プロセスの模倣性判定に基づくUML設計間の整合性検証

横川 智教[†1] 宮崎 仁[†2]
佐藤 洋一郎[†1] 有本 和民[†1]

これまでに著者らは，プロセスの模倣性判定を利用して，UMLのシーケンス図を対象とした詳細化関係の検証手法を提案してきた．本稿では，この手法を状態マシン図へと拡張し，状態マシン図をプロセスとして表現することで，状態マシン図とシーケンス図に対する整合性検証をプロセスの模倣性判定として実現する．

Consistency verification of UML diagrams based on weak simulation

Tomoyuki YOKOGAWA ,[†1] Hisashi MIYAZAKI ,[†2]
Yoichiro SATO [†1] and Kazutami ARIMOTO [†1]

We had proposed the method for refinement check for UML sequence diagrams. In this method, refinement of sequence diagrams is represented as weak simulation of processes obtained from the sequence diagrams. In this paper, we propose the method for verifying consistency of a sequence diagram and state machine diagrams. We represent state machine diagrams and a sequence diagram as processes and verify the consistency by checking weak simulation of the processes.

1. まえがき

UMLによるソフトウェアシステムの設計では，モジュール間で行われるべきやり取りをシーケンス図によって記述し，それを満たすようモジュールの動作を状態マシン図として設計する．ここで状態マシン図がシーケンス図の振る舞いを満たすこと，すなわち整合性を確認する必要があるが，この確認作業を人手で行うことは，シーケンス図や状態マシン図の全ての振る舞いを考慮する必要があり，非常に困難である．

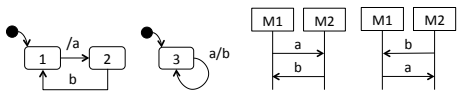
筆者らは，あるシーケンス図が別のシーケンス図の振る舞いを満たすことを，プロセスの弱模倣性に基づいて検証する手法を提案している[1)]．この手法では，シーケンス図のメッセージ送受信をイベントとみなすことでその振る舞いをプロセスとして表現し，それらが弱模倣関係を満たすか否かをモデル検査ツールLTSAを用いて検証している．本稿では，状態マシン図の振る舞いをプロセスとして表現し，シーケンス図から求めたプロセスと弱模倣関係であるかを判定することで，整合性検証を実現する．

†1 岡山県立大学
Okayama Prefectural University
†2 川崎医療福祉大学
Kawasaki University of Medical Welfare

2. 整合性検証

2.1 状態マシン図とシーケンス図

状態マシン図は，状態とそれらを結ぶ遷移によって構成される．遷移には，トリガおよびアクションとしてメッセージがそれぞれ割り当てられる．遷移はトリガの受信により実行され，状態の変化とともにアクションが送信される．また，初期状態として1つの状態が定められる．シーケンス図は，モジュールに対応したライフラインと，それらの間のメッセージ通信によって構成される．メッセージの送受信処理をオカレンスとよび，オカレンスの順序関係はライフライン上の位置で表現される．図1(a)(b)に状態マシン図の例を，図1(c)(d)にシーケンス図の例を示す．



(a) M1 (b) M2 (c) SQ1 (d) SQ2

図1 状態マシン図とシーケンス図

状態マシン図とシーケンス図の振る舞いは，メッセージ処理系列(実行とよぶ)の集合として表すことがで

きる．図 1 の状態マシン図 $M1$ はメッセージ a を送信する遷移とメッセージ b を受信する遷移を繰り返し，$M2$ は a を受信して b を送信する遷移を繰り返す．よって，メッセージ m の送受信を $m!$ と $m?$ で表すと，得られる実行は $a!a?b!b?\cdots$ の繰り返しのみとなる．シーケンス図からも同様に実行を求めることができ，図 1(c) および (d) から得られる実行は $a!a?b!b?$ および $b!b?a!a?$ となる．

2.2 整合性の定義

本稿では，状態マシン図とシーケンス図の整合性を実行の包含関係として定義する．状態マシン図の実行は無限長の系列となるため，有限長のプレフィックスに対して，シーケンス図の実行と比較する．

