

超分散システム制御ネットワークシミュレータ用モデル検証のための適合性試験法 -モデルの実時間制約のための仕様記述法と適合性試験法-

北海道大学 ○魚住 一 金井 理 岸浪建史
モトローラ 上広 清 山元 進
旭川工業高等専門学校 戸村 豊明

要旨

現在、超分散システム制御シミュレーションでは、制御デバイスモデルの実時間特性は状態遷移仕様に記述されていない。従って、実時間特性は評価されていない。そこで有限状態機械により表現された制御デバイスモデルの状態遷移仕様内に、実時間制約が規定できる様拡張すると共に、実時間制約に対する仕様と制御デバイスモデルの適合性を検証することが可能なシミュレーションメカニズムを提案する。

1. はじめに

ネットワークを介して 1 万個以上の制御対象を制御する超分散システムが FA、BA (BuildingAutomation) 等で普及し始め、そのシステムの制御性能の事前予測を目的としたシミュレーションに対する要求が高まっている。シミュレーションの高精度化には、シミュレーションモデルの仕様と実装コードとの一致性を保証する必要がある。そこで本研究では、StateChart[1]で記述された超分散システム制御シミュレーションの制御デバイスモデルの挙動を表す状態遷移仕様と、仕様から作成されたモデルの Java 実装コードとの一致性を検証する適合性試験法について提案してきた[2]。しかし従来の研究では StateChart に状態遷移挙動のみが記述され、シミュレーションで必要となる状態遷移の実時間制約までは仕様に記述していなかった。そこで、本報では、状態遷移仕様の実時間制約を仕様に記述する手順と、実時間制約に関する仕様と実装コードとの適合性試験法を提案する。

2. ネットワークシミュレーション用モデル検証のための適合性試験法

ネットワークシミュレーション用モデル検証のための適合性試験法の位置付けを図 1 に示す。実時間制約が記述されていない StateChart で与えられる仕様に、時間制約の追加規則に従い、実時間制約を StateChart に組み込む。実時間制約が追加された StateChart で与えられる仕様を基に開発された制御対象モデルの

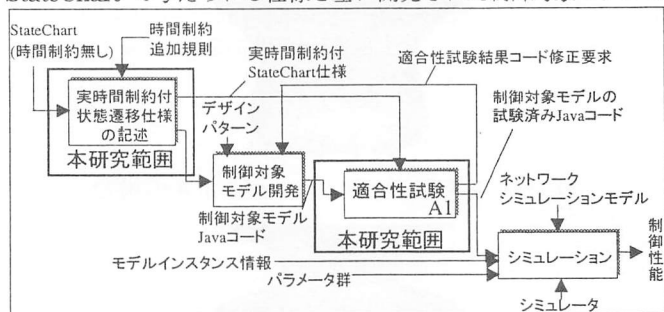


図 1: 適合性試験の位置付け

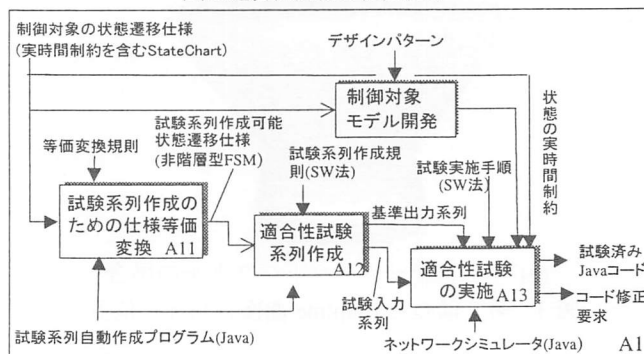


図 2: 適合性試験の概要

Java コードに対し、仕様から作成された試験系列を用いることにより仕様が定める状態と状態遷移の確認を行い、同時に実時間制約の試験を行う。試験を通過したモデルの Java コードがシミュレーションに用いられる。従来、通信制御ソフトウェアに対する適合性試験法は、各種提案されている[3]。しかし、通信制御ソフトウェアの仕様記述は、有限状態機械(FSM)に基づいているため、階層型有限状態機械である StateChart に対する適合性試験法として直接用いることができない。そこで本研究では、図 2 に示す様に StateChart を階層性の無い FSM に変換し、試験を行う手法を採用する。

3. StateChart と FSM の定義

StateChart(SC)は式(1)で定義される。

$$SC = \langle S_0, S', X, Z, \delta', \omega, \sigma \rangle \quad (1)$$

ここで S_0 は各状態階層における初期状態の集合、 S' は状態集合、 X は入力記号の集合、 Z は出力記号の集合、 $\delta': X \times S' \rightarrow S'$ は状態遷移関数、 $\omega': X \times S' \rightarrow Z$ は出力関数、 $\sigma: S' \rightarrow 2^S$ (但し、 2^S は S' のべき集合)はある状態に含まれる子状態集合を与える関数である。

