

責務分析に基づくハードウェア/ファームウェア/ソフトウェア機能分割法に関する研究

田口 昌親† 金子 正人†† 武内 惇†† 藤本 洋†† 大関 隆一††† 藺田 孝造†††

日本大学大学院工学研究科† 日本大学工学部†† 富士通キヤドテック株式会社†††

〒963-8642 福島県郡山市田村町徳定字中河原 1

電子メール : { taguchi@fujimo.ce.nihon-u.ac.jp, kaneko@ultra7.ce.nihon-u.ac.jp,
takeuchi@csse00.ce.nihon-u.ac.jp, fujimoto@fujimo.ce.nihon-u.ac.jp,
ryu@fjct.ts.fujitsu.co.jp, sonoda@fjct.ts.fujitsu.co.jp }

組込みシステムを対象に責務分析に基づくハードウェア/ファームウェア/ソフトウェア機能分割法について研究を行っている。この方法論は、次の特徴を有する。第一に、構成要素として責務を検討することで、ハードウェア、ファームウェア、ソフトウェアの視点からシステムの特性を検討でき、ハードウェア、ファームウェア、ソフトウェアどちらで実現するのかの選択について議論できる。第二に、責務を用いることにより機能分割の検討プロセスを残すことができ、さらに、他のシステムの構築に対してこの検討プロセスを再利用できる。

本稿では、責務とコンポーネントの定義、責務分析に基づくハードウェア/ファームウェア/ソフトウェア機能分割法の詳細について提案する。

A Study on the Partitioning Method for Hardware/Firmware/Software based on the Analysis of Responsibilities

Masachika Taguchi†, Masato Kaneko††, Atsushi Takeuchi††, Hiroshi Fujimoto†††,

Ryuichi Ohzeki†††, Kouzou Sonoda†††

Graduate school of Engineering Collage of Engineering Nihon University†,

Collage of Engineering Nihon University††, Fujitsu Cadtech Limited†††

Koriyama, Fukushima, 963-8642, Japan

E-mail : { taguchi@fujimo.ce.nihon-u.ac.jp, kaneko@ultra7.ce.nihon-u.ac.jp,
takeuchi@csse00.ce.nihon-u.ac.jp, fujimoto@fujimo.ce.nihon-u.ac.jp,
ryu@fjct.ts.fujitsu.co.jp, sonoda@fjct.ts.fujitsu.co.jp }

We have been studying a hardware, firmware, and software-partitioning method for embedded system based on the analysis of responsibilities. This method has two characters. The first, considering a responsibility as a system construction element, we can represent system characteristics using responsibilities from hardware, firmware, and software viewpoint. So hardware, firmware, and software designer can discuss design alternatives using the same design representations of responsibilities. The second, using responsibilities we can record the partitioning decision process, and reuse this decision process for other system constructions.

In this thesis, we will propose a definition of responsibility and component, and partitioning method for hardware, firmware, and software based on the analysis of responsibilities in details.

1. はじめに

(1) 従来のシステム開発の問題

組み込みシステムの大規模化、多様化、複雑化により、システム開発において設計生産性の向上が最大の

課題となっており次の3つの問題がある。第一に、システムの機能分割が設計者の勘と経験により行われているため、システム統合時に多くの不具合が生じる可能性がある。第二に、ハードウェアの仕様が決まり、その後、ファームウェア、ソフトウェアの仕様が決まるので、

ハードウェア、ファームウェア、ソフトウェアの開発を同時に開始することができない。第三に、ハードウェア、ファームウェア、ソフトウェアがそれぞれ独立して設計が行われているため、システムの統合時に多くの不具合が生じる。

(2) 研究目的

従来のシステム開発には、以上に述べたような問題があるため、システム統合時に多量の不具合が生じ、設計工程の後戻りにより設計工数が増加する。このため、開発期間が長期化するとともに、開発コストの増加によるシステム開発の生産性が低下する。このような問題を解決してシステム開発の設計生産性を改善するために、デザイン技術を研究している。

