5L-8

組み込みソフトウェア開発における ハードウェア要件の分析と検証

岡田 康治† 松浦 佐江子‡

芝浦工業大学 システム工学部 電子情報システム学科†

芝浦工業大学 大学院理工学研究科 機能制御システム専攻‡

1. はじめに

組み込みソフトウェアの開発では，ハードウェアの仕様がソフトウェアの実現可能性や実装方法に影響を与える．そのため，要求分析でハードウェアを用いて何を実現する必要があるかを明らかにし，実際のハードウェア仕様による実装方法等への影響を調査する必要がある．しかし，要求分析で想定したハードウェアの利用目的・方法がシステムを実現するためには不充分なまま実装へ進んでしまい，手戻りが発生することが多い．したがって，ハードウェアを用いて何を実現すればシステムが実現できるのか，つまりシステムを実現するためにハードウェアで実現しなくてはいけない機能は何かを要求分析段階で決定し，それがシステムの開発のために妥当であると保証することが出来れば，あとはその機能が実際にハードウェアで実現できることを確認することで後工程の手戻りを無くすことができる．

ハードウェアで実現するある機能が，システムの目的の実現のために必要であることを示すためには，その機能がシステムの目的の実現という観点から導出できる事を示すことが有効であり，逆にそれらの導出した機能がシステムの目的の実現に十分であることを示すためには，それらの機能を組み合わせて用いることによってシステムの目的が実現できるということを示せる必要がある．

本研究では，この 2 点を示すことで，システムを実現するために必要なハードウェアが実現すべき機能を分析し，その妥当性を保証する手法を目指す．本稿ではその内のモデリング言語である SysML と UML を用いてハードウェアの実現すべき機能を分析する部分について，例題への適用例を示す．

2. 例題

本研究では、参考文献[1]でモデル検査ツール UPPAAL の利用例として述べられている列車運行管理システムをベースに幾つかの条件を加えた例題を元に手法を考案している．

このシステムは複数の線路がある一部分で 1 つに合流し，再び別れるという環境上で走行する列車の運行を管理し，すべての列車が合流部分を通過できることと列車同士が衝突を起こさないことが目的である．

システムの前提は以下のとおりである．

線路は区間によって 3 種類に区別され合流地点手前 100m を待機線路，合流地点から分岐地点までの 100m を共通線路，それ以外の部分 400m を個別線路と呼び，線路は循環している．列車とゲートのサブシステムに分かれ，列車には移動用モータ，位置測定用 GPS，通信用無線，ゲートには，通過物検出用赤外線センサ，通信用無線がハードウェアとして搭載されている．各列車の性能は一様で，ゲートは共通線路の開始地点に設置されている．

3. 例題によるハードウェア要件の分析

3.1. 概念モデルの作成

手法適用前にすでに明らかになっているシステムの仕様を UML クラス図で纏め，分析と検証における共通の情報にする．図 1 に本稿の例題で明らかになっている仕様をまとめた概念モデルを示す．

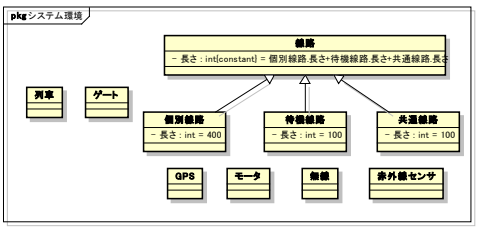

図 1 概念モデル

概念クラス内のハードウェアのクラスに記述された定義がハードウェア要件となる．この時点ではすでに明らかになっている情報をクラスとしてまとめただけである．特に図中の下に並ぶハードウェアクラス，つまりハードウェア要件は完全に空となっている，言わばハードウェアを全く利用せず，仕様に関する制約も存在しない状態である．

3.2. システムの満たすべき性質の分析

SysML 要求図を用いてシステム性質の分析を行う．ここでは，システムの実現方法に依らずに満たすべきシステムの性質を分析し，その性質を論理式的に表現する．図 2 に例題のシステムの列車同士が衝突を起こさないという目的を達成するために必要な性質を分析した結果を示す．

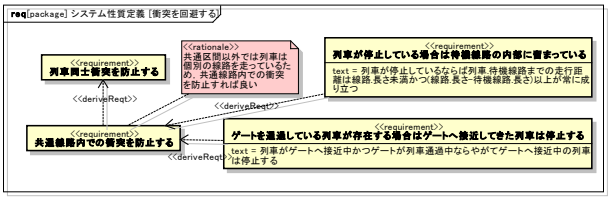

図 2 列車が衝突しないために必要な性質

衝突を防止するという目的から，2 つのシステムの性質を導出した．この 2 つの性質を列車同士の衝突を防止するという要求から直接導出してしまうと，その根拠が分からなくなるため，間にその 2 つの性質で目的が達成できるという根拠を示すための要求をおいて図全体での要求の導出関係を理解できるようにしている．

3.3. 性質からの振る舞いの分析

前項で分析した各性質について，再び要求図を用いて

Hardware Requirements Analysis and Validation in Embedded Software Development

†Kouji Okada ‡Saeko Matsuura

†Department of Electronic Information System, collage of System Engineering, Shibaura Institute of Technology

‡Division of Functional Control Systems, Graduate School of Science and Engineering, Shibaura Institute of Technology

