工業高等専門学校の学生に対する 形式手法 B-Method の学生実験の実践

大西孝臣[†1]　吉村斎[†1]　阿部司[†1]　稲川清[†1]　山本椋太[†2]　堀武司[†3]

概要：本稿では，ソフトウェア技術者の素養としての形式手法の体験を目標に，工業高等専門学校（工業高専）の本科 4 年生のクラス全員に対して形式手法 B-Method の学生実験を実践した．当実験は，形式仕様記述・定理証明とモデル検査の双方を実施項目に含有しており，「フォーマルメソッド利用のレベル」におけるレベル 0 よりレベル 2 までの手順を一通り体験させることを目指す．結果，受講学生より，ツールの使用手順の理解度や B-Method への親近感について一定の評価を得ており，ツールを導入したいという希望が出ているなどの一定の関心を得ており，学生の形式手法に対する敷居を下げることに成功しており，目標を達成するものであり，人材の養成確保の問題解決に向けて寄与するものと考える．

A Practice of Student Experiments of the Formal Method, B-Method Applied for Students of Institute of Technology

TAKAOMI OHNISHI[†1]　HITOSHI YOSHIMURA[†1]　TSUKASA ABE[†1]
KIYOSHI INAGAWA[†1]　RYOTA YAMAMOTO[†2]　TAKESHI HORI[†3]

1. はじめに

ソフトウェアシステムの開発現場において，自然言語記述による仕様には曖昧な表現が含有されている理由から内容の矛盾や誤りの指摘が困難になる場合があるため，仕様の記述が厳密で曖昧性を排除できる形式手法が注目されている[1]．一方，形式手法の普及は進んでいるが初歩的な技術利用に留まる場合が多く，普及推進のための技術教育が求められており[2]，以前よりその導入コストの高さと並行して人材の養成確保の問題が課題に挙げられている[3]．

文献[1]によると，モデル検査ツールを使いこなすには数学や論理学の知識が必要になってくるが，そのつど勉強すれば良いとあり，文献[4]によると，形式手法の導入は簡単ではなく“難しい”が，初めから全てを先に学ぶ必要は無く，ツールの使い方の基礎を学習して仕様を記述し始める試みでも良く，“難しさ”とされるものの本質は形式手法の使用の是非とは無関係なところにあるとの立場もある．

海外においては形式手法の教育に関する研究は多数存在しており[5]，例を挙げると，米国には大学学部レベルや高校学校・中学校レベルも含め，ソフトウェア科学の教育課程に数学的論理思考（mathematical reasoning）の導入を試みるコミュニティが存在する[6]．

国内における形式手法の教育としては，主に企業の若手エンジニア向けの取り組みとして，国立情報学研究所のトップエスイープロジェクト[7]が 2004 年より 10 年超の実績[8]を挙げており，大学と共同した普及活動も目指している．

大学における教育としては，情報処理学会による，大学の情報系学科・コースの学士課程の教育における情報学分野の参照基準であるカリキュラム標準 J17-SE[9]において，形式手法は学習対象たる知識体系を構築している項目である BOK(Body of Knowledge)として位置付けられているのと同時に，情報基礎科目の上位としてソフトウェアエンジニアリング技術を扱う SE 科目として位置付けられている．

大学の情報系学科・コースと同じく工学系の高等教育機関であり，筆者大西らが所属する工業高等専門学校（以下，工業高専と呼ぶ）の情報系の学科・コースにおいても，情報技術教育の裾野を広げるべく形式手法の導入を検討している．しかし，冒頭で述べた通り，形式手法には人材の養成確保の問題があり，その上，企業や大学などでの導入事例と比較して，工業高専において形式手法を導入する際には，以下の考慮すべき問題が存在する．

- 情報系においてもハードウェア分野の教授にも時間を使っているため[10]，形式手法の教授に費やすための講義・実験・実習そのものの時間数が不足している．
- 教養系にて大学教養レベルの数理論理学を教授しておらず[11]，専門系でも教授する時間が少なく[10]，学生は，形式手法に触れる時点で初見に近い状態である．
- 講義・実験・実習の時間数に収まる規模で，かつ，ソフトウェアシステムの開発現場で活かせる内容を備えた題材が普及しておらず，実践例も少ない．

上記の問題については，工業高専の学生（以下，工業高

†1 苫小牧工業高等専門学校
National Institute of Technology, Tomakomai College
†2 名古屋大学
Nagoya University
†3 北海道立総合研究機構
Hokkaido Research Organization

専生と呼ぶ）に導入しようとする工夫や試みがなされていなかったところにも起因するものと考える．また，前述の文献[1]や文献[4]により見る立場を踏まえれば，まずツールに触れさせるアプローチにて工業高専生への形式手法の導入を試みるべきであり，プログラミング言語での実習を行っている時分の工業高専生に対して，形式手法のツールに触れさせることを通じて「ソフトウェア技術者の素養としての形式手法を体験させる」ところに学生の将来にとっての意義があるものと考える．ただし，時間数の制約や工業高専生のスキルレベルを踏まえ，本稿において筆者らが考える「ソフトウェア技術者の素養」の核として，形式手法のツールを活用した数学的論理思考の導入により「無欠陥のコード生成に寄与することを知ること」に据えたい．

工業高専における形式手法の教育の実践については，本稿と同時期である今年度よりの工業高専専攻科生向けの選択科目の講義の例[12]があるが，工業高専本科生のクラス全員に対する必修での実践例は皆無である．

本稿では，形式手法における人材の養成確保の問題への解決策の 1 つとして普及推進の技術教育の裾野を広げるべく，筆者大西らが所属している苫小牧高専の本科 4 年生（以下，学生と呼ぶ）のクラス全員に対して，形式手法 B-Method（以下，B と呼ぶ）を手段として，「ソフトウェア技術者の素養としての形式手法の体験」を目標に，形式仕様記述・定理証明とモデル検査の双方を実施項目に含有する学生実験を実践した．

2. 学生実験の実施

2.1 形式手法 B-Method について

種々ある形式手法の中より，本稿における実験においては B を採用した．B は Z 記法を含有する，VDM と同様のモデル指向型あるいはモデル規範型（model oriented）の形式手法である[13] [14]（さらに 2.3 節後半の，Atelier B におけるモデル記述の特徴についても参照）．

苫小牧高専において B を選択した背景として，筆者らが産業界よりの知見を得た経験がある．筆者大西らは平成 26 年度よりの 3 年間，経済産業省北海道経済産業局の戦略的基盤技術高度化支援事業『農業機械のさらなる高度化と海外進出に資する次世代電子制御ソフトウェア基盤の開発』[15]に参画しており，当事業を通じて B に関する，工業高専としては得難い知見を蓄積することができた．

B は文献[16]で言及されている「フォーマルメソッド利用のレベル」のレベル 2 でのシステム開発の成功事例の実績が存在しており[17]，ソフトウェアシステムの開発現場に適用できるレベルのモデル検証支援ツールを無償で入手できる．そのため，実践的な環境を無償で構築でき，学生実験への適用に向いている．同一モデルにて形式仕様記述・定理証明とモデル検査の双方を実施できるツールが[18] [19]存在する（2.3 節参照）．そのため，これらの双方を実験項目に含有することができることとなり，「ソフトウェア技術者の素養としての形式手法の体験」という実験の目標達成の手段に適している．