(3) 研究アプローチ

図1に、システム開発プロセスとデザインの研究アプローチについて記述する。

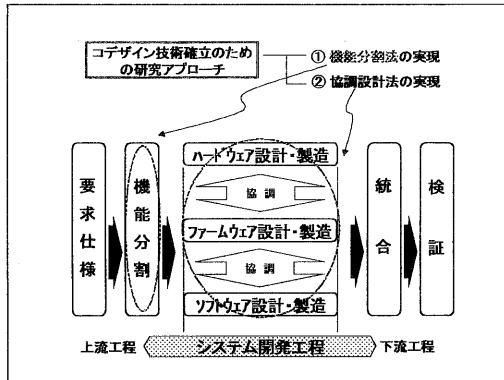


図1. デザインの研究アプローチ

デザイン技術を確立するためのアプローチは、機能分割法と協調設計法の2つに分類する。

機能分割法とは、高次設計においてハードウェア、ファームウェア、ソフトウェアの高精度なシステムの分析に基づき、システムに対して最適なハードウェア/ファームウェア/ソフトウェアの機能分割を行うための手法論である。協調設計法とは、設計段階ごとにハードウェア、ファームウェア、ソフトウェアそれぞれの設計内容を確認し、設計プロセスの後戻りを改善するための方法論である。

現在、デザイン技術を実現するために機能分割法について研究を進めている[1][2][3]。

2. 機能分割実現の課題とアプローチ

機能分割法を実現するための課題とその解決のアプローチは、以下の通りである。

(1) H/F/S を統一的に表現する方法の開発

ハードウェア、ファームウェア、ソフトウェアの多様な視点からシステムの振舞いを検討するために、ユースマップ[4]を基にした責務、コンポーネントを用いる。そのために、責務、コンポーネントの使用法を明確にする必要がある。

(2) システムの特性を検討できる仕組みの開発

システムの実時間制約の検討については、インタラクション図によって責務の実行時間、実行タイミングを検討することにより解決する。システムの再利用性の検討については、責務関連図によって責務の変更可能性、再利用性を検討することにより解決する。検討するシステムの特性を表現するために、機能分割の仕組みの中でインタラクション図、責務関連図の取扱法を決める必要がある。

(3) システムの機能を H/F/S へ割り付けるための仕組みの開発

システムの機能をハードウェア、ファームウェア、ソフトウェアへ割り付けるためにシステム設計者が検討すべきプロセスを明らかにすることにより、ハードウェア、ファームウェア、ソフトウェアのトレードオフを検討する仕組み（機能分割事例集、責務分析表）を開発する。

3. 責務分析に基づく H/F/S 機能分割法

3.1 責務、コンポーネントの定義

(1) 責務の定義

責務は責任分担を有する作業であると定義する。つまり、責務は、システムの役割を果たすために、システムの各部分が行うべき作業である。責務は、次のような特性を持つ。第一に、責務は達成すべき目的を有する大粒度（高い抽象レベル）の作業群である。第二に、一連の責務の実行過程（パス）において、原因・結果の関係がある。第三に、外部実体（アクタ）からの刺激により、パスの最初の責務が実行される。第四に、責務はリスを処理するため時

間を要する。

責務間で原因、結果関係が存在するため、同一責務でもその事前条件が異なると処理時間等が異なる。

(2)コンポーネントの定義

システムや装置は、コンポーネントの集合体でありかつシステムの役割を実行する主体であると定義する。また、コンポーネントは、責務をグループ化したものである。コンポーネントは、次のような特性を持つ。第一に、協調するコンポーネントの集合がシステムであり、その協調の結果、システム全体としての目標を達成するものである。第二に、責務が遂行されている間だけ、そのコンポーネントが必要とされる。第三に、各コンポーネントには階層関係が存在する。

システムの役割に注目し、関係する責務のグループ化を行うことにより、いくつかのコンポーネントが抽出できる。すなわち、システム自身が最も大きなコンポーネントとなり、その内部の役割に着目したコンポーネントが階層的に存在する。

3. 2 機能分割の作業プロセス

3. 2. 1 作業手順

表1に、機能分割の作業プロセスを記述する。要件定義の段階では、ユーザ要求を明らかにするためにユースケースの抽出を行う（作業プロセスは、アタリ一覧表、ユースケース一覧表、ユースケース記述表、ユースケースモデルである）。

表1. 機能分割の作業プロセス

工程	作業分類	作業プロセス
要件定義	ユーザ要求の抽出	ユースケースの作成
高次設計	システムの振舞いの明確化	責務の抽出 コンポーネントの抽出
	責務の特性の明確化	インタラクション図の作成 責務関連図の作成
	責務のH/F/Sへの分割	責務分析表の作成

高次設計の段階では、まずシステムの振舞いを表現するために、ユースケースマップを作成する（作業プロセスは、責務一覧表、コンポーネント一覧表、ユースケースマップである）。次に、責務の特性を明確化するために、インタラクション図、責務関連図を作成する。最後に、前

工程で明らかになった責務の特性から責務毎にハードウェア、ファームウェア、ソフトウェアに分割（実装方式を決定）する（作業プロセスは、機能分割事例集、責務分析表である）。

