鉄 道 総 研 報 告
特集：輸送計画・情報技術

信号設備の安全性に関する仕様検証手法の適用検討

寺田　夏樹*　　遠山　　喬*

Application of Verification Methods to Specifications in Concern with Safety of Signalling Equipment

Natsuki TERADA　　Takashi TOYAMA

Formal methods are expected to increase reliability of software, including that of signalling systems. We modeled the specification of automatic block systems for single line with formal specification languages, and verified the model with theorem proving and satisfiability problem solver. For theorem proving we used B-Method which is characterized with theorem proving and stepwise refinement. It yields a very powerful result as far as safety is concerned, but it requires a lot of effort especially when many variables are used. On the other hand, satisfiability problem solver can easily find truth of the difficult proposition, but the domain of the proposition has some restrictions. We also have compared two methods.

キーワード：ソフトウェア，形式手法，単線自動閉そく装置，B メソッド，SMT ソルバ

1．はじめに

信号設備に使用されている電子化機器については，ハードウェアとしてフェールセーフを確保するための技術開発がなされ，現在では広く利用されるに至っている。今後も処理能力の高度化が期待されるが，その場合，ハードウェアに搭載されるソフトウェアの安全性をいかに確保するかがますます重要となってくる。

保安装置用のソフトウェアにおいては安全性を確保するために，N バージョンプログラミングや，シングルスレッド・定周期起動方式の使用など，多くの技術が適用されているが，そもそもプログラムが誤っていた場合には安全性が確保できない。

我々はプログラムの安全性を向上させる手法として形式手法（formal methods）に着目してきた[1) 2)]。形式手法とは，システムやプログラムを何らかの数学的な背景を持つ形式で記述し，それを数学的に検証したり，コンピュータの支援による妥当性や正当性を検証したりするものである（図 1 参照）。例えば数学的な背景を持った仕様記述言語で仕様を記述することで，仕様のあいまいさを排除し，ラピッドプロトタイピングを通じて早期段階で妥当性検証が可能となる。またコンピュータ上での定理証明などの技術を用いることで，システムに誤りがなく，整合性が維持されているなどといった正当性を検証できる。また，検証済みの仕様を自動的にプログラムに変換できれば誤りの混入を減らすことができる。

鉄道において，形式手法は海外での適用事例が見受けられるものの，国内ではあまり採用されていない。形式手法の鉄道信号分野の普及のためには様々な信号保安装置について形式手法を用いたモデル化を行い，例示することが有効と考えている。その際にはモデルの検証についても例示できるのが望ましい。この方針に基づき，鉄道の単線区間の運転方向を制御する自動閉そく装置について，形式手法の 1 手法である B メソッド[3)] を用いて，仕様記述の検討と定理証明による検証を実施した。また，検証手法として充足可能性問題判定を行う SMT ソルバ[4)] を用いた検証の適用についても報告する。

2．単線区間向けの閉そく装置

信号保安装置の代表としては連動装置や ATS/ATC 装置が考えられるが，ここでは連動装置ほどは複雑でないものの，多くの条件が関連して制御されている単線区間向けの閉そく装置をモデル化（形式手法に基づく記述）および検証の対象とした。運転方向制御のうち，最も要となる方向回線の制御に着目し，共通点が多く見られる

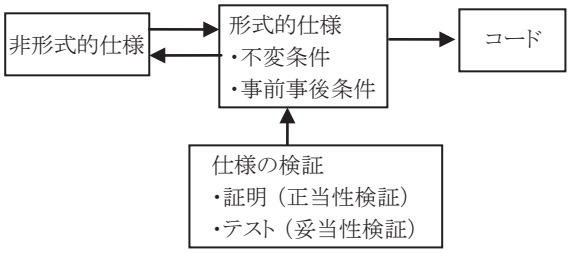

図 1　フォーマルメソッドによるシステム開発

＊　信号・情報技術研究部　信号システム研究室

単線自動閉そく式，自動閉そく式（特殊），特殊自動閉そく式の各装置を対象に仕様のモデル化を実施した。実際には閉そく装置は単独で存在するのではなく，連動装置の一部であり，継電連動装置の標準結線図[5) 6)]にも閉そく機能として結線の記載があるが，ここでは既に実績のある結線図を検証するのではなく，閉そく機能だけを取り出して装置化した場合に安全性が検証できるかという視点でモデル化を行った。そのためリレーと1対1で変数に対応させるのではなく，リレーが実現している機能に着目してモデル化した。

