

## 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

小澤 貴之<sup>†</sup> 鷲崎 弘宜<sup>†</sup> 深澤 良彰<sup>†</sup>

Simulink モデルは組み込みソフトウェア開発に適している。一方、大規模で煩雑なモデルになりやすく検討・修正が困難である。サブシステムによるモデルを抽象化は属人性が強いためにモデルの理解性が低い。そこで、我々は Simulink モデルの保守性の向上を目指して、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 モデルと UML モデルの双方向変換を提案する。UML モデルの静的・動的なモデルに変換する事で、責務等を明らかにして理解性を向上させる。及び、双方向変換でモデル間の対応関係を維持する。

## Research on the clustering towards the improvement in conservativeness of a Simulink model, and bidirectional conversion with an UML model

Takayuki Ozawa,<sup>†</sup> Hironori Washizaki<sup>†</sup>, Yoshiaki Fukazawa<sup>†</sup>, Zhenjiang Hu<sup>†</sup> and Soichiro Hidaka<sup>†</sup>

Simulink model is suitable for embedded software development. On the other hand, it is difficult to fix, often tend to be large and complicated model. Understandability of the model is low abstraction is dependent on developers by subsystem, the model. Therefore, we conducted an appropriate abstraction aims to improve the maintainability of the Simulink model to apply the clustering method does not depend on the developer. In addition, we propose a two-way conversion of Simulink models and UML models. By converting to static and dynamic model of the UML model, to improve the understanding and clarifying the responsibilities. And to maintain the correspondence between the model in two-way conversion.

### 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

組み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

ってしまう。つまり、このサブシステムによるモジュール抽象化は属人的・非体系的であると言える。

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 モデルを UML モデルの静的観点・動的観定の異なる観点で捉えたモデルに変換する事で、明確でなかった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. モデルベース開発の問題点

#### 2.1. UML

UML(Unified Modeling Language:UML, 統一モデリング言語)[2]は、現在ソフトウェア開発において最も普

<sup>†</sup>早稲田大学理工学術院 基幹理工学研究科  
Graduate school of Fundamental Science and Engineering,  
Faculty of Science and Engineering, Waseda University.

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図と動的な振る舞いの2種類に分けることができる。

システムを静的な構造と動的な振る舞いの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

Simulink モデルの問題点としてモデルが大規模であるため、責務の把握や実現手段の検討・修正が困難な点が挙げられる。

また、Simulink モデルを抽象化する手法としてサブシステム化がある。しかし、サブシステムの作成は開発者に大きく依存してしまうため、Simulink モデルの抽象化・モジュール化は属人的・非体系的であるという問題がある。

さらに、本手法では Simulink モデルを抽象度の高い、UML モデルに変換するが変換後、検討・修正した内容を元の Simulink モデルへ反映させる難しさが問題として残る。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。

## 3. Simulink モデルのモデルビュー提案

### 3.1. 提案手法の全体像

P1 に対して、本研究では責務を把握可能な抽象化をすることで解決する。そのために UML モデルを用いる。UML モデルに変換する事で、異なる観点でのモデル評価と整理することができ、Simulink モデルの理解性が高め、保守性が向上させる。

UML に変換する前の Simulink モデルをサブシステム化し、適切な抽象化を行う必要がある。そのため、サブシステムをルール化する事により属人性を排除し、体系的な抽象化を行う事で、P2 の問題を解決する。

最後に Simulink モデルと UML モデル間の変換を自動双方向変換にすることにより、モデル間のトレーサビリティを維持し、P3 を解決しモデルの保守性を保つ。

### 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、モデル変換を人手で行うと、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[3]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。GroundTram を用いてモデル変換できるように 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. 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や他の UML モデルとの変換手法を検討したいと考える。そして、Simulink モデルをより多くの観点でモデル評価を行い、Simulink モデルの保守性向上につなげたいと考える。

## 参考文献

- [1] TheSimulink, <http://www.mathworks.com/products/Simulink>
- [2] UML, <http://www.uml.org/>
- [3] GroundTram, <http://www.biglab.org/>

## 謝辞

本研究に際して、鷺崎弘宜准教授、深澤良彰教授、国立情報学研究所の胡振江教授、日高宗一郎助教から、丁寧かつ熱心なご指導を賜りました。ここに感謝の意を表します。