3. 2. 2 作業プロセス

(1)ユーザ要求の明確化

①ユースケース図の作成

システムが持つ機能を利用する外部の実体としてアクタを抽出する。また、システムがそのアクタに対して何らかの形で提供する機能を表現するものとしてユースケースの抽出を行う。そして、ユースケース記述表にユースケースの要件をまとめて、ユースケースモデルを作成する。これにより、システムに対する外部からの要求とシステム内部の機能を明確に区別する。

(2)システムの振舞いの明確化

①責務の抽出

前工程で作成したユースケースモデルの各ユースケースからそれぞれ責務の抽出を行う。抽出される責務は、各ユースケースを実現するためにシステムが遂行すべき作業の系列として表現される。ユースケース記述表のユースケース記述（基本系列）の各手順を一つの責務と考えて責務の抽出を行い、その結果を責務一覧表に記述する（表2）。

表2. 責務一覧表の記述項目

視点	記述する内容	備考
ユースケース名	責務が属するユースケースの名称を記述する。	必須
名前(識別)	責務を一意に指定することができる名称を記述する。	必須
責務の内容	責務が遂行するアクションを記述する。	必須
実行主体	責務が属するコンポーネントの名称を記述する。	必須
事前条件	責務を実行する前に満たしていなければならない条件を記述する。	必須
事後条件	責務を実行した後に満たしていなければならない条件を記述する。	必須
再利用性	特に他のユースケースで利用されるような責務の場合、この欄にチェックを入れる。	
変更可能性	特に変更される可能性のある責務の場合、この欄にチェックを入れる。	
時間制約	特に時間制約を持つ責務の場合、この欄にチェックを入れる。	

ユースケース名、名前(識別)、責務の内容、実行主体、

事前条件、事後条件は責務の基本属性と位置付け責務が責務であるために最低限必要な項目であり、再利用性、変更可能性、時間制約は、追加属性と位置付け責務の実装方式を検討するために必要な項目である。次に、抽出した責務をユースケースマップで表現する（初期段階のユースケースマップの作成）。各ユースケースでそれぞれ抽出した責務同士で同一のものがあれば統一して表現する。

②コンポーネントの抽出

コンポーネントは、関連のある責務をグループ化することにより抽出する。開発対象のシステムに関して知識がある時は、コンポーネントが先に抽出されることもある。抽出したコンポーネントは、コンポーネント一覧表に記述してまとめる（表3）。

表3. コンポーネント一覧表の記述項目

視点	記述する内容	備考
ユースケース名	コンポーネントが属するユースケースの名称を記述する。	必須
名前（識別）	コンポーネントを一意に指定することができる名称を記述する。	必須
コンポーネントの内容	システムにおけるコンポーネントの役割を記述する。	必須
責務関連	コンポーネントにおいて遂行される責務を記述する。	必須
階層関係	コンポーネントが有するコンポーネントを記述する。	

ユースケース名、名前（識別）、コンポーネントの内容、責務関連は、コンポーネントの基本属性と位置付けコンポーネントがコンポーネントであるために最低限必要な項目であり、階層関係は追加属性と位置付け、コンポーネント自身がコンポーネントを所有する場合そのコンポーネントを記述する。

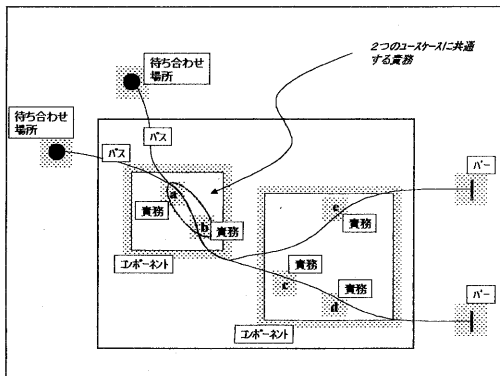


図2. ユースケースマップ

責務とコンポーネントの抽出が終わったら、責務一覧表、コンポーネント一覧表から初期段階のユースケースマップをもとにしてユースケースマップ（図2）を作成する。ここでは、初期段階のユースケースマップにコンポーネントを追記する。

(3) 責務の特性の明確化

①インタラクション図の作成

前工程で作成したユースケースマップから、インタラクション図を作成する（図3）。

この図は、責務が持つ実時間制約を検証するために作成する。ここでは、責務の時間制約（責務間の実行タイミング）を表現する。この情報により、責務、コンポーネントの構成についてシステムのリアルタイム性を考慮して検討する。

