

システムレベル設計における通信仕様の探索

小林憲貴† 山崎亮介† 吉田紀彦† 楢崎修二††

† 埼玉大学大学院理工学研究科

〒 338-8570 埼玉県さいたま市桜区下大久保 255

†† 長崎大学工学部

〒 852-8521 長崎県長崎市文教町 1-14

E-mail: †{kazutaka, ryosuke, yoshida}@ss.ics.saitama-u.ac.jp, ††narazaki@cs.cis.nagasaki-u.ac.jp

あらまし システムとは、機能と通信によって成り立つ。システムが複雑化される中、通信も複雑化しその設計効率が問題となっている。設計効率を改善する手段として、ソフトウェアとハードウェアの協調設計方法論であるシステムレベル設計を用いた通信の設計が考えられる。しかし、現状のシステムレベル設計では、効率的な通信の設計手段が確立していない。そこで、本研究では、システムレベル設計に通信仕様の探索を導入し、仕様の段階で通信における設計空間を限定することで、設計効率の向上を図る。車載ネットワークの設計を例に提案手法の説明を行う。

キーワード システムレベル設計、通信設計、仕様探索、車載ネットワーク、アスペクト

Exploration of Communication Specifications in System Level Design

Kazutaka KOBAYASHI†, Ryouyusuke YAMASAKI†, Norihiko YOSHIDA†, and Shuji NARAZAKI††

† Graduate School of Science & Engineering, Saitama University
Shimoōkubo 255, Sakura-ku, Saitama-shi, Saitama, 338-8570 Japan

†† Nagasaki University

Bunkyo-machi 1-14, Nagasaki-shi, Nagasaki, 852-8521 Japan

E-mail: †{kazutaka, ryosuke, yoshida}@ss.ics.saitama-u.ac.jp, ††narazaki@cs.cis.nagasaki-u.ac.jp

Abstract Systems consist of functions and communications. Recently, a complexity of systems has increased rapidly and communication also has become more complex. This complexity causes a problem for design efficiency. System level design has been invented as a solution to the problem. This is a design methodology for software and hardware co-design. However, the communication design has not been established in this methodology under the present situation. This paper proposes exploration of communication specifications in system level design, and presents an application example to vehicle network design.

Key words System Level Design, Communication Design, Specification Exploration, Vehicle Networks, Aspects

1. はじめに

システムを設計するに当たり、通信の設計は必要不可欠である。システムへの要求が複雑化する中、通信設計も同様に複雑化しており、通信設計における設計効率が問題となっている。この設計効率を改善する手段として、ソフトウェア(以下SW)とハードウェア(以下HW)の協調設計手法であるシステムレベル設計が考えられる。

システムレベル設計とは、System-on-Chip(以下SoC)設計の設計効率を改善するために考案された手法である。この設計手法は、SWとHWの統合仕様から、段階的詳細化を行い、設計効率を向上させている。この段階的詳細化では、仕様⇒アー-

キテクチャ⇒通信⇒実装の順で詳細化を行っている。

この設計手法での通信の設計は、主に「合成」による設計、つまり、既存ライブラリの活用に重みを置いている。しかし、市場要求の多種多様化、既存ライブラリの増加に伴い、「合成」を主とした通信の設計が困難となってきている。

上記の問題を解決するためには、仕様の段階で通信仕様の探索を行い、その後の段階的詳細化における通信の設計空間を狭めることで、通信の設計効率を向上させる必要がある。本研究では、システムの仕様設計の段階で、通信仕様の探索を行うことを目的とする。

通信仕様を探索するためには、通信仕様の洗い出しと設計空間の探索が必要となる。設計空間の探索を容易に行うために、

アスペクト指向を導入する。

本研究の題材として、車載ネットワークの設計に通信仕様の探索を応用する。車産業では、規制や市場要求の変化に伴い、従来の機械的な制御が難しくなっており、Electronic Control Unit(以下 ECU)を用いた電子的制御に移行されてきている。この ECU 間の通信において、様々な特徴を持つプロトコルが存在するため、通信仕様の探索が必要である。

以下、2. で現在の通信設計について説明し、3., 4. でシステムレベル設計の説明とその通信設計のための改良について説明する。5. で通信仕様の探索について説明し、6. で提案手法による例題設計を行う。最後に7. で本稿のまとめと今後の課題について述べる。

2. 通信設計

システムを設計する場合は、機能と通信の設計が必要となる。従来は、システムの機能への要求が少なく、通信も単純な設計で可能であった。しかし、現在では、機能への要求が複雑化するにつれ、通信への要求も多種多様化しており、また開発期間の短縮も求められている。しかし、これまでは通信に対する設計方法論は必要なかったため、方法論が確立しておらず、設計効率が問題となっている。

