

ハードウェア/ファームウェア/ソフトウェア機能分割法の提案

3Z-8

ダエン マロワ* 田口 昌親* 武内 惇* 藤本 洋* 大関 隆一** 藪田 孝造**

*日本大学工学部

**株式会社 富士通キヤドテック

1. はじめに

組込みシステムの複雑化や大規模化に伴い、システムを構成するハードウェア/ファームウェア/ソフトウェア（以下H/F/S）の効果的な機能分割法や協調設計法の実現が望まれている。筆者等は組込みシステム開発の品質や生産性の向上を図るため、システム構成要素の変更容易性と再利用性の実現に注目し、オブジェクト指向に基づくH/F/Sの機能分割法を提案している[1]。

前回の提案では第一段階で、変更容易性と再利用性に注目して責務を求め、第二段階では、責務に求められる処理時間制約に注目して責務の分割・併合を行いH/F/Sの機能分割を行う。そのため、処理時間制約が大きいシステムの機能分割が段階的に進まないという問題がある。

本稿では、時間制約をはじめに検討することによりこれらの問題を解決可能とする、時間制約の大きい組込みシステム向けのH/F/Sの機能分割法を提案する。

2. 変更容易性・再利用性に注目したH/F/S機能分割法の課題

2.1 機能分割法の概要

第一段階では、システムの変更容易性と再利用性を重視して要求仕様より責務を抽出した。各責務をその役割を基に、ドメイン4階層に割り当てる。（表1参照）

表1. 責務の役割とドメイン4階層の割り当て

責務の役割	ドメイン4階層
要求されるデータ処理を行なう	アプリケーション層
データの入出力などユーティリティとしてデータ処理で共通的に利用される	サービス層
責務の実行制御を行なう責務	アーキテクチャ層
実際に責務を実行する機構を実現する	インフラストラクチャ層

その後、H/F/Sの変更容易性に基づいて以下のように責務の実現機構を最初に、H/F/Sに分類する。

- ①アプリケーション層とサービス層の責務は変更可能性が高いためソフトウェアで実現する
- ②インフラストラクチャ層の責務は変更可能性が低いことからハードウェアで実現する
- ③アーキテクチャ層は変更可能性が高くないためファームウェアで実現する

第二段階で分類した責務の特性をユースケースやインタラクション系列、クラス関連図を用いて明確にしなが、処理時間要求を満たすように責務の分割・併合を行い、H/F/Sの機能分担を改善する。

2.2 機能分割法の課題

- ① 第一段階では、システムの変更容易性と再利用性の実現に注目したため、時間制約の大きい組込みシステムのH/F/Sの機能分割を行なう時、ファームウェアで実現すべき責務が曖昧で責務の抽出が難しい。
- ② この方法では責務に求められる変更容易性と再利用性の実現をはじめに検討するため、時間制約の大きい組込みシステムのH/F/Sの機能分割を行う時、第一段階でアーキテクチャ層の責務の抽出は難しく、第二段階で関連する多数の責務の分割・併合の対象が必要になる。