閉そく機能の中心となるのは方向回線と運転方向てこである。上記3種の装置において，詳細は異なるが，基本は以下のとおり共通している。

- 両駅の方向てこを一致させることにより方向回線が構成される（取扱上は着駅側から取り扱う）。方向回線は到着側から電源が印加される。
- 一度方向回線が構成されると到着側の方向てこを扱っても運転方向は変わらない。
- 出発側の方向てこについては，出発進路の設定，出発進路の進路鎖錠の成立，駅間の在線等により鎖錠される。この間に方向てこを扱っても運転方向は変わらない。

今回は，両駅が列車を送出する条件にならないこと，着駅でのてこ操作が無効になる等の要件についての検証を行うこととした。

3．Bメソッドによる単線自動閉そく装置の仕様検証

3.1　Bメソッドの紹介と定理証明の位置づけ

今回使用したBメソッドは検証済みプログラムの生成に重きを置いた手法であり，仕様から生成される定理の証明と，仕様を段階的にプログラムまで詳細化するという概念をもった手法である。

ただし，Bメソッドでは仕様を記述するのが難しいため，比較的記述が容易で，日本語の使用が可能なVDM[7)]と呼ばれる別の手法を併用した。VDMは仕様の記述とその妥当性検証に重きを置いた手法である。VDMのうち，オブジェクト指向対応のVDM++と呼ばれる仕様記述言語にて記述を行い，その後検証やプログラム作成を行うため，B（Bメソッドのうち，仕様記述言語の部分を単にBと呼んでいる）に手作業にて変換し，証明を行った。

VDM++でもBでもシステムを記述する変数の間に不変条件と呼ばれる制約を記述できる。「両駅が列車を送出する条件にならない」ことを検証するには，これを不変条件として記述し，常に保持されることを証明すればよい。また，「着駅でのてこ操作が無効」であることを保証するには，着駅でてこ操作を行った場合に方向回線が変化しないことを証明すればよい。このように，正当性を示すために証明すべき定理を証明責務と呼ぶが，これは仕様から自動的に生成される。証明責務をコンピュータ上で証明するのが実際の検証となる。

3.2　自動閉そく式（特殊）のVDMモデル化

まず自動閉そく式（特殊）の装置をVDM++で記述した。対象を方向回線と駅装置の2つに分割した。駅装置については，それぞれに運転方向リレーに相当する変数「運転方向」と，運転方向鎖錠リレーに相当する変数「運転方向鎖錠」，方向てこに相当する「方向てこ」などを記述した。起点方駅装置および終点方駅装置はそのリレー名称で共通のものが多いので駅装置クラスを作成して，その継承として終点方駅装置クラスを記述した。駅装置クラスの記述例を図2に示す。

```
class 駅装置
types
   public てこ型 = <設定> | <未設定>; --<設定>=反位
   public 鎖錠型 = <鎖錠> | <解錠>;
instance variables
   public 位置 : 方向回線`位置型;
   public 進路てこ : てこ型;
   public 進路鎖錠 : 鎖錠型;
   public 方向てこ : 方向回線`方向型;
   public 運転方向 : 方向回線`方向型;
   public 運転方向鎖錠 : 鎖錠型;
   public 反対側方向 : 方向回線`方向型;
   public 方向回線在線 : 方向回線`在線型;
  inv ….（不変条件）
end 駅装置
```

図2　駅装置のVDMモデル（一部）

一方，方向回線は3本の回線で構成される2対の回線であり，1対は方向てこを制御するための回線，もう1対は軌道回路の条件を照査して運転方向表示リレー（FKR）を動作させるための回線である。これに対応して両駅のてこの状態と軌道回路の状態で方向回線を記述した。

その後，これらのクラス記述の中に，てこを引く，列車が移動して軌道回路の状態が変わるなどといった条件の変化に対応する操作を記述した。

3.3　自動閉そく式（特殊）のBモデル化

VDM++の記述をBで書き直す。VDM++とBとでは構造化の仕組みに違いがあるが，VDM++の各クラスをBでの1つのモジュールに対応させればシステムの構造を大きく変える必要はない。しかし様々な細かい表現の違いが存在する。例えばVDM++では逐次演算が記述できるのに対し，Bの最初の抽象的な段階では逐次演算を記述できない。一方BではVDM++と異なり，演