電気通信などの通信がメインとなる分野においては、仕様の設計記述 [1] は確立している。

現状では、SW 工学においても通信設計が認識され始め、UML2.0 [2], [3] において、Message Sequence Chart や Timing Chat, Specification and Description Language [4] の取り込みなど通信設計を意識した図が追加された。

しかし、設計方法論は確立しておらず、システムの通信部分の仕様から実装までの設計は困難となっている。

3. システムレベル設計

通信の設計効率の問題を解決するため、システムレベル設計を用いて通信設計を行う。システムレベル設計とは、SW と HW の協調動作である System-on-Chip の設計において、設計効率を向上させるべく考案された方法論である。

主な特徴として、以下の二つが挙げられる。

- 段階的な詳細化
- 機能と通信を切り離した設計

従来の設計方法論では、機能と通信が分離していなかったのそれぞれを独立に考える設計が出来なかったが、システムレベル設計によって機能の段階的な詳細化・通信の段階的な詳細化をそれぞれ独立に進めることが可能になり、設計効率の向上が可能となった。

システムレベル設計には、主に SpecC 方法論 [5]~[7] と SystemC 方法論 [8] の二つの方法論がある。以下に各方法論の特徴を記す。本研究では、SpecC 方法論を使用した。

- SpecC 方法論:理論が体系化されており、仕様から実装まで一貫性のある方法論であるため開発・教育向きである。
- SystemC 方法論:設計者依存の詳細化を行う。技術のある設計者や、設計期間の限られた現場向きの方法論である。

3.1 SpecC 方法論

SpecC 方法論は抽象度の高いモデル設計から始まり、徐々にモデルの抽象度を落としていく。抽象度は定められた手順に従って段階的に落としていく。各段階のモデルがシステムをどのように表現しているか定義されている。最終的に目的の抽象度のシステムモデルが得られる。図 1 に SpecC 方法論での設計フローを示す。

抽象度の段階には、以下の4つがある。

- 仕様モデル:システムの機能を忠実に再現したモデル
- アーキテクチャモデル:実装するアーキテクチャに仕様モデルの機能を割り当てたモデル
- 通信モデル:コンポーネント間の通信を実装できる抽象度に落としたモデル
- 実装モデル:実際のアーキテクチャに対応するモデル

仕様モデルは、SpecC 設計理論で最も抽象度の高いモデルである。これは、SoC に組み込まれる SW の仕様とそれを実行する HW の仕様を統合したシステム仕様のモデルである。仕様モデルに対して、実際のアーキテクチャを割り当てる。この時、実行時間や消費電量などの設計制約を見積もり、どのようなアーキテクチャに割り当てるのが良いか考える。その結果、アーキテクチャモデルが得られる。次に、各コンポーネント間の通信に実装可能なプロトコルを埋め込むことで、通信モデルができる。最後にクロック精度のモデルに変化させることで、実装モデルが得られる。この実装モデルは、実際のアーキテクチャとそのアーキテクチャ上で動く組み込み SW である。これら全てのモデルは、SpecC 言語でモデル化・シミュレーションが可能である。

この SpecC 方法論での通信の設計は、仕様モデルでの「動作と通信の切り離し」、通信モデルでの「通信合成」である。この方法論による通信設計の問題点は図 2 に示されるように以下の2点である。

- (1) 通信の詳細化が「通信合成」のみであること
 - (2) アーキテクチャ探索において、機能のアーキテクチャ設計と通信のアーキテクチャ設計が混在していること
- (1) は、通信合成は既存の部品ライブラリから最も最適な部

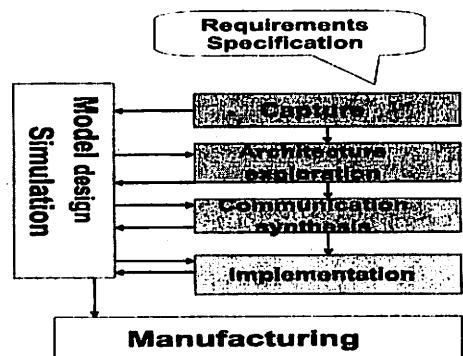


図 1 SpecC 方法論
Fig. 1 SpecC Methodology

品を選択することを意味している。この設計方法での問題点は、既存部品が急増している現在では、設計空間が大きいため最適な部品を選択することが困難であることや、選択の困難さや既存部品からの選択であることから、選択した通信がシステムに対して必ずしも最適な通信である保証がないこと、また、新しい通信の設計は考慮されていないことである。