3. 処理能力に注目したH/F/S機能分割法

3.1 機能分割プロセス

図1に処理時間制約に注目したH/F/S機能分割プロセスを示す。

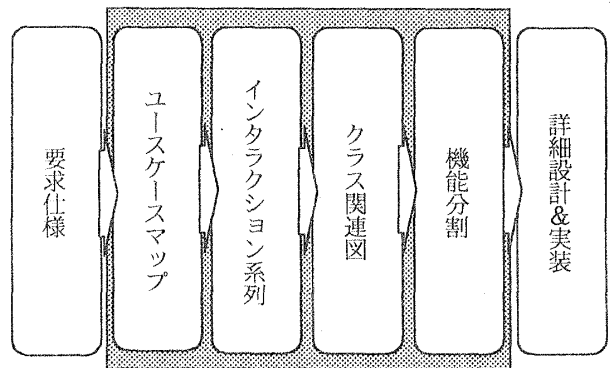


図1. 機能分割プロセス

責務の処理時間と共に責務の処理時間に影響を与える実行回数（アクセス回数）も処理時間制約を検討するための検討項目とする。

3.2 ユースケースマップモデル (作成) によるアクセス回数の判定法

ユースケースマップモデル (図2参照) の作成で、責務のアクセス回数を考慮する。

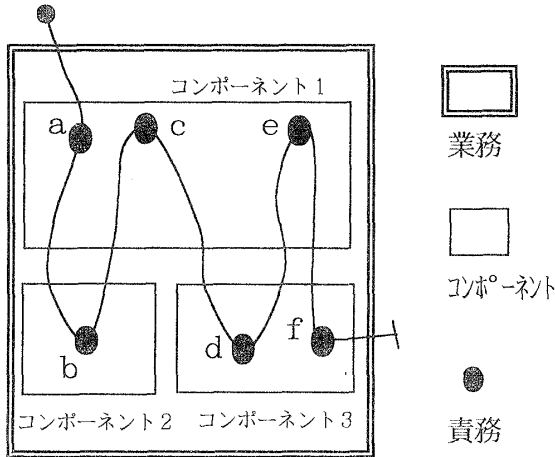


図2. ユースケースマップモデル

アクセス回数はユースケースマップで表現される。一連の責務実行中に実行される責務の実行回数の最大値である。アクセス回数が大きな責務は高速処理を行うためハードウェアで実装する部分が大きくなるものと考えられる。

3.3 インタクション系列作成による処理時間制約のある責務の判定法

インタクション系列 (図3参照) の作成で、責務の実行速度を考慮する。

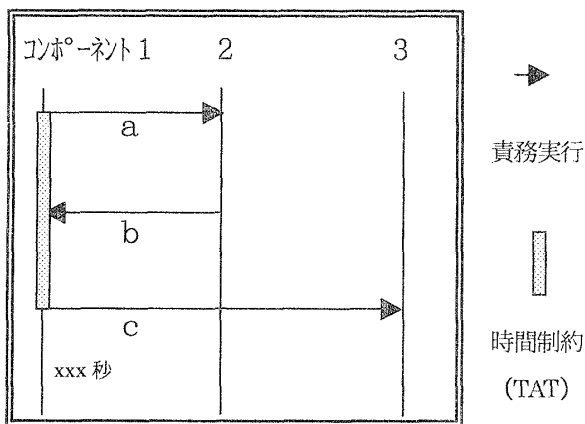


図3. インタクション系列

処理時間制約はインタクション系列で表現されるいくつかの責務によって実行される責務の TAT (Turn Around Time) を満たすために求められる特定の責務の処理時間制約である。

処理時間制約が大きな責務は高速処理を行うためハードウェアで実装する部分が大きくなるものと考えられる。

3.4 クラス関連図による変更容易性及び再利用性の判定

業務を実現するために使用される全ての責務を集め、責務をクラスと考えてクラス関連図を作成する。ここでは、複数のクラスを抽象化することによるスーパークラスを設定したり同様の責務を持つクラスを親クラスと設定することにより責務の再利用や変更を容易にする。スーパークラス (親クラス) はソフトウェアで実装する。一方、処理時間制約の大きい責務に関してはスーパークラス (親クラス) は作らず、単独で実装する。

3.5 H/F/S 機能分割法

責務を以下の視点に従って H/F/S に分割し、それぞれ名称を付与する。

表2. H/F/S の名称付与ルール

	H/F/S 分割の視点	名称付与ルール
ソフトウェア	アプリケーション	要求仕様書に表われる処理名、業務
ファームウェア	ハードの変更により変更	ハードウェア動作の制御名
ハードウェア	変更されない	ハードウェアの機能名、動作名

責務を構成する H/F/S 部分 (当該責務のハード責務、ファーム責務、ソフト責務と呼ぶ) に分割する。ハード責務、ファーム責務、ソフト責務毎にそれぞれの名称をもとに共通化可能な責務を見つけ1つに結合する。

4. おわりに

現在、本方式を実時間処理の厳しい通信システムの H/F/S 機能分割に適用中である。

参考文献

- [1] 田口、マロワ、佐藤、武内、藤本、大関、蘭田：ハードウェアファームウェアソフトウェア機能分割の一考察、電子情報通信学会総合大会、1999年
- [2] 佐藤、マロワ、武内、藤本、蘭田：デザイン技術によるハードウェアファームウェアソフトウェア機能分割法、第41回日本大学工学部学術研究報告会、情-2-7、1998年
- [3] R.J.A.ブーア、R.S.カッセルマン：ユースケースマップ、(株)トッパン、1998年
- [4] S.Schlaer, S.J.Mellor：続オブジェクト指向システム分析、(株)近代科学社、1995年