定義 状態マシン図の実行の集合を C_{SM}，シーケンス図の実行の集合を C_{SQ} とする．すべての $q \in C_{SQ}$ に対して，$q \in Pre(r)$ となる $r \in C_{SM}$ が存在するとき，状態マシン図とシーケンス図は整合性を満たす．

ここで $Pre(r)$ は，実行 r のすべてのプレフィックスからなる集合である．図 1 の状態マシン図は，実行のプレフィックス $a!a?b!b?$ が $SQ1$ の実行と一致するので $SQ1$ との整合性は満たされるが，$SQ2$ との整合性は満たされない．

実行の包含関係はプロセスの弱模倣性判定に帰着して検証できる．LTSA は安全性として，プロセスが特性プロセスの動作に違反しないことを検証できる．ここで，プロセスの弱模倣性は安全性として LTSA で検証可能である[1]．よって，状態マシン図のプロセスを特性プロセスとしてシーケンス図のプロセスと並行合成することで，LTSA による整合性検証が実現できる．

3. プロセス表現

状態マシン図は，遷移によるメッセージ処理と状態の変化をローカルプロセスとして記述し，合成することでプロセスとして表現できる．例として，状態マシン図 $M1$,$M2$ は，以下のプロセス M1,M2 で表現できる．

```
M1 = S1,
S1 = (as->S2),
S2 = (br->S1).
M2 = S3,
S3 = (ar->bs->S3).
```

ここで，as と bs，ar と br はそれぞれメッセージ a と b の送受信を表すイベントである．メッセージの送受信に関する振る舞いは，以下のプロセスで表現される．

```
A = (as->ar->A).
B = (bs->br->B).
```

そして，それら全ての並行合成プロセス SM として，状態マシン図全体の振る舞いが表現できる．

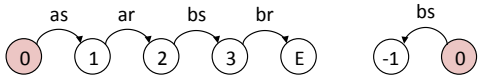
```
||SM = (M1||M2||A||B).
```

シーケンス図は，ライフライン上のオカレンスの順序関係をプロセスとして記述し，メッセージを表すプロセスと合成することでプロセスとして表現できる．例として，シーケンス図 $SQ1$,$SQ2$ は，以下のプロセス SQ1,SQ2 で表現できる．

```
SQ1M1 = (as->br->END).
SQ1M2 = (ar->bs->END).
||SQ1 = (SQ1M1||SQ1M2||A||B).
SQ2M1 = (br->as->END).
SQ2M2 = (bs->ar->END).
||SQ2 = (SQ2M1||SQ2M2||A||B).
```

最後に，以下のように，プロセス SM を特性プロセス SPEC とし，プロセス SQ1 および SQ2 とそれぞれ並行合成することで，整合性検証が可能となる．

```
property ||SPEC = (SM).
       ||Verify1 = (SPEC||SQ1).
       ||Verify2 = (SPEC||SQ2).
```



(a) Verify1　(b) Verify2

図 2 得られたプロセス

LTSA による検証の結果，図 2 に示すように Verify1 は終了状態 E へと正しく遷移したが，Verify2 は違反状態 -1 へと遷移し，整合性違反が検出できた．ここで，違反状態へ至る遷移がイベント bs をもつことから，図 1(d) における b の送信処理が状態マシン図の実行に含まれておらず，整合性違反の原因であることがわかる．このように，違反状態へ至る系列のイベントをもとに，違反箇所を特定することが可能である．

4. あとがき

本稿では，状態マシン図とシーケンス図をプロセス表現することで，整合性検証を行う手法を提案した．今後の課題として，階層構造をもつ状態マシン図をプロセス表現できるよう，提案法の拡張を行う．

参考文献

1) Miyazaki, H., Yokogawa, T., Amasaki, S., Asada, K. and Sato, Y.: Synthesis and Refinement Check of Sequence Diagrams., *IEICE Trans. on Inf. and Syst.*, Vol.E95-D, No.9, pp. 2193–2201 (2012).