B を手段とする実験の実践に向け，筆者大西と堀は，これまで工業高専生に向けた実験に導入するための題材探しを行い，B によって記述したモデルを提案してきており[20] [21]，このモデル記述の際に求められる数理論理学の指導を含めて実験の時間内に収める目途を立てた．

当実験の実践においては，対象とするモデルの規模は小さいが，文献[16]で言及されている「フォーマルメソッド利用のレベル」におけるレベル 0「形式仕様記述」，レベル 1「形式的開発および検証」，レベル 2「機械支援による証明」を一通り体験させることを目指す．

2.2 実験を受講する学生について

実験においては，約 10 名を 1 班とする学生を相手にする．クラス全員を対象にし，本実験を除く形式手法の教授の機会は無いため，受講者の内の約 9 割にとっては，当実験は形式手法に接することのできる唯一の機会となる．

学生は，実験受講前までに，教養系にて高校数学レベルの集合と命題の関係を学んでいる．専門系の座学にて本実験の実施と同じ学期に同時進行の形で，集合・写像，命題論理・述語論理・正規言語・文脈自由言語の基本概念やソフトウェア開発の手順を学んでおり，本実験の受講が先行する学生が存在する．本実験よりも後の学期の演習・実験において，（形式的手法を伴わない）UML を活用したモデリングやプロジェクト形式のソフトウェア開発を学ぶ．

2.3 実験環境について

学生用 PC は CPU:Intel Core2Duo(3.00GHz), RAM:4.00GB, OS:Windows10Professional を 1 名につき 1 台与える．学生用 PC においては，2.5 節で述べる実験指導書などを PDF ファイルにて全て閲覧できるようにした．

B のツールとしては，B のモデル記述支援および定理証明ツールである仏国 ClearSy 社の Atelier B Ver.4.5.1[18]，および，独国ハインリッヒ・ハイネ大学が開発した B 記述のモデル検査ツールである ProB Ver.1.8.0[19]を用いる．

Atelier B におけるモデル記述においては，図 1 に示すように， 上位の抽象機械コンポーネントにおいては，要件・要求に対する分析・定義や基本設計の段階で仕様として詳細に規定することを避けるために集合や非決定的代入などの記法が用意されており，リファインメント（詳細化）[16] [17]と呼ぶ段階的変更を経て，下位の実装コンポーネントにおいては決定的で実行可能なプログラム相当の記述へと到達させる．B には，ソフトウェアシステムの開発に際して，このリファインメントの過程を経ることにより，モデル記述におけるループ構造や抽象的データなどを実装段階相当へと到達させる機能が用意されている[14] [22].

2.4 実験の構成・達成目標の設定について

実験は 3 時限（実質 140 分）×2 回で行う．

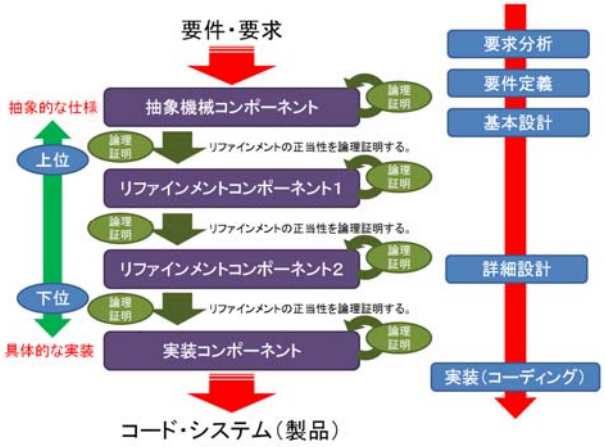

図 1　Atelier B におけるリファインメント

1 回目（第 1 週目）の実験（以下，実験 1 と呼ぶ）では静的モデルの記述をテーマとして，文献[23] にある演繹論理パズルに対する形式モデルを記述させて，ProB の制約充足ソルバ機能用いてパズルの求解をさせる．

2 回目（第 2 週目）の実験（以下，実験 2 と呼ぶ）では，実験時間の制約や状態爆発問題の回避のため，高々4 つの状態に遷移する小さな動的モデルの記述をテーマとして，Atelier B におけるリファインメントおよび定理証明，ProB におけるアニメーション[19] [24]，LTL 式を用いたモデル検査を行わせる．Atelier B におけるリファインメントと定理証明の過程を通じて，数学的論理思考の導入による「無欠陥のコード生成に寄与することを知ること」を教授する．

「ソフトウェア技術者の素養としての形式手法の体験」という目標を達成すべく，実験 1，実験 2 のいずれにおいても，「フォーマルメソッド利用のレベル」[16]におけるレベル 0 から 2 までの内容の実施を目指すために，次の工夫をした．

実験 1 においてはレベル 0 とレベル 1 を重視する．仕様が正しく記述されていれば，定理証明において，自動証明による検証を通過することでき，制約充足ソルバが唯一解を導出可能になるよう，題材を採択して実験設計をした．

実験 2 においてはレベル 1 とレベル 2 を重視する．比較的規模は小さいが冗長な仕様をリファインメントによって洗練させ，リファインメントの前後の記述に矛盾が無いことを定理証明で検証させられるよう，実験設計をした．ただし，レベル 1 においては「詳細化により仕様からプログラムを作成する」とあるが，実験時間の制約より，プログラムの自動生成を行わず，プログラムを生成する直前の段階までを実施することとした．

これまでに工業高専における形式手法の教授の実績などが無く，カリキュラム標準 J17-SE[9]の形式手法の学習目標・発展学習目標よりは当実験の目標である「ソフトウェア技術者の素養としての形式手法の体験」に見合う形式手法の教育指標を見つけられず，本稿における実験の実践を経るまでは暗中模索の状況であったため，当実験においては，3.1 節の学生アンケートの表 3，表 4 にある「形式仕様記述対象における，明文化された構成要素を定義できる.」などの，実験 1 においては 13 項目，実験 2 においては 15 項目の達成目標を独自に設けた．

2.5 学生への実験指導書などの配布

実験を受講する学生には，以下の実験指導書他の文書を実験前の自学習用に印刷物として配布した．

(1) **実験指導書ー形式手法ー**
実験手順の他に，ソフトウェアシステム開発における形式手法の特徴[25]，形式手法に基づく種々の手法の中での B の特徴[16] [17] [25]，B によるモデリングの特徴であるリファインメント[16] [17]について載せた．

(2) **実験指導書別冊**
各実験班で共通とする例題について，問題文，学生によるモデルの記述の発散を防ぐための集合名・定数名/変数名・変数の初期設定・操作名などの問題に対する制約，学生に完成させるために未完成のままにした写像・関数の対応図や状態遷移表を載せた．

(3) **B-Method の文法についてーAtelier B 編ー**
B の文法の説明の他に，B によるモデル記述の際に求められる写像・関数，述語（全般的な説明/不変条件），代入（決定的/非決定的），操作に係る説明を載せた．

以下の PDF ファイルを，学生用 PC にて閲覧できるようにした．

(4) **Atelier B の操作について（Atelier B 操作手順書）**
プロジェクト・コンポーネントの作成・編集，自動証明/対話証明の手順を載せた．

