0-008

機能仕様と振る舞い仕様間の整合性
Consistency between Functional and Behavioral Specifications

安田 佳宏† 新川 芳行†
Yoshihiro Yasuda Yoshiyuki Shinkawa

1. まえがき

UML2.0 ではシーケンス図に時間を含む制約の記述が可能となっているが，シーケンス図はシステムの動的な振る舞いしか記述できず，処理の意味的もしくは機能的側面を考慮した制約の記述は困難である．また、UML 自体にモデル検証能力がないため，作成したモデルがこれらの制約を満たすか否かは他のツール等により行う必要がある．本論文では，形式手法の一つである B メソッドにより記述されたクラスの仕様とシーケンス図のマッチングにより導出した，機能仕様および時間制約を含む時間オートマトンをモデル検査ツール UPPAAL により検証することで，モデルの機能面，振る舞い面，および時間制約という複数視点からの評価を可能とする手法を提案する．

2. B メソッドによる UML モデルの形式化手法

2.1 B メソッドの概要

B メソッドは，仕様記述からプログラムの導出までの一連の過程を支援する形式仕様記述言語である．B メソッドによる形式化では，仕様化→詳細化→実装化という段階的詳細化法に基づいて，抽象機械からいくつかの中間的な記述を経て，与えられた計算機環境で効率よく実行できるように内部構造を定めていく．

仕様化段階は，要求されるシステムの振る舞いの数学的なモデルである，抽象機械と呼ばれるコンポーネントで表現する．抽象機械から，いくつかの中間的な記述を経て，プログラムを生成できる詳細度の抽象機械を構成する実装段階へと詳細化を進める．実装段階のモデルはインプリメンテーションと呼ばれ，プログラムと同等の詳細度で記述されていなければならない．

2.2 抽象機械とインプリメンテーション

B メソッドでは，抽象機械を構成単位として仕様を記述する．抽象機械は，内部構造の詳細を捨象して表現したシステムであり，以下に示すような特徴を持つ．

- 内部状態を持つ
- オペレーションを外部に提供する
- オペレーションを通じて内部構造が変わる

図 1 では，抽象機械の基本要素を示す略図と B 抽象機械記法を用いて抽象機械を記述した例を示している．図 1 で記述した抽象機械は，名前が M であり，スカラー値のパラメタ p をとる抽象機械を記述した例を示している．CONSTRAINTS 節ではパラメタに対する制約条件を CN(p) と定義する．SETS 節，CONSTANTS 節，VARIABLES 節においてそれぞれ，集合 s，定数 c，変数 v を宣言している．集合と定数に関する制約条件は PROP(s, c)，変数に対する制約条件(以下，不変条件)を表現する INV(p, s, c, v)は，変数の型と変数が取り得る値の制限を宣言する述語である．不変条件は，抽象機械のオペレーションが実行されて変数の値が変化しても常に成り立たなければならない．

