) メッセージ通信の定義(3.2.1.参照)。

(2-4) オブジェクトの動作の定義(3.2.2参照)

(3) 動作の予測(3.2.3参照)

(4) シミュレーション

設計情報をプログラミング言語(プログラムのスケルトン)に変換し、それをシミュレートする。このため、設計直後でも、オブジェクト実装終了後でも、同様にシミュレートできる。

(4-1) スケジューリング方式の決定

(4-2) 機能シミュレーション

I/O画面から外部イベントを入力して全体の動作を確認する。オブジェクトは、実装前なら動作仕様(から生成されたスケルトン)、実装後ならそのコードを元に動作する。

(4-3) 性能シミュレーション

仮想タイミングチャートの条件部(状況、外部イベント列)を入力として全体を動作させて得られた実行タイミングチャートを、仮想のそれと照合することにより、タイミング仕様が満足されているかどうかを検討する。実行タイミングチャートの作成には、オブジェクトの各処理の見積処理時間を使用する。

(5) オブジェクトの実装

段階的に、実装が終わったオブジェクトからシミュレートを行い、機能性能の確認を行いながら開発を進める。

(6) リアルタイムシステムへの移行

4. 結論

我々は本稿でオブジェクト指向設計手法を用いたリアルタイムソフト開発手法を提案した。本手法は、Booch法、OMT法を補足したもので、タイミングチャートを用いた性能の予測と検証を重視している。タイミング仕様の形式化、仮想及び実行タイミングチャートの自動検証によるタイミング仕様検証の自動化は、今後の研究課題である。この開発手法を支援する開発支援環境は、今後構築予定である(図3参照)。

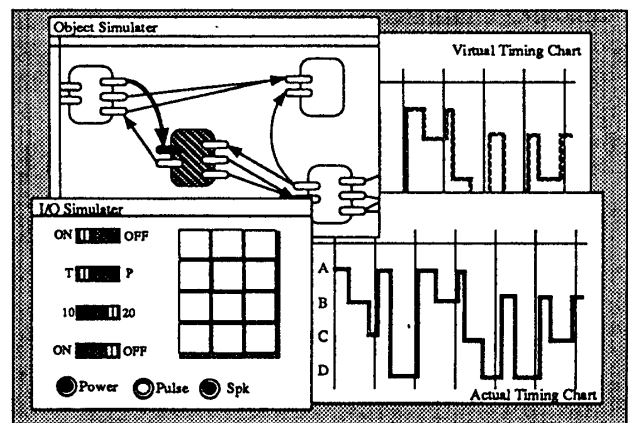


図3 システム画面イメージ

[参考文献]

[1] Grady Booch. Object-Oriented Design With Applications. Benjamin/Cummings, 1990
 [2] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. Object-Oriented Modeling and Design. Prentice Hall, 1991