算の事後条件を直接表現できない。このようなことを考慮に入れて記述を行った。B での表現の例を図 3 に示す。

```
xx <-- DownSet =
  ANY dir WHERE
    dir : DIRECTION &
    ((direction = up & directlock = unlock)
                  => dir = down) &
    ((direction = down or directlock = lock)
                  => dir = direction) &
    transition(direction, directlock, oppositedirect,
                       dir, directlock, oppositedirect)
  THEN
    Set(routelever, routelock, down, dir, directlock,
            oppositedirect ,existence) ||
    xx := dir
  END;
```

図３　B モデルの例（一部）

3.4　B モデルの詳細化と検証

B メソッドでは最初に概要的な（抽象的な）仕様を記述したあと，より詳細化した演算を記述する。詳細化段階においては演算結果が詳細化前の段階の条件を満たすように演算を記述することが求められる。最終的にはプログラミング言語に変換可能な形にまで詳細化する。それを変換してコードが得られる。

各段階で証明責務が生成されるが，大部分は自動的に証明される。残りはツールを使って対話的に証明する必要がある。今回の仕様記述に対しては証明責務を全て証明することができた。実際に証明した数を表 1 に示す。この数は最初の抽象的な段階での証明から実装段階の証明までを含んだものであり，その大小は仕様の複雑さを示す目安となる。表 1 においては最初の段階での B の記述行数についても示した。今回はおよそ 8 割の証明責務が自動的に証明されたことになる。

表１　証明の数（自動閉そく式（特殊））

モジュール	行数（B）	証明責務	自動証明	対話的証明
起点方駅装置	209	99	85	14
終点方駅装置	209	99	85	14
方向回線	60	21	17	4
全体システム	207	243	195	48
計		462	382	80

3.5　単線自動閉そく式のモデル化と検証

続いて単線自動閉そく式のモデル化を行った。単線自動閉そく式の場合，駅中間に閉そくが複数あり，軌道回路に現示情報の伝送を担わせるため，運転方向により軌道回路の送電方向が変わったり，在線状況により送電の極性が変わったりする。標準結線図[6]では下り用の送受信器と上り用の送受信器が設備され，方向回線が設定された方向と同じ方向用の送受信器が使用される。方向回線は設定された方向の軌道回路のいずれかが落下すると鎖錠される。この現示制御を含めてモデル化を実施した。

方向回線に関するモデルについては自動閉そく式（特殊）と大きな違いがない。一方，軌道回路に関して現示の制御部分を含めるとモデルが複雑となる。そこで軌道回路部分を別のクラスとして表現した。証明責務の数を表 2 に示す。これらは全て証明することができたが，軌道回路モデルが複雑となった結果，全体としても複雑となり，全体システムの証明の数が大幅に増えているのが分かる。駅装置や方向回線も数が増えているが，これは鎖錠の条件となる軌道回路が複数となったことによる。

表２　証明の数（単線自動閉そく式）

モジュール	行数（B）	証明責務	自動証明	対話的証明
起点方駅装置	217	183	159	24
終点方駅装置	217	157	133	24
方向回線	69	48	42	6
軌道回路（駅間）	118	471	257	214
全体システム	333	9160	8435	725
計		10019	9026	993

3.6　特殊自動閉そく式のモデル化と検証

特殊自動閉そく式（軌道回路検知式）では駅間に軌道回路が設備されない。その代わり，駅間への列車の出入りを検出する OT と CT と呼ばれる 2 つの短小軌道回路を両駅に設ける。その上で方向鎖錠保持リレー (FSR) を設け，出発側の CT 進入時から到着側の CT 通過時まで方向回線を鎖錠する。FSR の挙動は構内の軌道回路の落下・動作の順序に大きく依存する。方向回線は 2 本で構成されており，列車の到着時には到着側から方向回線に送出する電源を一時的に転極して出発側に伝達する仕組みである。これらが前 2 つのシステムと大きく違う所である。さらに列車が退行することを考慮し，列車が停車場内に残った状態から退行後に扱う退行解錠ボタンや停車場外に進出した後に退行するために扱う場内代用てこが設備され，取扱いの手順も決められている。出発信号が出ていないのにも関わらず列車が出発しようとするのを検出する誤出発検知リレーも設備される。駅間に列車が在線していないことを保証するために，駅構内のリレーの数は増えている。