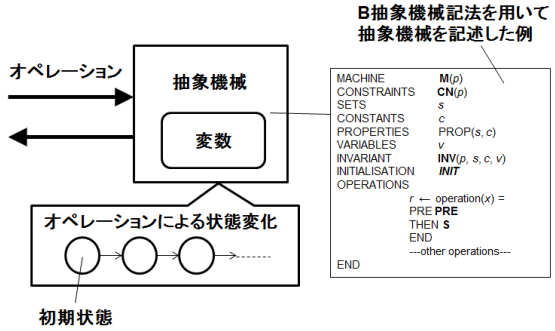

図 1: 抽象機械

また，インプリメンテーションは，抽象機械を計算機で実装可能なデータ表現と，決定的な状態変化によって実現したものである．

2.3 仕様の整合性検証と詳細化の正当性検証手法

抽象機械の仕様の整合性検証を行うためには，以下に示す 5 つの項目について証明を行う必要がある．[]内は図 1 の各節の述語論理式を用いた証明責務を表している．

- **パラメタ制約の無矛盾性**[∃p.CN]
  抽象機械のパラメタで，CONSTANTS 節の制約を満たすものが存在することの証明
- **集合と定数の制約の無矛盾性**[CN⇒∃(s, c).PROP]
  PROPERTIES 節の制約を満たす集合と定数が存在することの証明
- **不変条件の無矛盾性**[CN∧PROP⇒∃(v).INV]
  抽象機械の取り得る状態で，不変条件を満たすものが存在することの証明
- **初期化の整合性**[CN∧PROP⇒[*INIT*]INV]
  初期状態が不変条件を満たすことの証明
- **オペレーションの整合性**[CN∧PROP∧INV∧PRE⇒[S]INV]
  オペレーションが正しく起動されるならば，オペレーション実行後の状態が不変条件を満たすことの証明

リファインメントやインプリメンテーションのような中間的なモデルの場合でも同様に，整合していることの検証を行う．また，中間的なモデルの場合は，抽象機械などの上位仕様からの詳細化が正当であることの検証を行う必要がある．詳細化の正当性検証では，抽象機械の状態と中間的なモデルの状態の対応関係(リンク不変条件)に基づいて証明を行うことによってその詳細化が正当であったかの検証を行うことができる．

3. UML 間の相互関係と時間制約の記述・検証

3.1 UML シーケンス図における時間制約の記述

UML モデルの中で時間の制約を扱えるものにシーケンス図がある．シーケンス図では，タイミングの制約を記述することによって時間の制約を表現することができる．ま

† 龍谷大学，Ryukoku University

た，図 2 に示すように時間制約を持った実時間シーケンスモデルは時間オートマトンに変換することができる[2]．この変換によってモデル検査ツール UPPAAL を用いて，振る舞い仕様および時間制約の視点から検証を行うことが可能となる．

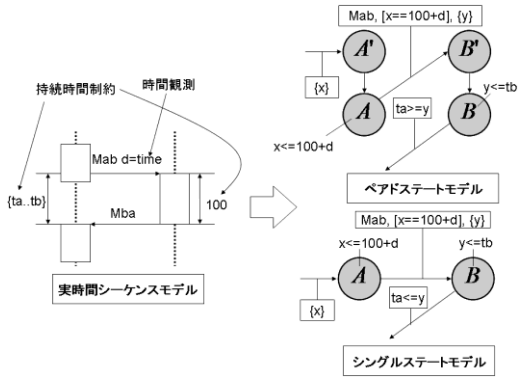

図 2: シーケンスモデルから時間オートマトンへの変換手法

3.2 シーケンス図とクラス図の相互関係

クラス図はオブジェクトを特定の基準によって分類し，共通要素を取り出して抽象化したクラス及び，それらの間の関係を表すものである．クラスには属性と操作があり，属性にはオブジェクトがもつ状態などをクラスやオブジェクトを特徴付けるデータの名前を記述し、操作にはオブジェクトによる処理動作を記述することができる．しかし，クラス図では，操作の流れやどのようなタイミングで呼び出されているか判断することが難しい．そこで，シーケンスモデルを作成することにより，システム内部のオブジェクトがどのようなメッセージを受けてどのように振る舞うのか明確になる．両者はそれぞれシステムの静的及び動的な側面を表し，性質が異なるため，独立して作成することが可能であるが，クラス図の操作がシーケンス図のメッセージ処理に対応すると考えることができる．

3.3 クラスモデルから抽象機械への変換

クラスモデルを具象化したものがオブジェクトである．一般的なシステムは，これらのオブジェクト同士が協調動作を行うことで動作する．本論文では，図 3 に示すようにクラスモデルと抽象機械間において 1 対 1 の対応付けを行う．

また，オブジェクトに対して，抽象機械とインプリメンテーションの対応付けを行っている(図 3)．クラスモデルから抽象機械への変換によって，クラス図の機能仕様に関する整合性の検証を行うことが可能となる．