(5) **ProB の操作について（ProB 操作手順書）**
静的モデルに対する制約充足ソルバ機能を用いた演繹論理パズルの解導出までの手順，小さな動的モデルに対するアニメーションの実行の手順，LTL 式の書式の説明，LTL 式を用いたモデル検査の手順，LTL 式を用いた状態パスの探索の手順を載せた．

(6) **B-Method 文法簡易一覧**
各文法の概要の他に，使用される記号の ASCII 表示，JIS 表示，数学表示（Unicode の符号位置）を記載した．

(7) **その他，追加説明書**
実験 1 において，当初，例題のモデルにある記述のみの提示では，関数・逆関数を複合で参照を行う際にマップレット（|->）を使用できないという B の文法を学生が読み取れなかったため，追加の説明資料を作成した（3.2 節(2)参照）．

(8) **課題（実験 1・実験 2）**
各実験班で異なる課題を，完成したモデル記述を除い

・４名の人物イ，ロ，ハ，ニが，ａ歳，ｂ歳，ｃ歳，ｄ歳と全員が異なる年齢（順不同）である．これらの人物は，それぞれ，とある物を１つ以上（自然数）所有している．
・ｂ歳の人物とロの２名で，計４つ所有している．
・ｃ歳の人物とハの２名で，計３つ所有している．
・ａ歳の人物とニの２名で，計３つ所有している．
・ｃ歳の人物とイの２名で，計４つ所有している．
・イ，ロ，ハ，ニの４名で，計８つ所有している．
◆上記の「人物－年齢－物の所有数の関係」を具体的に述べよ．
（答え：イ-ｄ-３，ロ-ａ-２，ハ-ｂ-２，ニ-ｃ-１）

図 2　演繹論理パズルの問題と答えの要約（実験 1）

```
SYSTEM
    PuzzleSample

SETS
    PERSON = {イ, ロ, ハ, ニ};
    AGE = {a, b, c, d}

CONSTANTS
    ages, stuff

PROPERTIES
    ages  ∈ PERSON >+>> AGE  ∧
    stuff ∈ PERSON ---> 1..5 ∧

    stuff(ages~¹(b)) + stuff(ロ) = 4 ∧
    /*「b歳の人物とロの2人合計で，計4つ所有している．」*/
    ロ↦b ∉ ages                    ∧
    /*「b歳の人物とロは別人，つまりロはb歳では無い．」*/


    《中略。》


//    ∧ (ages  ≠ {イ↦d, ロ↦a, ハ↦b, ニ↦c} ∨
//       stuff ≠ {イ↦3, ロ↦2, ハ↦2, ニ↦1}  )

END
```

図 3　演繹論理パズルの B の構文構造への対応付け

て，上記(2)と同じ書式にて用意した．

2.6 実験 1：静的モデルの記述

実験 1 においては，文献[23]にある演繹論理パズルに対する形式モデルを学生に記述させて，ProB の制約充足ソルバ機能を用いてパズルの求解をさせる．例として，文献内の演繹論理パズルにおける問題とその答えについて，装飾を外した要約を図 2 に示す．

図 2 の演繹論理パズルの問題文を基に，図 3 にあるように，Atelier B のシステムモデルの構文構造への対応が付けられたモデルを完成させて，定理証明をさせる．図 3 の最下部のコメントアウト部は，パズルの求解の際に，別解が存在しないことを確認させるために設けたものである．

図 2 の演繹論理パズルに対する形式モデルに対する定理証明の結果を図 4 に示す．

この定理証明において，全ての証明責務は自動証明が行われ，対話証明を求められないように題材を採択している．

コンポーネント	型チェック	証明責務生成	証明責務	証明済	未証明
PuzzleSample	OK	OK	25	25	0

図 4　静的モデルに対する定理証明の結果

問２　ゆずる君はアイススケートが得意です。友達は、ゆずる君が氷の上で上手に４回転ジャンプを決める所を見たがっています。
表１のように論理変数を割り当てたときに、JKフリップフロップを用いて順序論理回路を設計して、必要な論理式を全て答えよ。
《Mr. Yuzuru is good at skating. His friends want to see his quadruple jumps on the ice. Design a sequential logic circuit using JK flip-flops. Answer all of the essential boolean expressions with the assumption that the boolean variables are assigned as "表１".》

表１　順序論理回路の変数の割り当て

論理変数の名称	論理変数が持つ真理値の意味
状態：Q_1Q_0	00：スケート靴を履いて氷上にいる。《Standing on the ice with putting skate shoes on.》 01：スケート靴を履かずに氷上にいる。《Standing on the ice without putting skate shoes on.》 10：スケート靴を履いてストーブの前にいる。《Standing in front of the stove with putting skate shoes on.》 11：スケート靴を履かずにストーブの前にいる。《Standing in front of the stove without putting skate shoes on.》
入力：X_1X_0	00：氷の上に行く。《Go to the ice link.》 01：スケート靴を履く。《Put on his skate shoes.》 10：ストーブの前に戻って、スケート靴を脱ぐ。《Go back to the front of the stove, and put off his skate shoes.》 11：４回転ジャンプが見事に成功する。《His quadruple jump succeeded well.》
出力：Z_1Z_0	00：スケート靴を履いて氷の上にいるように期待される。《Hoped to be on the ice with putting his skate shoes on.》 01：すぐに４回転ジャンプが成功することが期待される。《Hoped his succeeding in quadruple jump coming soon.》 10：友達が拍手する。《Friends clap their hands.》 11：（ドントケア）《(Don't care)》

図 5　小さな動的モデル問題の学生への提示（実験 2）

続いて，ProB にモデルを読み込ませることで，演繹論理パズルの求解を実施する．不完全な記述や誤った記述のモデルを読み込ませた場合，導出される解候補数の規模が大きくなり，求解の結果の表示が遅くなる懸念があるため，実行時間を計測した結果として，ProB の環境変数の設定により解候補数の最大値を 1000 に制約することにした[21].

2.7 実験 2:小さな動的モデルの記述(1)モデル記述（SETS 節より INITIALISATION 節までの記述）

実験 2 においては，高々4 つの状態に遷移する小さな動的モデルを学生に記述させて，定理証明およびモデル検査のそれぞれのアプローチによるモデル検証を行わせる．

題材の数を確保しやすいという理由により，筆者大西が作成した本科 3 年生の授業「論理回路」の試験問題をモデル記述に導入した. 問題例を図 5 に示す. この問題は本来，小規模の順序論理回路を学生に設計させるものである．

図 5 においては，いわゆる「失敗」に値する状態『スケート靴を履かずに氷上にいる』が定義されている．学生には，図 5 における「失敗」の状態はどれであるかを確認させた上で，この「失敗」状態に陥らないという安全性が保証されたシステムのモデルを記述させるために，安全要求としての不変条件を提案させる．

学生によるモデル記述は，Atelier B のシステムモデリングプロジェクトとして実施する．実験の実施に際して，学生によるモデルの記述の発散を防止するために，集合名・変数名を指定し，状態変数の初期値，つまりシステムの初

```
SETS
    type_location =
        {front_of_stove, on_ice};
    type_skate_shoes =
        {putting_off, putting_on};
    type_jumping =
        {not_succeed, succeed}

VARIABLES
    location, skate_shoes, jumping

INVARIANT
    location    ∈ type_location    ∧
    skate_shoes ∈ type_skate_shoes ∧
    jumping     ∈ type_jumping

INITIALISATION
    skate_shoes := putting_off    ||
    location    := front_of_stove ||
    jumping     := not_succeed
```