(2)は、アーキテクチャ探索において、システムのアーキテクチャの探索を行うが、この時、機能についてのアーキテクチャの探索と通信についてのアーキテクチャの探索が混在しており、探索する上での見積もりが機能と通信で曖昧なものになってしまい、最適なアーキテクチャの探索が困難となっている。

システムレベル設計を用いて、効率良く通信の設計を行うためには、上記二つの問題点を解決しなければならない。

4. システムレベル設計の改良

3.1の(1)の解決策として、仕様設計に「通信仕様の探索」を導入することを考える。これにより、仕様の段階で設計空間を限定することが可能である。設計空間の限定は、通信合成でのライブラリの選択や、新しい通信の設計を容易にする。

3.1の(2)の解決策として、既存のシステムレベル設計のアーキテクチャ探索を機能のアーキテクチャ探索と通信のアーキテクチャ探索に切り分けることを考える。これにより、探索において、機能としての見積もり項目と通信としての見積もり項目に分けることができ、より確実に最適なアーキテクチャの探索が可能となる。

図3に提案した設計フローを示す。本論文では、通信仕様の探索に焦点を当てる。

5. システムレベル設計による通信仕様の探索

4.で提案した通信仕様の探索を行う。通信仕様の探索を行うためには、「通信仕様の洗い出し」による設計空間の定義と、設計空間の「探索」が必要となる。

5.1 通信仕様の洗い出し

通信仕様を探索するに当たり、まず最初に探索空間を定めな

ければならない。通信とは、「送信」と「受信」で成り立つため、それぞれの仕様を洗い出す。

送信仕様「いつ・だれが・どこに・なにを」送信するのかを定める。

- いつ : 送信タイミング
- 誰が : 送信者
- どこに : 受信者 (通信媒体経由)
- 何を : 送りたいデータ

送信の仕様で一番重要となるのは「いつ」送信するのかを決定することである。送信のタイミングは主にタイム・トリガ型とイベント・ドリブン型の2つが存在する。

受信仕様「いつ・だれが・どこから・何を」受信するのかを定める。

- いつ : 受信タイミング
- 誰が : 受信者
- どこから : 送信者 (通信媒体経由)
- 何を : 必要なデータ

受信仕様で重要となるのは「いつ」受信するのかを決定することと、送られたデータが必要なデータであるかどうかを判断することである。受信タイミングについては、送信仕様と同様にタイム・トリガ型とイベント・ドリブン型の2つとなる。データの識別は、送られたデータが何を表すデータなのか識別する必要があり、受信タイミングにより識別方法が変化する。

以上の通信仕様の洗い出しによって検出された通信仕様である「タイム・トリガ型」と「イベント・ドリブン型」について以下に示す。

タイム・トリガ

あるオブジェクトが送受信するタイミングが静的に定まっているものである。図4に概要を示す。

利点: 通信タイミングが静的に定まっているため、リアルタイムの制約を満たしやすく、システムの設計は容易である。

欠点: 新たな送信ノードの追加には時間制約が付随する。また、データ転送の必要が少ないネットワークでは、通信頻度が必要以上に高くなる。

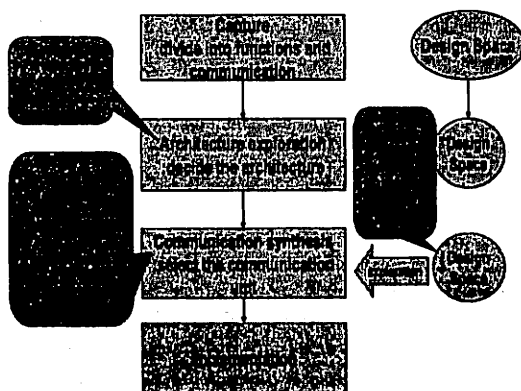


図2 SpecC 方法論:設計問題
Fig.2 SpecC Methodology:Design problems

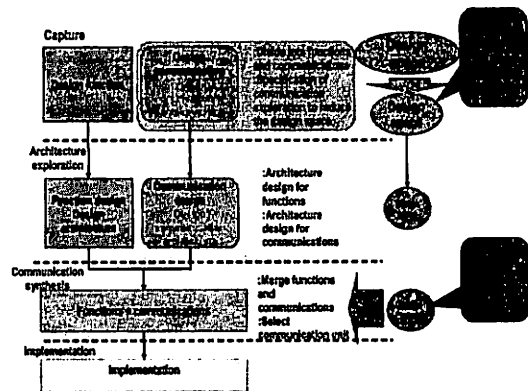


図3 設計フロー
Fig.3 Design Flow

イベント・ドリブン

オブジェクトは、オブジェクト自身のタイミングでデータ送信が可能である。つまり、送受信タイミングは動的に定まる。また、ネットワーク形態によっては、送信のためにアービトレーションが必要であったり、受信のためにフィルタリングが必要である。図5に概要を示す。