モデル化では両駅の OT, CT や駅構内のホームトラックと，進路上の軌道回路を記述した。また，実際には様々なリレーの状態によって表現される列車位置を記号で表現することとした。例えば VDM の記述では以下のとおりとなる。

列車位置型 = < なし > | < ホーム > | < 進路内 > | <CT> |

<OTCT>| <OT> | < 駅間 > | < 接近 > | < 到着 >

なお, 線路閉鎖は簡略化のためモデルから除外したが, 場内代用てこや退行解錠ボタン，誤出発検知リレーは異常時の列車の移動に関わるためモデルに含めた。

各列車位置において各軌道回路が落下・動作した場合に，装置が新たな列車位置をどう認識するかを，結線図を元に網羅的に調べた。この中には装置として想定していない動作も含まれるが，無定義にはできないため何らかの列車位置を割り当てることとし，その割当を元に仕様記述を行った。

その結果，例えば軌道落下・動作に対応する演算に関しては，列車位置やその時の軌道回路の状態によって場合分けをすることとなった。

正当性を確認するのに必要な証明の数は表 3 のとおりである。駅装置や全体システムで証明の数が非常に多い。これは駅構内の軌道回路の状態による条件分岐が増えたためであると考えられる。1 回条件分岐を使用すると，分岐した後のそれぞれの状態に対して不変条件等が満たされることを示す必要があり, 証明の数がほぼ倍となる。なお，表 3 では証明の数は 3 万以上に上るが，自動的に証明する場合でも一通りの検証に数時間を要する。

また，単線自動閉そく式では起点・終点それぞれ 217 行であるのに対し，特殊自動閉そく式では 604 行となっており，記述も増えていることが分かる。

表３　証明の数（特殊自動閉そく式）

モジュール	行数（B）	証明責務	自動証明	対話的証明
駅装置（起点・終点計）	604	11370	9763	1607
方向回線	103	112	85	27
全体システム	342	25989	25482	507
計		37471	35330	2141

3.7　B メソッドによる検証のまとめ

B メソッドを用いて単線区間向けの閉そく装置仕様の検証を行った。最も複雑な特殊自動閉そく式でも検証は可能であった。しかしながら，変数が複雑に関連する場合や，内部状態による場合分けがある場合には，必要な検証項目が増え，検証に時間を要することも分かった。

4．SAT/SMT ソルバを使った検証

B メソッドでの検証では検証済みコードを得られるという大きな特徴を持つが，関連する変数の数が増えてくると，それに応じて証明責務が増え，解決が困難となる。現行機器仕様の証明やデータ検証手法を考えた場合，別のアプローチの検討も必要と考えられる。システムの検証手法として，システムを状態遷移系で記述してその性質を網羅的に調べるモデル検査法があるが，変数が多くなると状態爆発の問題がある。そこでモデル検査法と関連しながらも別の手法である SAT/SMT ソルバを利用して，不変条件の保持に関する検証問題への適用を検討した。

4.1　SAT/SMT ソルバとは

SAT ソルバ（SATisfiability problem solver）とは，2 値変数からなる命題論理式が与えられたときに，論理式を真にする変数値の組が存在するかどうか（充足可能性問題）を判定するプログラムである。例えば

$$(A \vee B) \wedge (A \vee \bar{B}) \wedge (\bar{A} \vee \bar{B}) \quad (1)$$

を真にする A と B の値の組が存在するかという問題を考える。A = 真 , B = 偽とすると式 (1) は満たされるので，値の組は存在すると言える。これを判定するプログラムが SAT ソルバである。理論的には充足可能性問題は NP 完全問題とされ，任意の論理式の充足可能性を判定する場合には変数の数の指数で時間がかかるが，実際の問題に対しては探索アルゴリズムを工夫することで，実用的な時間で充足判定性を行う事が出来る場合が多い。SAT ソルバの性能は近年急速に向上しており，10^6 以上の変数を扱う事ができるようになっている[4)]。