図 6　学生に事前に提示する記述（SETS 節，VARIABLE 節，INVARIANT 節，INITIALISATION 節）

```
OPERATIONS
    go_to_ice_link =
    BEGIN
    END;

    put_on_skate_shoes =
    BEGIN
    END;

    go_to_front_of_stove_and_put_off_skate_shoes =
    BEGIN
    END;

    try_quadruple_jump =
    BEGIN
        jumping
        ∈(jumping ∈ type_jumping                ∧
          ((location$0    = on_ice      ∧
            skate_shoes$0 = putting_on  ) ⇒
              jumping = succeed              ) )
    END
```

図 7　学生に事前に提示する記述（OPERATIONS 節）

期状態も指導教員があらかじめ指定した．

学生に事前に提示する，SETS 節，VARIABLE 節，INVARIANT 節（モデルへの安全要求としての不変条件を除く）INITIALISATION 節を図 6 に示す．

図 6 の例では，モデルの状態空間を構成する 2 つの変数 location と skate_shoes，および，LTL 式への適用（表 2 参照）のために便宜的に設けた変数 jumping を用意した．

2.8 実験 2:小さな動的モデルの記述(2)モデル記述（状態遷移表に則した OPERATIONS 節の記述）

図 5 の論理回路における 4 通りの入力に対応させて，モデルにおける 4 つの操作を定義した．学生による記述の発散を防止するために，それぞれの操作名を指定した．

学生に事前に提示する，OPERATIONS 節の内容を図 7 に示す．

図7における操作の中身は基本的に空であるが，操作try_quadruple_jump については，実質的に状態空間を構成する変数への新たな代入は無く，変数 jumping への代入が本質であるため，あらかじめ関係箇所のみの記述を行っている．

学生には，「失敗」に値する状態に陥らせないための安全

```
INVARIANT
    (location = on_ice          ⇒ skate_shoes = putting_on)  ∧
    (skate_shoes = putting_off ⇒ location = front_of_stove)
```

図 8　学生に提案させる「不変条件」

表 1　学生に空欄を埋めさせる状態遷移表の 1 つ

go_to_ice_link

状態遷移前の変数値		状態遷移後の変数値	
location$0	*skate_shoes$0*	*location*	*skate_shoes*
front_of_stove	putting_off	**location$0**	**skate_shoes$0**
front_of_stove	putting_on	on_ice	skate_shoes$0
on_ice	putting_off	(don't care)	(don't care)
on_ice	putting_on	location$0	skate_shoes$0

要求として，不変条件『ゆずる君は氷上にいるならば，スケート靴を履いている.』および『ゆずる君はスケート靴を脱いでいるならば，ストーブの前にいる.』を提案させて，INVARIANT 節に図 8 にあるような述語を追加させる．

図 7 の 4 つの操作に対応させて，4 つの未完成の状態遷移表を印刷物として配布し，学生に空欄を埋めさせることで，安全要求を満たす状態遷移モデルを設計させる．

操作 go_to_ice_link に対応させた完成版の状態遷移表を表 1 に示す．学生には表の網掛けのセルが空欄となっており，空欄を埋めさせて 4 つの状態遷移表を完成させる．

学生が表 1 などの状態遷移表の空欄を埋める際には，リファイメントにおける便宜を図るために，状態遷移の前後で変数値に変化が無い場合には，具体的な値では無く，状態遷移前の変数値を意味する「(変数名) $0」の記述を埋めさせる．表 1 において学生が埋めるべき項目の内，太字箇所については，学生にとっては事前に提示された情報のみでは正解が得られない“曖昧な仕様”に当たる箇所であり，システムへの安全要求を満たすために学生が指導教員に問い合わせて明文化させなければならない．表 1 の例では，『ゆずる君がスケート靴を脱いでいるならば，氷上に行けと言われても行かない』という挙動を指導教員が用意する．

4 つの操作について，本来ならば，図 10 にあるような if 文などによる単純な記述をすべきところを，あえて状態遷移表の記述を基にして，学生には図 9 の BEGIN～END 間にあるような becomes such that 非決定的代入[a] [26]の記法を使った，冗長ではあるが状態遷移表に則した記述を完成させて，Atelier B の定理証明によるリファインメント間の無矛盾の検証を経ながら，実装に近い表現に近づけるべく，単純な記述へと洗練させる．

表 1 の状態遷移表の記述を基にして完成させた操作 go_to_ice_link を完成させた結果は，図 9 のようになる．

a 与えられた述語を満足する値を変数に非決定的に割り当てることができる代入の手段である[26]. 代入の対象とする変数を x とする場合,x に係る述語において，代入前を「x$0」，代入後を「x」と表記することとなっているため，x を状態変数とするならば，「x$0」が状態遷移前，「x」が状態遷移後に対応する．

```
OPERATIONS
    go_to_ice_link =
    BEGIN
        location, skate_shoes
        ∈(location    ∈ type_location                ∧
          skate_shoes ∈ type_skate_shoes             ∧
          (location$0 = front_of_stove ⇒
            (skate_shoes$0 = putting_off ⇒
              location    = location$0    ∧
              skate_shoes = skate_shoes$0  ) ∧
            (skate_shoes$0 = putting_on  ⇒
              location    = on_ice        ∧
              skate_shoes = skate_shoes$0  ) ) ∧
          (location$0 = on_ice           ⇒
           location    = location$0    ∧
           skate_shoes = skate_shoes$0  )          )
    END
```

図 9　システムコンポーネントでの 1 つの「操作」の記述

```
OPERATIONS
    go_to_ice_link =
    BEGIN
        IF
            skate_shoes = putting_on
        THEN
            location := on_ice
        END
    END
```

図 10　リファインメントされた「操作」の記述

コンポーネント	型チェック	証明責務生成	証明責務	証明済	未証明
J3test_iceskate	OK	OK	9	9	0
J3test_iceskate_2r	OK	OK	11	11	0
J3test_iceskate_3r	OK	OK	6	6	0
J3test_iceskate_r	OK	OK	15	15	0

図 11　小さな動的モデルに対する定理証明の結果

2.9 実験 2:小さな動的モデルの記述(3)（コンポーネントのリファインメント）

図 9 のようにシステムコンポーネントに記述された操作に対して becomes such that 非決定的代入での記述における条件分岐の冗長な箇所についての統合を繰り返すという洗練化を行い，定理証明によるコンポーネント間の無矛盾の検証を行いながらリファインメントを実施して，最終的に図 10 にあるような，実装に近い記述まで導出させる．

図 9 および図 10 の例では，3 段階のリファイメントを経ている．状態空間を構成する変数を location と skate_shoes の 2 つにしたため，1 段階目と 2 段階目においてはこれらの 2 つの変数に係る条件分岐の冗長箇所の統合を行わせること，3 段階目においては if 文記述に向けた条件式の整理を行わせることを指針とした．システムコンポーネント J3test_iceskate.sys および各段のリファインメントコンポーネント（J3test_iceskate_r.ref，…_2r.ref，…_3r.ref，）に対する定理証明の結果を図 11 に示す．