利点：必要最小限の通信頻度で使用可能であり、新たな送信ノードの追加が容易である。

欠点：通信量の増大によって、リアルタイム制約が守れない可能性がある。

以上により、通信仕様を洗い出し、設計空間を定義する。

5.2 探索方法

通信仕様の探索を行うには、設計空間を定義した後に探索を行う必要がある。そこで、設計効率を上げるためには、効率的に探索を行わなければならない。本研究では、システムの通信機能をアスペクト指向[9]とフレームレット[10]を用いて設計する。

5.2.1 アスペクト指向&フレームレット

システムとは、機能と通信に切り分けることが可能である。しかし、通信は全ての機能に付随する処理である。このような処理を「横断的」な処理という。この横断的な処理を他の動作との関連を弱めることで、再利用化や、処理の切り替えを容易に行う技術がアスペクト指向である。

アスペクト指向は以下の二つの考えから成り立つ。

- アスペクト:アスペクト指向のモジュール単位。アスペクトは以下の二つを持つ。
 - ポイントカット:横断的な処理をどこに挿入するのかを定める。

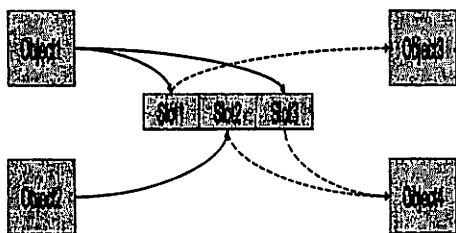


図4 タイム・トリガ
Fig. 4 Time Trigger

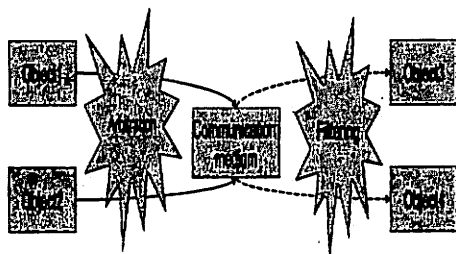


図5 イベント・ドリブン
Fig. 5 Event Driven

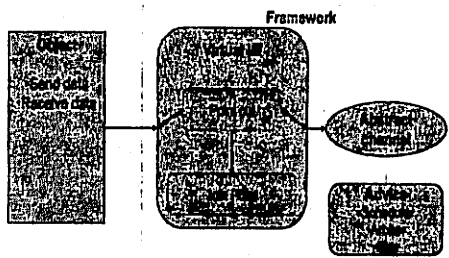


図6 フレームワーク
Fig. 6 Framework

- アドバイス:横断的な処理そのもの。
- ジョインポイント:ポイントカットで選択される場所を示す。

このアスペクト指向と類似した技術としてフレームレット技術がある。フレームレットとは、フレームワークの各機能を個別化したものであり、設計が困難なフレームワークを機能毎に分離して設計を行う技術である。

フレームレットは、デザインパターンを用いて機能を実現し、仮想インタフェース(以下仮想IF)を持つことで、どのような処理からのアクセスも可能とする技術である。

フレームワークはシステムの各処理が使用できる機能を備えたものであるため、フレームレットはフレームワークのためのアスペクト指向と考えることができる。

このアスペクト指向とフレームレットを用いて、通信を設計することで、通信機能のプラグアンドプレイが可能となり、通信仕様の探索が容易となる。

5.2.2 アスペクト指向の応用

通信機能をアスペクト指向とフレームレットを用いて設計する。

通信機能は、仮想IF・抽象チャンネル・機能に分け、この三つをフレームワークとする。それぞれの役割を以下に示す。

- 仮想IF:内部メソッドとしてジョインポイントを持ち、外部からアクセス可能なメソッドとしてポイントカットを持つ。システムの機能側はこの仮想IFに接続される。
- 抽象チャンネル:抽象的な通信を表す。仮想IFに接続し、機能を用いて通信を行う。
- 機能:アスペクト指向のアドバイスとなる部分である。通信仕様の探索を行う場合、この機能のプラグアンドプレイにより可能となる。

図6にシステムの全体図を示す。

6. 実験

6.1 車載制御ネットワーク

本研究では、制御ネットワークで最も活用され、最も多く研究されている車載制御ネットワークを設計題材とした。

近年、市場要求や規制の変化により、自動車の制御方式が変化しつつある。従来は機械的連動で制御を行っていたが、現在では、ECUを用いた電子的制御となっている。電子的制御に