一方 FSM は次の(2)式で定義される。

$$FSM = \langle s_0, S, X, Z, \delta, \omega \rangle \quad (2)$$

$s_0 (\in S)$ は初期状態、 S は状態集合、 X ならびに Z は StateChart と同一のものを用いる。 δ は状態遷移関数、 ω は出力関数である。

4. 状態遷移における実時間制約の記述と方法

本研究で対象としているネットワークシミュレータの制御デバイスモデルの状態遷移仕様では、一定時間の経過後に内部イベントが発生し、状態が遷移する形の時間制約が多いため、本研究ではこの実時間制約のみを仕様記述と試験の対象とする。図 4 は、状態 Heartbeat Starting に遷移した直後から T1 時間後に、内部イベント $tm(T1)$ によって状態 Heartbeating に遷移する部分を含む状態遷移仕様である。一般にこのような実時間制約を記述するには、StateChart 以外に TimingDiagram を併用して実時間制約の記述を行い、その後 StateChart に変換するのが、最も実時間制約の記述に適しているとされている[4]。そこで、本報では TimingDiagram を併用して StateChart の状態遷移仕様に実時間制約を追加する方法を提案する。

4.1 TimingDiagram

TimingDiagram は、時間を横軸、状態を縦軸にすることによって、複雑な状態遷移の時間的順序関係を簡明に記述することができる。図 4 の StateChart の時間による状態遷移を TimingDiagram として表すと、図 5 の様に表現可能となる。

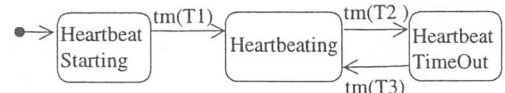


図 4: 実時間制約が記述された StateChart

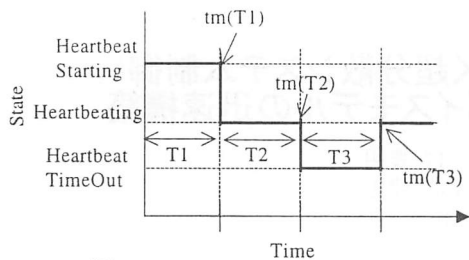


図5:TimingDiagram

4.2 時間制約の記述法

以下に示す①～④の手順に従い TimingDiagram(以下 TD)を用いて、実時間制約を StateChart に追加記述する方法を述べる(図6)。

- ①状態遷移要求条件から時間制約の無い StateChart を作成する。
- ②①で得た StateChart の状態を TD の縦軸とし、状態遷移仕様の部分的な状態遷移の経路順に従って時間制約無し TD を作成する。
- ③②の TD の横軸に、状態に停留する時間を横軸の長さとして TD 上に表現した時間制約付 TD を作成する。
- ④③の TD に基づき状態の停留時間をパラメータとして tm イベントとし、その状態をソースとする遷移上に tm イベントを追加し、実時間制約付 StateChart を作成する。

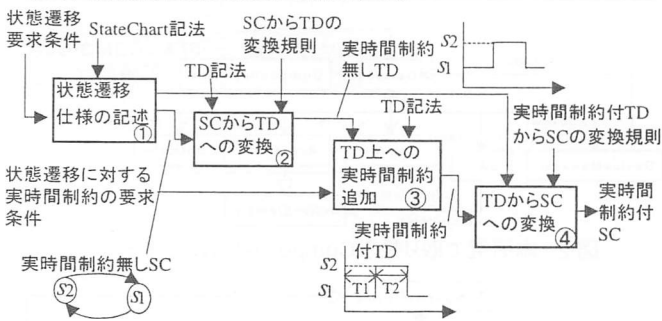


図6:実時間制約付 StateChart の作成法

5. 仕様の等価変換

提案する制御対象シミュレーションモデルの実装コードに対する適合性試験は、図2のように行う。以下の手順に従い、従来の適合性試験法が実施可能な仕様に変換し、適合性試験を行う。

- ①並行サブ状態を含む CompositeState を等価な FSM へ変換する。
- ②階層性のある StateChart を階層性の無い等価な FSM へ変換し、更に適合性試験が実施できる条件の FSM へ変換する。
- ③従来の試験系列作成法を用いて、適合性試験が実施できる FSM の仕様から試験入力系列と基準出力系列の作成を行う。
- ④得られた試験入力系列を、仕様から別途実装された制御対象のシミュレーションモデル Java 実装コードに入力し、得られた実出力系列と基準出力系列の比較を行い、実装コードと状態遷移仕様の適合性の判定を行う。