全てのコンポーネントについて，well definedness properties[b] [27]に係る証明責務は皆無であった．システムコンポーネント J3test_iceskate.sys において，9 個の証明責務の中で 3 個の対話証明が必要となるが，全ての対話証明が PP1（最初のレベルの仮説で述語証明）1 回のみの手続き（当該ボタンの左クリック）で証明済になるように，全ての例題・課題を設定している．3 つのリファインメントコンポーネントについては，全て自動証明のみで証明済になった．

b 対象とする数式において，その数式に相応の有意義な意味が常に割り当てられ，数式が無意味なることが無い保証がされていることである[27]．例を挙げると，全射では無い写像においては値域の要素に写像の対応漏れが存在しうるため，逆写像を参照する際に，対応付けされている値域の要素からのみに限定するという保証がされていることである．

表 2　学生に導出させる LTL 式

ゆずる君が 4 回転ジャンプを決めるまでの経歴を探索するための LTL 式
¬◇({jumping = succeed})
「ゆずる君がスケート靴を履かずに氷上にいることはない」ことを証明するための LTL 式
□(¬({skate_shoes = putting_off ∧ location = on_ice}))
「ゆずる君がスケート靴を履いていないならば氷上にはいかない」ことを証明するための LTL 式
□({skate_shoes = putting_off} ⇒ ¬○({location = on_ice}))

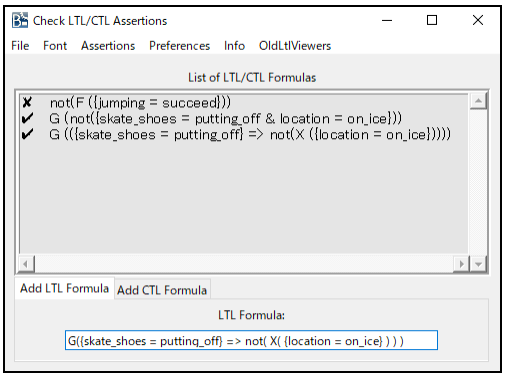

図 12　ProB での LTL 式を用いたモデル検査の結果

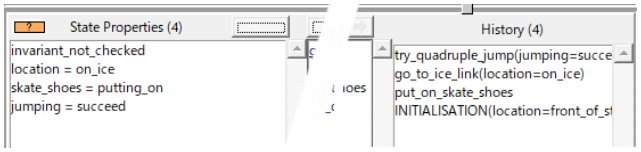

図 13　ProB での LTL 式を用いた状態パスの探索結果

2.10 実験 2:小さな動的モデルの記述(4)（モデル検査）

続いては，ProB にモデルを読み込ませて，LTL 式を用いたモデル検査を実施する．状態パスを探索したいゴールの要件や検査したい安全要件について学生に日本語で示し，表 2 にあるように学生には LTL 式を導出させる．

ProB において LTL 式を用いたモデル検査の結果の画面を図 12 に示す．状態パスを探索する表 2 や図 12 の第 1 式を用いた結果の画面の一部を図 13 に示す．

図 13 の History ビューにおいてゴールまでのパスが表示され，State Properties ビューにおいて現状態としてのゴールが表示されている．

3. 学生アンケートの実施結果の考察および実験実践で得た知見・意見

3.1 学生アンケートの実施と考察

実験 1 および実験 2 の各実験終了時点にて，実験実施の

表 3　学生アンケート（実験 1）の集計結果(N=19)

アンケート設問 達成目標							
思う←	⑤	④	③	②	①	→思わない	平均点
■*Atelier Bについて*							
パズル問題の構成要素を論理式によるモデルで表現できたと思いますか？							
形式仕様記述対象における、明文化された構成要素を定義できる。							
思う←	6	8	4	1	0	→思わない	平均: 4.00
パズル内の要素の関係性を写像(関数)で表現できたと思いますか？							
形式仕様記述対象における、明文化された関係性を定義できる。							
思う←	5	5	8	1	0	→思わない	平均: 3.74
パズル問題のそれぞれの文章と、モデル内のそれぞれの論理式との対応付けを明確にできたと思いますか？							
形式仕様記述対象における、明文化された制約や性質を記述できる。							
思う←	8	8	3	0	0	→思わない	平均: 4.26
パズル問題の文章に書かれていない事実も明らかにできたと思いますか？							
形式仕様記述対象において直接的に明文化されていない事柄の存在を理解できる。							
思う←	8	3	5	3	0	→思わない	平均: 3.84
実験の例題・課題を行う過程で、Atelier Bの使用手順を理解したと思いますか？							
形式仕様の記述支援および論理証明ツールの基本的な使用手順を理解できる。							
思う←	8	10	1	0	0	→思わない	平均: 4.37
■*ProBについて*							
モデルを書き進めて、解の候補数が絞られることを確認できたと思いますか？							
仕様に制約式を追加することで、制約充足ソルバ機能において解空間が限定されていくことを理解できる。							
思う←	17	1	1	0	0	→思わない	平均: 4.84
モデルを書き上げると、パズルの正解が導けることを確認できたと思いますか？							
仕様における制約の記述が適切ならば、制約充足ソルバ機能によって求解できることを理解できる。							
思う←	19	0	0	0	0	→思わない	平均: 5.00
「パズルの正解の論理否定」を条件に加えて、パズルの別解が無いことも確認できたと思いますか？							
仕様における制約の記述が適切ならば、制約充足ソルバ機能によって求解できることを理解できる。							
思う←	17	1	0	0	1	→思わない	平均: 4.74
実験の例題・課題を行う過程で、ProBの使用手順を理解したと思いますか？							
検査ツールの制約充足ソルバ機能の基本的な使用手順を理解できる。							
思う←	11	6	2	0	0	→思わない	平均: 4.47
■*B-Method全般について*							
B-Methodを使った、論理式の表現法に慣れたと思いますか？							
自然言語による仕様を、論理式を用いて表現できる。							
思う←	1	7	6	4	1	→思わない	平均: 3.16
パズル問題に対する論理式を用いた厳密で明解な表現が、正解を自動的に導くことに役立つと理解したと思いますか？							
形式仕様記述の厳密性・明解性の意義を理解できる。							
思う←	12	4	3	0	0	→思わない	平均: 4.47
形式手法を利用することの価値を感じることができたと思いますか？							
形式仕様記述の意義を理解できる。							
思う←	12	7	0	0	0	→思わない	平均: 4.63
実験の例題・課題を行う過程で、形式手法B-Methodに親しみを持てたと思いますか？							
形式仕様記述、論理証明およびの制約充足ソルバ機能による求解の手順の意義について支持できる。							
思う←	10	6	3	0	0	→思わない	平均: 4.37

表 4　学生アンケート（実験 2）の集計結果(N=19)