は、カーナビからエンジン制御まで様々な種類があり、それぞれに対応した通信プロトコルも複数存在する。代表的な通信プロトコルを以下に挙げる。

- CAN

エンジンやトランスミッション、ブレーキなどのパワートレイン系の制御通信に使用される。メディア・アクセス方式はマルチマスタでのイベント・ドリブンとなっている。現状では、通信ネットワークを流れるデータが急増しており、CANでの対応が困難となってきている。

- LIN

CANのサブネットワークに位置するプロトコルである。主にドアやシート、ミラーなどのボディ系の制御通信に用いられる。メディア・アクセス方式はシングルマスタ方式のタイムトリガ通信が主な特徴である。

- MOST

カーナビやオーディオなどのマルチメディア系の制御通信に使用される。

- FlexRay

X-by-Wire(パワートレイン系における用途ごとに特化した制御通信)に使用される。次世代通信として開発途中である。当面の目的はCANの置き換えとなっている。メディア・アクセス方式はタイムトリガ方式である。

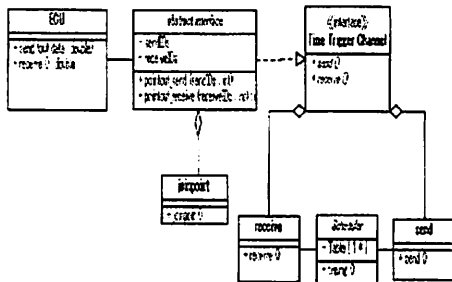


図7 タイム・トリガ UML
Fig. 7 Time Trigger UML

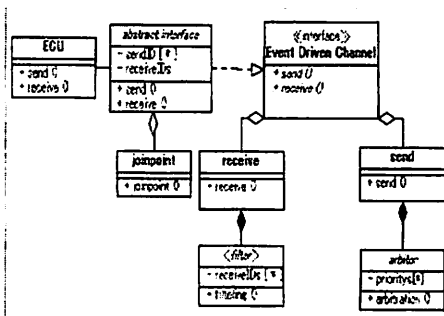


図8 イベント・ドリブン UML
Fig. 8 Event Driven UML

6.2 設計・評価

アスペクト指向を用いた通信仕様の探索を評価するため、例題の車載ネットワークを設計した。UMLを用いて設計を行い、動作確認のため、SpecCを用いて実装した。UMLの設計には、(株)テレロジックアート社[11]のPattern Weaverを使用し、SpecCによる実装には(株)インターデザイン・テクノロジー社[12]のVisual Specを使用した。図7、8に設計したタイムトリガとイベント・ドリブンのUMLを示す。

探索において、機能側では仮想インターフェースに接続するのみである。通信機能を切り替えることで、通信仕様の探索が可能となる。

7. まとめ

通信設計の効率向上のために、システムレベル設計を応用した。また、更なる効率向上のために、システムレベル設計に対して、通信仕様の探索を導入した。

今後の課題としては、通信仕様の探索における見積りである。探索するに当たり、評価基準が必要となる。その評価基準として、何のデータを測定するのか、また、そのデータから何を読み取るのかを考える必要がある。

文 献

- [1] 水野忠則. プロトコル言語. カットシステム, 1994.
- [2] (株)テレロジックアート. UML入門. エクスメディア, 2005.
- [3] 渡部雅彦, 飯田周作, 石田哲史, 山本修二, and 浅利康二. UML: 動的モデルによる組み込み開発. オーム社, 2003.
- [4] Ferenc Belina, Dieter Hogrete, and Amardeo Sarma. 仕様記述言語SDL. カットシステム, 1996.
- [5] Daniel D.Gajski, Jianwen Zhu, Rainer Dömer, Andreas Gerstlauer, and Shuqing Zhao. *SpecC: Specification Language and Methodology*. Kluwer Academic Pub, 2000.
- [6] Gerstlauer, Dömer, Peng, and Gajski. *System Design: A Practical Guide With Spec C*. Kluwer Academic Pub, 2001.
- [7] <http://www.ics.uci.edu/spec/>.
- [8] Thorsten Grötter, Stan Liao, Grant Martin, and Stuart Swan. *System Design with Systemc*. Springer, 2002.
- [9] 千葉滋. アスペクト指向入門. 技術評論社, 2005.
- [10] アレッサンドロ・パセッティ. フレームレット. 翔泳社, 2005.
- [11] <http://www.tech-arts.co.jp/>.
- [12] <http://www.interdesigntech.co.jp/vn1209.htm>.