5.1 CompositeState から等価 FSM への変換規則

図6のように親状態 S_a が点線で仕切られた複数の並行サブ状態 $c_1 \sim c_n$ を持つ場合を CompositeState と呼ぶ。 S_a 内では、 $c_1 \sim c_n$ の並行サブ状態が独立かつ並行に動作可能であることを表す。以下に示す並行サブ状態の等価変換規則によって、並行サブ状態を持つ CompositeState を FSM へ変換する[5]。

- ① $S_a, c_1, c_2, \dots, c_n, s_{11}, \dots, s_{nk_n} \in S', \sigma(S_a) = \{c_1, c_2, \dots, c_n\}$ (c_1, c_2, \dots, c_n は S_a の並行サブ状態、 S_a は CompositeState である)、 $\sigma(c_i) = \{s_{i1}, s_{i2}, \dots, s_{iki}\}$ として、状態の直積集合 $\sigma(c_1) \times \sigma(c_2) \times \dots \times \sigma(c_n)$ を求める。
- ②①で求めた直積集合の要素を新たな状態とし、状態遷移、出力を分解前の各並行サブ状態内の状態遷移、出力に基づき求める。上記の変換規則に従って、並行サブ状態の FSM への等価変換を行う。

5.2 StateChart から FSM への変換規則

式(1)で与えられる階層性を持つ StateChart は、等価な挙動を持つ FSM へ変換可能である。以下にその変換規則を記す。

- ①複数の子状態を持つ状態を始点とする状態遷移は、その全ての子状態を始点とする複数本の状態遷移へ変換する。
- ②複数の子状態をもつ状態を終点とする状態遷移は、下位階層の初記状態への遷移に変換する。
- ③最上位階層の状態を消去する。
- ④①～③を階層性が無くなるまで繰り返す(図7)。

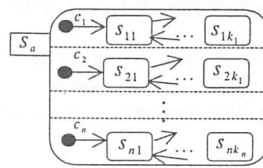


図6:CompositeState

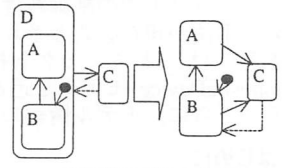


図7:階層性の分解

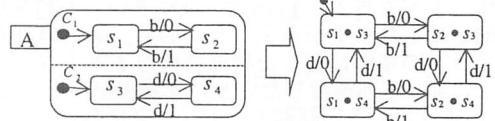


図8:CompositeStateの等価なFSMへの変換

5.3 適合性試験実施条件を満たす FSM への変換

適合性試験が実施可能であるには、FSM が①非冗長条件、②強連結条件、③完全定義条件の3条件を満たすことが必要である。一般の状態遷移仕様の FSM は上記②③の条件を満たしていないのが普通である。そこで②③の条件を満たす様に未定義部分等の追加を行い、適合性試験が可能な状態遷移仕様へ変換する[5]。

6. 適合性試験法と適合性試験の実施

適合性試験では、仕様が規定する状態と遷移が実装コード内に存在することを確認するために、仕様に基づいて入力系列と基準出力系列の作成を行う。試験系列作成手法は多種あるが、試験系列の存在保証性、試験系列の短かさという特徴から、本研究では SW 試験法[3]を採用した。SW 試験法は状態存在の確認に任意の2状態を判別するための入力系列の集合である W 集合[3]を用いている。

実時間制約が記述されていない制御対象デバイスモデルの Java 実装コードに対しては、上記の適合性試験法をそのまま実施し、仕様との一致性を検証する。一方、実時間制約が課された仕様に対する適合性試験は、ネットワークシミュレータに規定された時間ごとに、擬似的に内部イベントを発生させ、その出力を観測することによって仕様が定義する状態と遷移、のみならず状態に滞留している時間までが制御対象のシミュレーションモデル Java 実装コードに実装されているか否かの検証を行う必要がある。ただし、内部イベントによる遷移には、試験用の出力が各遷移に対して設定されること、出力は各遷移に対して固有の出力を持つことを仮定する。

6. まとめ

実時間制約が記述されていない StateChart に TimingDiagram を併用することによって、実時間制約を StateChart に追加記述する方法を提案した。また実時間制約が付加された状態遷移仕様の Java 実装コードに対して適合性試験法を実施する方法を提案した。

参考文献

- [1]OMG: "Unified Modeling Language Specification Ver1.3" June(1999), pp131-149
- [2]魚住他: "プラント制御対象モデルにおける状態遷移仕様の適合性試験に関する研究": 2000年度精密工学会春季大会学術講演論文集, p72
- [3]佐藤文明他: "有限オートマトンに基づくシステムの試験系列自動作成手法の提案": 電子情報通信学会論文誌(B-1), (1989) Vol. J72-B-1, No.3, pp.183-192
- [4]Bruce Powel Douglass: "Real-time UML", (1998), pp187-190
- [5]魚住他: "超分散システム制御ネットワークシミュレータ用モデル検証のための適合性試験法", 2000年度精密工学会秋季大会学術講演論文集