アンケート設問 達成目標							
思う←	⑤	④	③	②	①	→思わない	平均点
■*Atelier Bについて*							
問題より、漏れの無い場合分けを論理式で表現できたと思いますか？							
形式仕様記述対象に対して、明文的に条件分岐を網羅する取り組みができる。							
思う←	6	10	2	1	0	→思わない	平均: 4.11
問題における安全に関する要件を、不変条件という論理式で書けたと思いますか？							
明文化された機能安全要求を、論理式を用いて表現できる。							
思う←	7	8	3	1	0	→思わない	平均: 4.11
論理式の表現からリファインして、論理証明を進めながら、IF文表現による欠陥の無い記述までに導けたと思いますか？							
垂直リファインメントによって、実装コードに近くて欠陥の無い記述まで詳細化する過程を理解できる。							
思う←	10	6	3	0	0	→思わない	平均: 4.37
不変条件が常に成立することを論理証明して、安全を確かめたと思いますか？							
機能安全要求が満たされていることを、定理証明ツールを活用して確認できる。							
思う←	7	6	5	1	0	→思わない	平均: 4.00
実験の例題・課題を行う過程で、Atelier Bの使用手順を理解したと思いますか？							
形式仕様の記述支援および論理証明ツールの基本的な使用手順を理解できる。							
思う←	12	6	1	0	0	→思わない	平均: 4.58
■*ProBについて*							
モデルをアニメーション機能で動かせることを確認できたと思いますか？							
モデルのアニメーション実行の方法を理解できる。							
思う←	13	5	1	0	0	→思わない	平均: 4.63
アニメーション機能を実行してみて、システムが安全であることを見ることができたと思いますか？							
モデルのアニメーションを活用して、辿れる一部のパスで安全要求を満たしていることを視認できる。							
思う←	9	8	1	1	0	→思わない	平均: 4.32
LTL式を用いて、確認したい状態への経歴を探索できたと思いますか？							
LTL式を定義でき、モデル検査においてLTL式を与えて特定の状態パスを探索できる。							
思う←	6	10	2	1	0	→思わない	平均: 4.11
LTL式を用いて、システムの安全を検証できたと思いますか？							
モデルがあらゆるパスで安全要求を満たしていることを、LTL式を用いて検証できる。							
思う←	9	8	1	1	0	→思わない	平均: 4.32
実験の例題・課題を行う過程で、ProBの使用手順を理解したと思いますか？							
形式検証ツールの基本的な使用手順を理解できる。							
思う←	11	7	1	0	0	→思わない	平均: 4.53
■*B-Method全般について*							
B-Methodを使った、論理式の表現法に慣れたと思いますか？							
自然言語による仕様を、論理式を用いて表現できる。							
思う←	6	8	3	1	1	→思わない	平均: 3.89
不変条件の証明や、LTL式を用いた検証によって、ディペンダビリティ向上を目指せることを理解したと思いますか？							
形式手法がシステムのディペンダビリティ向上に寄与するという意義を理解できる。							
思う←	5	8	4	2	0	→思わない	平均: 3.84
LTL式を用いることにより、システムの全状態を網羅した上で、無欠陥を検証できることを理解したと思いますか？							
システムの全状態を網羅する形式検証の意義を理解できる。							
思う←	10	4	5	0	0	→思わない	平均: 4.26
とにかく動くコードを作るのとは異なる、無欠陥のコードを生成するために、形式手法は役立つことを理解したと思いますか？							
形式仕様が無欠陥のコード生成に寄与するアプローチであることを理解できる。							
思う←	13	5	1	0	0	→思わない	平均: 4.63
実験の例題・課題を行う過程で、形式手法B-Methodに親しみを持てたと思いますか？							
形式仕様記述、論理証明および形式検証の手順の意義について指示できる。							
思う←	12	4	3	0	0	→思わない	平均: 4.47

達成度を自己評価させるべく，学生に対するアンケートを実施している．アンケートの設問については，当初設定の目標達成の見きわめという目的としては曖昧な表現の設問や，形式手法の初学者である学生が読解できるように実験の達成目標に対応する平易な文章表現とした代わりに 2.4 節にある達成目標の趣旨より外れてしまう設問が含まれる．

本稿の提出時点（実験を行った学生数 19 名分）でのアンケートの集計結果を表 3 および表 4 に示す．

全ての設問において 5 段階評価の 3 点超である．8 割超の設問において 4 点超である．総じて一定の評価を得ていると考える．

後述の 1 つの設問を除き，評価点の分布が 2 つに割れる現象を見受けない．これらの項目においては達成度の低い学生群の出現を阻止できていると考える．実験 1 の設問「パズル問題の文章に書かれていない事実も明らかにできたと思いますか？」（つまり，図 2 における「b 歳の人物とロは別人」の類の事実の発見の有無）に評価値 5 と 3 の 2 つの分布の山と 3 名の評価値 2 を見受ける．平均が 3.84 と低い評価値では無いが，時間不足，指導不足の可能性もある．

例題における説明の強調と追加説明書による説明の補充による指導の強化を通じて実践の向上を試みたいと考える．

実験 1 の設問群において，Atelier B 関連より ProB 関連の評価が総じて高い．学生の視座よりは，Atelier B での定理証明によるモデルの無矛盾の確認よりも ProB の制約充足ソルバの機能によるパズルの求解がより明解であり，強い興味を惹いた結果であろうと考える．ProB 関連の 1 つの設問における特異的な 1 名の評価 1 については後述する．

実験 1 および実験 2 における 3 つの設問「実験の例題・課題を行う過程で，Atelier B の使用手順を理解したと思いますか？」「実験の例題・課題を行う過程で，ProB の使用手順を理解したと思いますか？」「実験の例題・課題を行う過程で，形式手法 B-Method に親しみを持てたと思いますか？」について，全学生について 3 以上の評価であり，平均点が 4 点超である．当実験の実践を通じて，ツールの使

用手順の理解度やBへの親近感について一定の評価を得ており，形式手法の敷居を下げる一定の効果を認める．

実験1および実験2における設問「B-Methodを使った，論理式の表現法に慣れたと思いますか？」について，形式仕様記述に関する高度な達成目標に対応する設問項目と考えるが，実験1における平均点は3.16，実験2においては3.89と実験回数を経るごとに向上している．当設問において低評価を出した学生の中には実験1・実験2を通して評価値2の学生1名，評価値1の学生1名が含まれており，この2名の学生を除いた全員は実験2において評価値3以上となっている．実験1・実験2を通じて評価値2の学生は，前述もしているが，実験1のProB関連の設問「「パズルの正解の論理否定」を条件に加えて，パズルの別解が無いことも確認できたと思いますか？」において，他の学生全員が4以上の評価値の中で特異的に評価1を答えている者であり，実験1・実験2を通じて評価値1の学生は，実験1･実験2を通じた残りの設問項目において全て評価値5を答えるという極端な回答をする者である．上記2名による低評価の回答については，回答の指向の特異性に起因する影響が大きいと考える．

2週に渡る実験を実践して,「無欠陥のコード生成するために，形式手法は役立つことを理解しましたと思いますか？」という設問には平均点は4点台の高い評価を受けており,「ソフトウェア技術者の素養としての形式手法の体験」という当初の目的は達成できていると考える．

3.2 実験実践で得た知見

本稿の提出時点である学生数 19 名の実験の実践を通じて,実験の状況および実践より得た知見は以下の通りある．

(1) 実験全体について

使用した学生用PCにおけるAtelier B，ProBの使用について，動作速度なども含め，トラブルは生じていない．

10名程度の各実験班において1名ずつほど，Atelier BおよびProBを自身のPCに導入したい，あるいは独自に無償版を導入したという学生が出現した．

(2) 実験１－写像・関数の扱いについて

比較的能力の低い学生に対しては（未完成の）写像・関数の対応図を与えておくことは必要と考える（2.5節(2)参照）．写像・関数の定義域・値域の対応付けが曖昧に理解されている場合が多く，実験が進むにつれ，課題の題意より関数･逆関数の参照の記述を導出の方法についての学生間での議論が始まるが，その際に写像の対応図が議論の道具として多く用いられる．

