

# 修士論文概要書

Summary of Master's Thesis

Date of submission: 2 / 1 / 2013 (MM/DD/YYYY)

専攻名 (専門分野) Department	情報理工学専攻	氏名 Name	小澤貴之	指 導 教 員 Advisor	鷺崎弘宜 印 Seal
研究指導名 Research guidance	高信頼ソフトウェア工学	学籍番号 Student ID number	5111B022-4		
研究題目 Title	Simulink モデルの保守性向上に向けたクラスタリングおよびUML モデルとの双方向変換に関する研究				

## 1. はじめに

モデルベース開発は、開発の早い段階で要求仕様をモデル化し、シミュレーションを行う開発手法である。

組込みソフトウェア開発で用いられるモデルとして Simulink[1]モデルが挙げられる。Simulink とは The MathWorks 社によって開発された動的システムのモデル作成・シミュレーション・解析を行うソフトウェアである。Simulink モデルは抽象度の低いモデルであり、大規模で複雑になりやすい。そのため、モデルの再利用や検討・修正が困難である。またサブシステムの形成により、モデルを抽象化できるが、属人性が強いためにモデルの理解性が低くなってしまふ。

そこで、我々は Simulink モデルの保守性向上を目指して、Simulink モデルに対し、クラスタリング手法を適用し、属人性の無い適切な抽象化を行う。さらに、Simulink モデルの責務把握のために、Simulink モデルと UML モデルの双方向変換を提案する。Simulink モデルの責務を明らかにし、開発者の理解性とモデルの保守性を向上させる。

## 2. モデル間の双方向変換の実現

### 2.1. UML

UML (Unified Modeling Language : UML, 統一モデリング言語) [1]は、現在ソフトウェア開発において最も普及しているモデリング言語である。UML モデルは大きく静的な構造を示す構造図と動的な振る舞いを示す振る舞い図の2種類に分けることができ、システムを複数の側面から捉えることで、システムの理解性が高まり、保守性を向上させる。また、UML はオブジェクト指向に基づくため、再利用性の高いモデルを作成することができる。

### 2.2. Simulink モデルの問題点

組込みソフトウェア開発で用いられる Simulink モデルは実際の開発現場では数億ものブロックを用いた開発になっている[2]。そのため、Simulink モデル全体像の把握が困難になり、修正が難しい。また、開発者の側で意味を付加できるような記法も存在していない。そのため開発者と第三者との意思疎通を容易にするというモデル本来の目的を果たしていない。このことから Simulink モデルは、開発資産としての価値が低く、保守性の低いモデルとなりがちである。

次に、Simulink モデルの抽象化が属人的であるという問題が挙げられる。例として、nxtOSEK/JSP で公開されている倒立振子のコントロールモデルを挙げる。nxt/OSEK とはオープンソースの LEGO MINDSTORMS-

NXT 用開発/実行環境である。この環境は教育用レゴ マインドストーム NXT の開発を行うことができる。倒立振子のコントロールモデルの一部を図 2. 1. に示す。

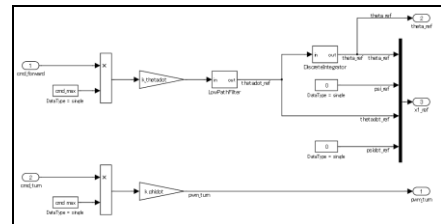


図 2. 1. 倒立振子のコントロールモデルの一部

このコントロールモデルをサブシステム化を用いて抽象化する。サブシステム化とは複数のブロックを1つのブロックとしてまとめる機能である。サブシステム化によって大規模な Simulink モデルを抽象化して見やすいモデルにすることができる。図 2. 1. を抽象化しようとすると、図 2. 2. のようにサブシステムの作成候補が複数存在し、サブシステムをどのように作成するかは開発者によって大きく異なってしまふ。

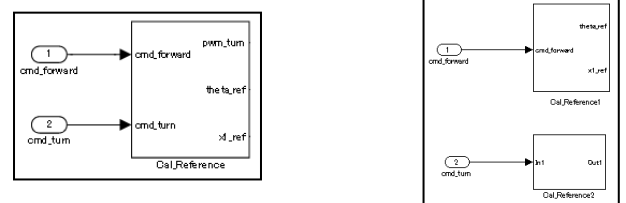


図 2. 2 サブシステム化の候補

これらのことから、Simulink モデルの抽象化・モジュール化は属人的・非体系的であり、第三者の理解を妨げているといえる。

これらの問題を解決するために、我々は Simulink モデルを理解性の高いモデルである、UML モデルに変換する。モデル変換では図 2. 3. に示すように、UML モデルで検討・修正を行った結果を常に Simulink モデルに反映し、複数モデル間のトレーサビリティを保つことが課題となる。また、人手によるモデル変換では誤りが生じてしまう可能性がある。

以上のことから、モデル変換はモデルの信頼性が保つのが難しい。つまり、問題は

- P1: Simulink モデルの責務が把握困難
- P2: 抽象化・モジュール化が属人的・非体系的
- P3: モデル変換後のモデル間の不整合が挙げられる。