インタラクション図の作成により、もし不具合が検出された場合は、責務・コンポーネントの構成を見直す必要がある。

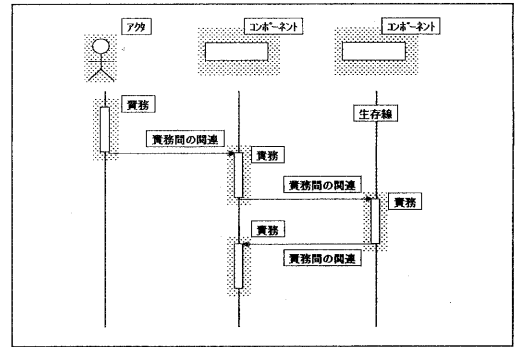


図3. インタラクション図

②責務関連図の作成

前工程で作成したユースケースマップから、責務関連図を作成する（図4）。

この図は、責務が持つ共通な機能を抽出し、責務の再利用性を検証するために作成する。ここでは、責務の変更可能性、再利用性を表現する。これらの情報により、責務、コンポーネントの構成についてシステムの再利用性を考慮して検討する。

インタラクション図の作成により、もし不具合が検出された場合は、責務・コンポーネントの構成を見直す必要

がある。

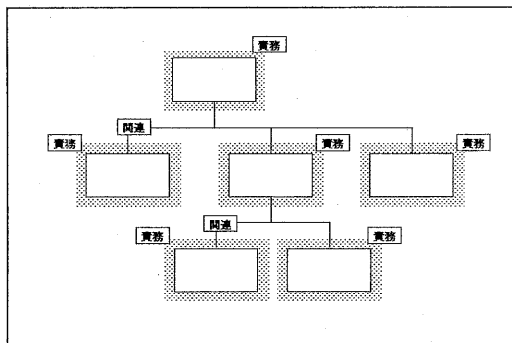


図4. 責務関連図

(4) 責務の H/F/S への分割

① 責務分析表の作成

機能分割事例集は、抽出した責務を実装するための技術項目を明確化するために作成する(表4)。

表4. 機能分割事例集

選択項目	実現方式	特性		
		コスト	速度	再利用
責務の実現法を決めるために考えるべき技術項目について記述する。	選択項目を実現するための方式を記述する。	責務を指定した実現方式で実装した時のコストを記述する。	責務を指定した実現方式で実装した時の速度を記述する。	責務を指定した実現方式で実装した時の再利用性を記述する。

責務分析表は、抽出した責務を実現するために考えられる機能分割のパターンとそのトレードオフの方式で実現する場合に予測されるコスト、速度、再利用性を表記することで、責務の実装方式を検討するための枠組みを提供する(表5)。この枠組みにより、責務を実現するためのハードウェア、ファームウェア、ソフトウェアの具体的な分担を決定することが可能になる。

表5. 責務分析表

トレードオフの方式	構成要素	判断基準		
		コスト	速度	再利用
可能な責務の実現方式の組み合わせを記述する。	どのような実現方式を選択したのかを記述する。	選択した責務実現方式のコストの評価値を記述する。	選択した責務実現方式の速度の評価値を記述する。	選択した責務実現方式の再利用性の評価値を記述する。

前工程で作成した機能分割事例集の選択項目の各実装方式を組み合わせることにより責務を実現するための方式を責務一覧表にあるトレードオフの方式の欄に記述する。判断基準の欄に機能分割事例集に記述されている特性に関する評価値を総合した値を記述する。

② 責務の実現方式の検討

機能分割は、責務一覧表の追加属性と責務分析表で示された判断基準を照合することにより行う。機能分割の難しい部分は、機能分割の方式が1通りではなく、複数考えられることである。すなわち、同一の機能を実現するさまざまな方式のうち、最適な方式を決定する必要がある。

4. おわりに

提案する機能分割の方式を時間制約の要求される通信システムに適用し、システムの時間制約・再利用性を検討できること、および、その設計プロセス(検討内容)が明確に表現できることから、本方式の有効性に関する見通しが得ている。

今後は本機能分割法をさまざまなシステムに適用して検証を進める予定である。

参考文献

- [1] 田口、金子、武内、藤本、大関、藪田：責務分析ベースのデザイン技術に関する研究、平成12年度修士論文中間発表会、B-2、2000
- [2] 田口、金子、武内、藤本、大関、藪田：責務ベースのH/F/S機能分割基準の明確化に関する一考察、電子情報通信学会総合大会、D-2-6、2000
- [3] 田口、アワ、佐藤、武内、藤本、大関、藪田：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、D-6-8、1999
- [4] R.J.A.ブーア、R.S.カセルマン：エスケスマップ、(株)トッパン、1998
- [5] I.ワブツ、他：プロジェクト指向ソフトウェア工学(OOSE)、(株)トッパン、1995