当初，例題のモデルにある記述のみでは，関数・逆関数を複合で参照を行う際にマップレット（|->）を使用できないという B の文法を学生が理解できなかったため，追加で説明資料を作成した（2.5節(7)参照）．

(3) 実験２－状態遷移系の扱いについて

例題のモデルにある操作の形式記述におけるbecomes such that 非決定的代入より状態遷移表を学生に書き起こさせ，続いて学生に課題を取り組ませる際に，状態遷移表を書かせてから形式モデルの記述を行わせることは必要と考える（2.8節参照）．当実験の受講学生の能力では，状態遷移表を介さずに適切な条件分岐をしたbecomes such that非決定的代入による記述をすることは不可能に近いと考える．

学生より状態遷移表を書かせる時点において，課題の題意より，モデルの安全を保証するために求められる状態遷移系に課する新たな制約の必要性について，学生が実験指導者に問い合わせることができている．

比較的能力のある学生は，リファインメントの段数が1段でも定理証明，モデル検査を通過させられている（次節のLecomte氏のコメントを裏付けている）．

(4) 実験２－リファインメントの扱いについて(1)

能力に自身の無い学生，特にbecomes such that非決定的代入において文法理解より条件分岐の理解までの広い意味に渡るネスティングの理解が不十分なレベルにある学生は，例題での指針（2.9 節参照）に従って3段のリファインメントを実施しており，最大5段のリファインメントを実施する者が出現した．

(5) 実験２－リファインメントの扱いについて(2)

一部の学生においては，Atelier Bの証明器によって未証明の証明責務が無くなる事象を，プログラミング言語のコンパイルにおける「バグが無いこと」と同じ感覚で，「記述に論理的な誤りが無いこと」と誤解しており，文法的な問題は無く未証明の証明責務は無いが論理的な誤りのある記述を放置していた．結果，そのリファインメント後については正しい論理にて記述した際に，リファインメントの前後の記述における矛盾が生じて証明できない証明責務が発生した時点において，リファインメント元の対応箇所における記述に論理的な誤りを見出すという事例があった．

(6) 実験２－リファインメントの扱いについて(3)

システムコンポーネントにおいて，図7のようなモデルへの安全要求のための新たな不変条件を付加することを学生が失念しており，結果，システムコンポーネントにおける証明責務の数が 0 のままでリファインメントを進めているのを見受けた際に，不変条件の不備を指摘する事例があった．

3.3 ツール提供元よりの意見，およびBの教材について

当実験についての産業界よりの意見を得るため，Atelier B提供元の仏国ClearSy社のLecomte氏（謝辞参照）に，当実験の実践に関する報告をし,次のコメントを頂戴した．

- The first example is mainly a modelling example with

constants (so nothing to demonstrate). It is good to jump start formal modelling. Michael Leuschel used to model sudoku game or the queens problem[24].

- For the second one, maybe you could reduce the number of refinement levels and insist on (easy - with boolean predicates only - to ensure full automation) functional proof with Atelier B and non-functional proof (LTL) with ProB.

また，Lecomte 氏より Atelier B Ver.4.5.1 を活用した B の動画教材サイトである MOOC[28]の紹介を受けており，学生の今後の B の自学習にも活用できるものと考える．

3.4 実験実践の効果向上のためになすべきこと

当実験の実践に際して，今後の効果向上のために以下のことを行うべきと考える．同時に，海外の先進事例などを参考に，形式手法の客観的な教育効果の指標の整理を進めるべきと考える．

- 「フォーマルメソッド利用のレベル」のレベル 1 におけるプログラム自動生成までの手順を完結させる．
- 実験 2 において，図 4 の類の問題に代わる，より実製品の開発に近い題材探しを行う．
- リファインメント前後におけるリンク不変条件や，操作における事前条件/事後条件の考え方を追加する．

4. 関連分野の先例となる実践との比較について

4.1 仕様記述言語ツールを活用した実践例

1 章前半において言及したが，形式手法を活用しつつ数学的論理思考をソフトウェア科学関連の大学学部レベルやそれ以下のレベルの教育に導入する実践例は多数存在する．

文献[5]においては，米国の複数の大学が連携して，教育目的に独自に開発したプログラミング言語上の仕様記述言語を融合する開発環境 RESOLVE を活用した教育プログラムを紹介し，20 年程度に渡り実践していると報告している．

文献[29]においては，形式手法の様々な教育を 30 年近く取り組んでいる米国の大学が，既存ツールにおける教育適用への複雑さを排除する目的でプログラミング言語上の仕様記述言語である Spest を独自に開発し，学部生に対する授業において，既存の JML との 2 種類のツールそれぞれについての自動テスト生成による教育実践を報告している．

本稿での実践と比較して，両者とも長年の経緯に基づいて独自のツールを開発して教育実践していると評価できる．一方，形式仕様記述ツールの適用に限定しており，前者においては，具体的な数値による評価報告が無く，後者においては，本稿での設問形式に近い学生アンケート評価や効率比による分析を行っているが，教育効果の分析よりも開発したツールの評価の意図を感じ，教育効果に関する評価を見受けない．

4.2 モデル検査に基づく実践例

文献[30]においては，インドにおける大学の学部生・院生レベルとされるモデル検査の立場としての形式手法の教育の実践および普及推進の方向性について報告している．

形式手法の教育に際しての克服すべき 3 つの課題（学びたいという動機付け，実世界との関連性，関連する数学の必要性）を指摘し，普及推進のための 3 つの当事者の観点（学部生が学ぶ観点，教員が教授する観点，産業界が企業内研修で教える観点）を述べている．

学部生においては，形式手法そのものの知識は無いが，現状の教育課程で形式手法の基礎となるものを学んでいるため，実世界に則した問題の解決などを通じて学習への動機付けを行うべきと述べている．教師においては，大学での最新のモデル検査ツール（NuSMV，SPIN）に関する実用的な事例やパズルを扱った教員研修の成果を述べている．産業界においては，ワークショップを通じてモデル検査ツールの有効性を訴えて関連数学の必要性に係る神秘性からの脱却を説いて，形式手法の有効性を述べている．

最後には，形式手法の教育に対する要求を満足する唯一の方法は産学連携であると提唱し，実践の様子を報告している．

本稿での実践と比較して，形式手法の教育実践や普及推進への理念的な高い視座に立っていると評価できる．本稿においては産学連携が筆者らによる教育実践への動機付けになっているが，当事例では産学連携を教育実践への具体的な解決策までに結び付けている．一方，当事例はモデル検査ツールの適用に限定しており，受講者には相応の予備知識が事実上備わっていることが当文献における主張の前提となっている．具体的な教育効果の評価については，産学連携のワークショップにおいて，事前と事後のアンケートが実施されているとあるが，実施時の感触などの定性的な報告はあるが，数値による評価報告を見受けない．

文献[31]においては，独国の大学の学部生に対する要求工学の概論を教授する教育課程において，数学的論理思考に根差した「形式的記法」として決定表やシーケンス図などの UML の手法に基づいた教育プログラムを紹介し，SAT ソルバのようなツールの導入の可能性について言及しており，3 年以上に渡り実践していると報告している．