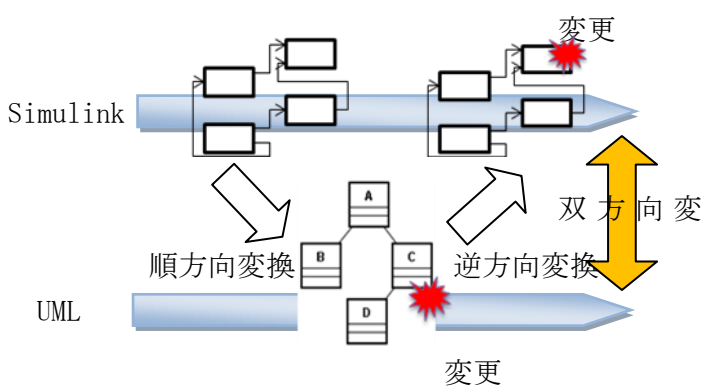


図 2.3. 双方向変換の自動化

### 3. Simulink モデルのモデルビュー提案

#### 3.1. Simulink モデルの UML モデルとしてのモデルビュー設定

Simulink モデルは静的な構造と動的な振舞いが混在したモデルであるのに対して、UML は静的な観点と動的な観点からシステムを記述する。

Simulink モデルと UML の対応関係を定義することで図 3.1. のように構造と振舞いが混在した Simulink モデルから、構造のみと振る舞いのみを抽出したそれぞれの新たなモデルビューを示す。

これによって、Simulink モデルでは構造と振る舞いが混在してしまっていたが、モデル変換を行うことで、異なる観点でのモデル評価が可能となり、Simulink モデル内の責務や関係が明示でき、理解性向上につなげ、保守性を高める。

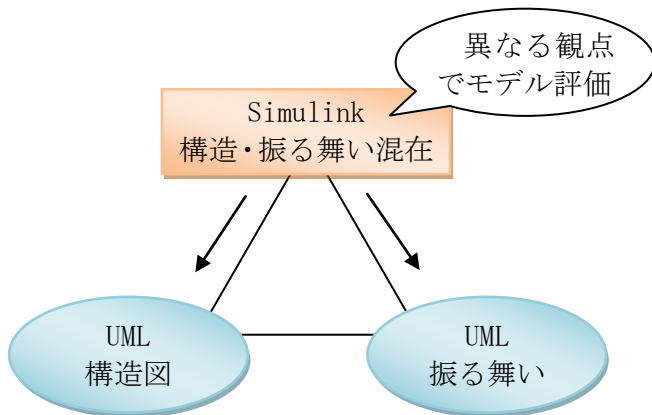


図 3.1. Simulink モデルのビューの設定

#### 3.2. K-means 法によるサブシステム化

Simulink モデルの抽象化から、属人性を排除するために K-means 法というクラスタリング手法を用いる。本研究では K-means 法を Simulink モデルに対応するために、最初にクラスタを割り当てる代表点を複数設ける。代表点は Sinks block・Signal routing block・四則演算ブロック・類似するブロック構造の 4 つを挙げる。そしてクラスタリングを行い、一定以上のクラスタの集合をサブシステムとする。この結果、属人性を排除し、体系的なサブシステム化を行うことができる。

#### 3.3. モデル間の双方向変換の実現

Simulink と UML をモデル変換しながらソフトウェア開発を行う場合、モデル間のトレーサビリティ維持が重要となる。よってモデル変換では片方向の変

換ではなく、双方向の変換が必要である。

また、モデル変換を手で行うと、誤りが生じる可能性がある。さらに人手によるコストもかかってしまう。そこで、モデル変換を自動化することにより、人手による誤りを無くし、トレーサビリティを維持する。

モデル間の双方向変換の自動化を実現させるために、NII の Hu らによって開発された GroundTram を用いる。GroundTram とは双方向変換言語 UnQL+を提案・実装しマッピングルールに従って変換を行うツールである。これを用いて Simulink モデルと UML モデルの対応関係を定義し、変換を行うものとする。これによって Simulink と UML の双方向変換の自動化が実現し、モデル間のトレーサビリティを保つ。

### 4. ケーススタディ

本手法を倒立振子のコントローラモデルに適用した。Simulink モデルを UML クラス図に変換する事で、責務が集中しているサブシステムが判明し、クラス図に修正を加える事ができた。また、その修正結果を自動で Simulink モデルに反映しトレーサビリティを維持できた。このことから本研究を用いることで Simulink モデルの保守性を向上できたと言える。

### 5. 関連研究

L. B. Brisolarara らは Simulink モデルを構造面と振る舞い面から捉えた。これらを分けることの重要性を述べ、複合構造図とシーケンス図への変換を行なっている。[3]しかし、サブシステム以外のブロックの扱いは記述されておらず、変換も Simulink から UML の片方向の提案のみとなっている。我々はモデル間の抽象度を考慮し、サブシステム以外のブロックも変換の対象として扱い、また双方向変換の自動化を行っている点が異なる。

### 6. おわりに

本研究では Simulink モデルの保守性を向上させることを目的として、Simulink モデルにクラスタリング手法を適用し、適切な抽象化を行った上で UML モデルとの変換を行った。

今後は他のクラスタリング手法の適用によるより適切な抽象化や他の UML モデルとの変換による新たなモデルビューを検討し、Simulink モデルの保守性向上につなげたい。

### 文 献

- [1] UML,  
<http://www.uml.org/>
- [2] Brisolarara, L., Becker, L., Carro, L., Wagner, F., Pereira, C. E., Reis, R., "Comparing High-level Modeling Approaches for Embedded System Design", ASPDAC, no8471640, pp986-989, jan2005
- [3] Carl-Johan Sjöstedt1, Jianlin Shi, Martin Törngren, David Servat, DeJiu Chen, Viktor Ahlsten, Henrik Lönn, "Mapping Simulink to UML in the design of embedded systems: Investigating scenarios and transformations", In OMER 4 Post Workshop Proceedings, 2008