SAT ソルバでは，対象とする変数は 2 値変数であり，論理式も乗法標準形 (Conjunctive Normal Form, CNF), すなわち式 (1) のように変数か変数の否定を論理和（∨）でつないだものを論理積（∧）でつないで与える必要がある。これに対して，充足可能性問題の対象を一階述語論理式にまで広げ，さらに整数や関数を適用できるようにしたものが SMT（SAT Modulo Theories）ソルバである。

SAT/SMT ソルバは証明エンジンにも取り入れられており，モデル検査法における探索技術としても使用されている。また，スウェーデンで形式手法による電子連動の検証を行った例があるが[8)]，この検証エンジンは SAT ソルバに基づくものである。

前述したように，小規模装置の検証であっても，B メソッドでは変数が増えることで証明責務が急激に増大し，実用的には証明が困難となってくるが，SAT/SMT ソルバでは変数の数が多くても対応可能と考えられるため，B メソッドで検証を試みた単線区間向けの閉そく装置仕様について別途 SMT ソルバでの検証を実施した。

4.2　不変条件の検証の定式化

B メソッドでの証明では, 演算により変数間の制約（不変条件）が保たれるかどうかを検証している。また，仕様を詳細化する際は，演算結果が前の演算の制約を満たしているかを検証する。SAT/SMT ソルバを用いる場合の検証法について考察する。ここで SAT/SMT ソルバは式

を充足する変数の値の組の有無のみを判定し，変数の値の組を全て列挙するのではないことに注意が必要である。

システムを記述する変数を $x_1,x_2,\cdots,x_n$ とする。さらに不変条件 $I(x_1,x_2,\cdots,x_n)$ が真であるとする。ここで演算を適用するための条件を $P(x_1,x_2,\cdots,x_n)$ が真であることとし，演算を適用した結果，$x_1,x_2,\cdots,x_n$ が $x'_1,x'_2,\cdots,x'_n$ になるとする。

このとき，不変条件を満たす任意の $x_1,x_2,\cdots,x_n$ の組に対して，$x'_1,x'_2,\cdots,x'_n$ が不変条件を満たすことを示すには $I(x_1,x_2,\cdots,x_n)$ および $P(x_1,x_2,\cdots,x_n)$ を真，かつ $I(x'_1,x'_2,\cdots,x'_n)$ を偽とする変数の組が存在しないことを示せばよい。これが示されることで $I(x_1,x_2,\cdots,x_n)$ が真の場合には，$I(x'_1,x'_2,\cdots,x'_n)$ は必ず真であるので，$x_1,x_2,\cdots,x_n$ の初期値が不変条件を満たせば，演算の適用結果は常に不変条件を満たす。

また，演算が事後条件 $F(x_1,x_2,\cdots,x_n,x'_1,x'_2,\cdots,x'_n)$ を真にするものと定義されている場合は，$P(x_1,x_2,\cdots,x_n)$ を真とする全ての値に対して $F(x_1,x_2,\cdots,x_n,x'_1,x'_2,\cdots,x'_n)$ を真とする値が存在する必要がある。この場合 $x'_1=f_1(x_1,x_2,\cdots,x_n)$, $x'_2=f_2(x_1,x_2,\cdots,x_n)$, …, $x'_n=f_n(x_1,x_2,\cdots,x_n)$ となる関数 $f_1,f_2,\cdots f_n$ を考える。$I(x_1,x_2,\cdots,x_n)$ および $P(x_1,x_2,\cdots,x_n)$ を真，$F(x_1,x_2,\cdots,x_n,f_1,f_2,\cdots,f_n) = F(x_1,x_2,\cdots,x_n,x'_1,x'_2,\cdots,x'_n)$ を偽とする $x_1,x_2,\cdots,x_n,f_1,f_2,\cdots f_n$ の組が存在しないことを示せば，演算が可能であることが示せる。

この定式化による検証を行う場合，1点注意が必要である。Bメソッドでは，抽象段階において，演算結果を非決定的に記述することが可能である。この場合，演算結果の値域の中に不変条件を満たす値が存在すればよいとされている。一方，上の定式化では演算の結果不変条件を満たさない値となる場合には，たとえ $I(x_1,x_2,\cdots,x_n)$ が真であっても $I(x'_1,x'_2,\cdots,x'_n)$ を偽とできるので，充足可能となる。このようなことを避けるためには，演算の結果が必ず不変条件を外れないようにする必要がある。事前事後条件での演算の記述についても同様である。