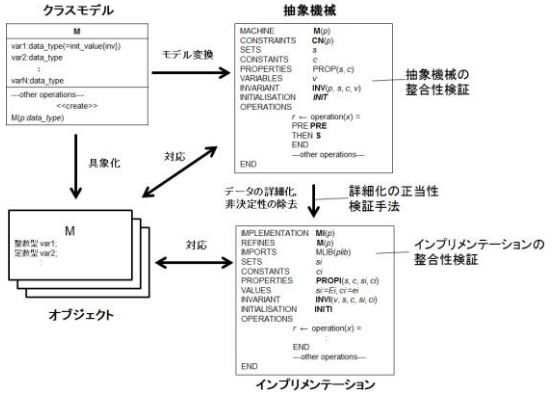

図 3:クラスモデルと抽象機械の変換例

3.4 シーケンスモデルと B メソッドモデルの対応付け

シーケンスモデルを B メソッドモデルへ変換した例を図 4 に示す．図 4(a)のシーケンスモデルは，3 つのクラス/オブジェクト間のメッセージのやり取りをモデル化したものである．A からのメッセージによって，B の操作を呼び出し，B からのメッセージによって，C の操作の呼び出しを行い，それぞれが操作後にメッセージによって値を返している．これを B メソッドモデルに変換したものが図 4(b)である．オブジェクト A，B，C をそれぞれ，インプリメンテーション A(以下，A)，ライブラリ機械 B(以下，B)，ライブラリ機械 C(以下，C)で置き換える．A は B を輸入(IMPORTS)しており，ライブ B は C を輸入しているものとする．ここでは，インプリメンテーションおよびライブラリが輸入したライブラリ機械のオペレーションによって値を参照・更新する．A は B のオペレーションによって，B の状態を変化させ，B は C のオペレーションによって C の状態を変化させる．その後，C，B それぞれのオペレーションからの出力値を出力パラメタ *pc*, *pb* として値を返している．

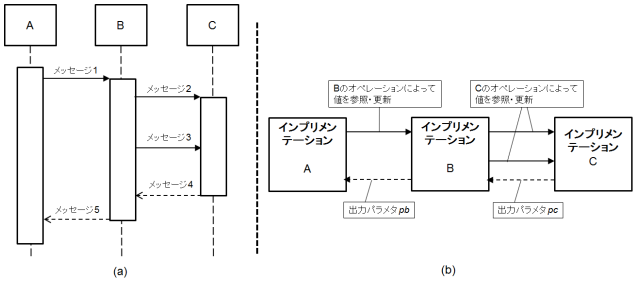

図 4: シーケンス図を B メソッドモデルへ変換した例

シーケンスモデルと B メソッドモデルを対応付けることによって，機能的仕様と振る舞い的仕様間の整合性の検証を行うことが可能となる．

4. 結び

本論文では，機能的側面，振る舞い的側的，時間制約を表現する UML モデルに対して，B メソッドと UPPAAL を補完的に用いて検証を行う方法について述べた．

これによって，機能，動作，時間制約というソフトウェアの 3 つの重要な品質特性の検証が可能となる．今後の課題として，時間制約を充足するコードの生成を行う必要がある．

参考文献

[1]来間 啓伸，”B メソッドによる形式仕様記述”，近代科学社(2007).
[2]安田 佳宏，”UML シーケンス図における時間制約の記述とその検証”，電子情報通信学会(2010).
[3]J.R.Abrial，”The B-Book: Assigning Programs to Meanings”, Cambridge University Press(2005).
[4]Theo Dimitrakos，“Compositional Structuring in the B-Method: A Logical Viewpoint of the Static Context”, International Conference of B and Z Users ZB2000 (2000)
[5]株式会社テクノロジックアート，“独習 UML 第 3 版”，翔泳社(2007).
[6]K.Guldstrand Larsen, principles of model checking, The MIT Press, 2008.
[7]中田明夫, 時間オートマトンのモデル検査, 電子情報通信学会論文誌, 2009, D Vol. J92-D No.5 pp.576-586.