必要なハードウェアの振る舞いを分析する．ハードウェア要件分析モデルでは，システムの満たすべき性質から，その実現に必要な振る舞いを段階的に分析し，クラスにメソッドとして定義することでハードウェア要件に結びつける． 下図に図２で分析した性質の一つ「ゲートを通過している列車が存在する場合はゲートへ接近してきた列車は停止する」という性質について，その性質から導出されるハードウェアの振る舞いを分析した結果を示す．

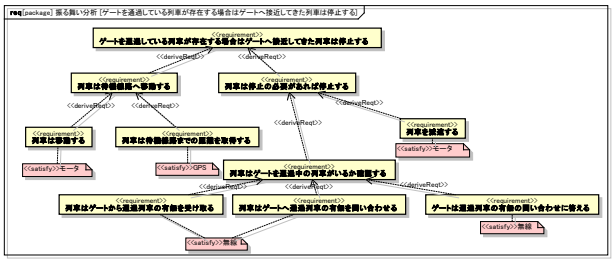

図 3 振る舞いの分析例

最上位の要求となるシステムの性質から，論理の飛躍が発生しないように達成する手段を分割してゆき，一つハードウェアで実現できる単一の振る舞いと言える粒度まで分割を行ったら末端としてその振る舞いを実現するハードウェアを対応付けて分析を終了する．

3.4. 振る舞いのフロー分析

前節までに分析したハードウェア要件に対してアクティビティ図を用いて詳細な内容の分析を行い，ハードウェアがシステムとしてどのように振る舞うかを定義したハードウェア振る舞いモデルを作成する．

このアクティビティ図ではサブシステムをパーティションの単位とする．

以下は列車についてのハードウェア振る舞いモデルである．

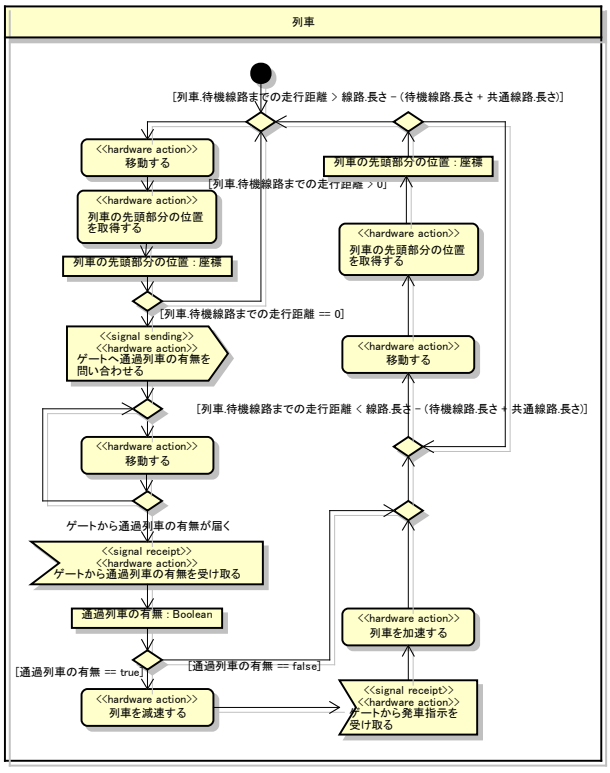

図 4 列車のハードウェア振る舞いモデル

各<<hardware action>>ステレオタイプがついたアクションノードはハードウェア要件分析モデルで分析されたハードウェアの振る舞いである．各アクションノードには以下のような定義を行う．

```
return void
time unlimited
behavior = {
        列車.待機線路までの走行距離
        -= 80 * 動作時間
         + (1/2) * モータ.減速加速度 * 動作時間 * 動作時間
        列車.走行速度 = 0
}
```

図 5 振る舞いの定義記述

return は振る舞いによって得られる情報，time は動作時間の許容範囲を表し，behavior 内の動作時間は実際のその振る舞いの動作時間を表す．この定義では新たに列車.走行速度やモータ．減速加速度等といった情報が出現したため，この情報をクラス図に追加する．なお，ハードウェアに対して減速加速度のような性能の属性を追加する場合は，仮定する性能の値も同時に追加する．

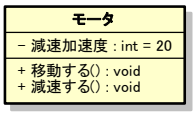

図 6 仕様が追加されたモータクラス

ここまでに分析した結果を図 1 にあるモータクラスへ反映したモータクラスが図 6 である．ここまでの分析によって，モータは減速加速度 20m/s^2 という性能を持ち，移動すると減速するという振る舞いが可能である必要があるという仕様を分析することができた．この仕様が実際にハードウェアで実現可能かどうか調査しなければいけないハードウェア要件となる．

4. まとめと今後の方針

本稿では列車運行管理の例題を用いて，運行管理システムの目的から，ハードウェアに要求する仕様を分析した．

今後は，本手法の有効性を判断するために UPPAAL による検証ができるようにし，現在の例題の分析内容の検証を行なう．

5. 参考文献

[1] Wang Yi, Paul Pettersson and Mats Daniels. “Automatic Verification of Real-Time Communicating Systems by Constraint Solving. In Proceedings of the 7th International Conference on Formal Description Techniques”, Berne, Switzerland, 4-7 October, 1994. North Holland Publisher, 1994. Pp243-258.