本稿での実践と比較して，学生アンケートなどでは無く，試験による客観的指標による教育効果のデータを蓄積していると評価できる．一方，当事例はモデル検査の考え方の導入に限定しており，ツールを使った実践をしておらず，教育効果についてもシーケンス図の試験においては 0 点の最低評価が毎年，発生している．

5. まとめ

本稿では，形式手法における人材の養成確保の問題への

解決策の 1 つとして「ソフトウェア技術者の素養としての形式手法を体験させる」ことを目標に，工業高専本科生に対して形式手法 B-Method の学生実験を実践した．

工業高専本科生向けの実践例が皆無のため，構想段階より実践にかけて暗中模索の状態が続いたが，実践の結果，当実験は受講学生よりツールの使用手順の理解度や B への親近感について一定の評価を得ており，学生よりツールを導入したいという希望が出ているなどの一定の関心を得ており，学生の形式手法に対する敷居を下げることに成功しており，当初の目標を達成するものであり，人材の養成確保の問題解決に向けて寄与するものと考える．今後の課題として，実験の題材・手順の改善を通じてのブラッシュアップを行い，学生にとっての形式手法の敷居を下げる効果の向上策を検討する．

謝辞

- 経済産業省北海道経済産業局の戦略的基盤技術高度化支援事業を通じて，筆者らは形式手法 B-Method の知見を得る機会を得たことをここに報告して謝意を表し，当事業において共同研究を行った株式会社ヴィッツ，株式会社アトリエ，アーク・システム・ソリューションズ株式会社の関係各位に謝意を表する．
- 当実験で使用しているツール Atelier B の提供元の窓口として日頃アドバイスを頂戴している，仏国 ClearSy 社 R&D プロジェクトディレクタ兼 B モデル・シニアエキスパートの Therry Lecomte 氏に謝意を表する．

参考文献

[1] 青木利晃．車載システム開発における形式手法実践の現状と課題．システム制御情報学会「システム/制御/情報」, 2018, Vol.62, No.4, pp.134-140.
[2] “ウィンターワークショップ 2018・イン・宮島 論文募集 : T4-形式手法－導入支援と技術教育－”, 2018, http://wws.sigse.jp/2018/cfp.html,（2019.07.20 参照）.
[3] 室修治．変革を求められる IT 人材－高信頼設計・検証を担う人材育成－. SEC Journal, 2012, Vol.8, No.1, pp.26-29.
[4] 栗田太郎．形式手法の実践に対してよく尋ねられる質問とその回答－モバイル FeliCa の開発における形式仕様記述を通して－. SEC Journal, 2011, Vol.7, No.1, pp.34-39.
[5] Heym, W. et al.. Integrating Components, Contracts, and Reasoning in CS Curricula with RESOLVE: Experiences at Multiple Institutions, 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T), 2017, pp. 202-211.
[6] “Integrating Mathematical Reasoning into Computer Science Curricula”. http://www.math-in-cs.org/,（2019.07.20 参照）.
[7] 本位田真一，粂野文洋，田原康之，鷲崎弘宜．トップエスイー:サイエンスによる知的ものづくり教育．情報処理, 2007, vol.48, no.11, pp.1264-1272.
[8] Ishikawa F., Yoshioka N. and Tanabe Y.. Keys and Roles of Formal Methods Education for Industry: 10 Year Experience with Top SE Program. The 1st workshop on Formal Methods in Software Engineering Education & Training(FMSEET2015), 2015, pp.35-42.
[9] “カリキュラム標準ソフトウェアエンジニアリング領域 J17-SE（情報処理学会情報処理教育委員会）”. https://www.ipsj.or.jp/annai/committee/education/j07/ed_j17-SE.html,（2019.07.20 参照）.
[10] “苫小牧高専情報科学・工学系 2019 年度 Web シラバス”. https://syllabus.kosen-k.go.jp/Pages/PublicSubjects?school_id=02&department_id=12&year=2019&lang=ja,（2019.07.20 参照）.
[11] “苫小牧高専共通専門科目 2019 年度 Web シラバス”. https://syllabus.kosen-k.go.jp/Pages/PublicSubjects?school_id=02&department_id=23&year=2019&lang=ja,（2019.07.20 参照）.
[12] “大分高専専攻科電気電子情報工学専攻 2019 年度 Web シラバス - 形式手法（西村俊二）”. https://syllabus.kosen-k.go.jp/Pages/PublicSyllabus?school_id=48&department_id=24&subject_code=31AES210&year=2013&lang=ja,（2019.07.20 参照）.
[13] 来間啓伸．トップエスイー実践講座 1　B メソッドによる形式仕様記述 ソフトウェアシステムのモデル化とその検証. 近代科学社, 2007.
[14] Schneider, S.. The B-Method: An Introduction, Palgrave Macmillan, 2001.
[15] “平成 26 年度戦略的基盤技術高度化支援事業の採択結果について”. http://www.hkd.meti.go.jp/hokis/20140728/index.htm,（2019.07.20 参照）.
[16] 荒木啓二郎．フォーマルメソッドの過去・現在・未来－適用の実践に向けて－．情報処理, 2008, vol.49, no.5, pp.493-498.
[17] 來間啓伸，石川冬樹．形式手法に基づくソフトウェア開発手法の紹介－B メソッドを中心に－，日本ソフトウェア科学会「コンピュータソフトウェア」, 2012, Vol.29, No.4, pp.50-58.
[18] “Atelier B とは”. http://www.atelierb.eu/ja/,（2019.07.20 参照）.
[19] “The ProB Animator and Model_Checker”. https://www3.hhu.de/stups/prob/index.php/The_ProB_Animator_and_Model_Checker,（2019.07.20 参照）.
[20] 大西孝臣．工業高専生への導入を目的とした B-Method のモデル記述（１）．ウィンターワークショップ 2018・イン・宮島, 2018, pp.38-39.
[21] 大西孝臣，堀武司．工業高専生への導入を目的とした B-Method のモデル記述（２）．ウィンターワークショップ 2019・イン・福島飯坂, 2019, pp.1-2.
[22] Abrial, J.-R.. The B-Book Assigning Programs to Meanings, Cambridge University Press, 1996.
[23] 小野田博一．論理パズル BEST100. PHP 研究所, 2015.
[24] Leuschel, M. et al.. Easy Graphical Animation and Formula Visualisation for Teaching B. Conference: The B Method, from Research to Teaching, the University of Nantes, France, 2008.
[25] 中島震. ソフトウェア工学の道具としての形式手法(改訂版). 国立情報学研究所「NII Technical Reports」, 2007, NII-2007-007J.
[26] ClearSy, B Language Reference Manual ver1.8.7, 2010
[27] ClearSy, Atelier B Interactive Prover Reference Manual ver4.0, 2009
[28] “Massive Open Online Course, The B-Method”. https://mooc.imd.ufrn.br/course,（2019.07.20 参照）.
[29] Fisher, G. et al. Making Formal Methods More Relevant to Software Engineering Students via Automated Test Generation. Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16), 2016, pp.224-229.
[30] Jeppu, N. et al. Teaching Formal Methods at Undergraduate/ Graduate Level: The three perspectives, 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2017, pp.310-315.
[31] Westphal, B.. An Undergraduate Requirements Engineering Curriculum with Formal Methods, 2018 IEEE 8th International Workshop on Requirements Engineering Education and Training (REET), 2018, pp.1-10.