4.3 検証の適用検討

今回の検証にはYices[9]（Version 1）と呼ばれるSMTソルバを用いた。Yicesに与える論理式には2種類の形式がある。1つはSMT-LIBと呼ばれる，様々なSMTソルバで共通化した言語，もう1つはYices単独の記法である。ここでは，scalar型と呼ぶ列挙型の変数が使えることを重視してYices独自の記法を使用した。

Yicesにおいてscalar型の変数は次のように宣言する。

```
(define-type DIRECTION (scalar up down))
```

ここではupもしくはdownの値を取るDIRECTION型を定義している。さらに

```
(define dir-o::DIRECTION)
```

と宣言することにより，DIRECTION型の変数dir-oを宣言できる。

他のSMTソルバも同様であるが，Yicesでは論理式の記述にポーランド記法を採用しており，演算子を被演算子の前に置く。そのため，例えば (A > B) ⇒ ((A = C) ∧ (B = D)) という論理式を記述する場合は

```
(=> (> A B) (and (= A C) (= B D)))
```

となる。

Bメソッドの仕様から変数を抽出し，型および演算前および演算後の変数を定義する。Bメソッドでは仕様記述はモジュール化されているが，SMTソルバに与える論理式は単一モジュールにして与える必要がある。

次に不変条件を記述する。Yicesでは変数間の制約をコンテキスト (context) と呼んでいるが，これを記述するのがassert文である。例えば

```
(assert (= dir-o up))
```

と記述すると変数dir-oは常にupでなければならないという制約となる。このassert文はいくらでも挿入できるので，不変条件を分割して挿入することも可能である。ただし，ここでは不変条件の真偽値を表す変数t1を別途定めることとした。

```
(define t1:: bool (and (not
  (and (= dir-o down) (= dir-d up) …)
```

こうするとt1はand以下で示される論理式と等価となる。演算後の不変条件の真偽値を表す変数t2について別途定め，その上で

```
(assert (and (= t1 true) (= t2 false)))
```

とすると，事前条件を満たし，かつ事後条件が満たされない例を探すことになる。その上で，演算の前後の値の関係を示すassert文を記述し，

```
(check)
```

の1行で探索を開始する。なお，

```
(!set-evidence true)
```

としておくと充足例が示される。

今回の変数は自動閉そく式（特殊）で24, 特殊自動閉そく式で66であったが，いずれも記述した最大34の演算に対する所要時間としては0.1秒以内であった。記述に誤りがあった場合は，充足可能，もしくは判定不能（制限時間内に判定できない場合）となるため，記述を繰り返し修正して再検証することとなるが，再検証も数秒で終了する。

単線自動閉そく式の検証については，最初に閉そくの数を任意とした。この場合，全称記号∀を使用する。しかし，1時間の探索でも終了しなかった。そこで閉そくの数を3つに単純化し，全称記号を省略した結果，自動閉そく（特殊）と同様に0.1秒以内に終了した。変更前の変数の数は36であるのに対し，変更後では60であった。従って，検証の難易度は変数の数で決まるのではなく，使用する論理式の種類に依存することが分かった。閉そくの数を増やしても，検証時間はそれほど変わらな

いと考えられるが，実際には記述に要する労力が伴う。

5．SMT ソルバと B メソッドの比較

B メソッドを用いた定理証明では，特殊自動閉そく式において証明責務が数万となり，解決するのに数週間を要した。それに対し SMT ソルバでの検証は秒単位で終了した。仕様に誤りがあった場合は修正をし，再検証する必要があるが，再検証の時間を考えると，検証に要する時間の差は歴然であり，SMT ソルバが有効であったと考えられる。現在の SAT ソルバは数百万の変数に対応出来るとされているが，数百進路の連動装置（連動図表）であっても変数の数は数千から数万程度に収まると考えられ，時間的に検証可能な範囲になると考えられる。

前述のモデル検査法が連動図表の検証手法として検討され，いくつかの適用事例 [10] があるが，単純に進路数が 10 としてそれぞれが鎖錠・解錠されていると考えただけで状態が 2^{10}=1024 に達することからも分かるように，進路数が多い場合には状態爆発の問題がある。このような場合であっても，不変条件の保持などについては SAT/SMT ソルバで解決される可能性がある。ただし，検証項目については再検討する必要がある。

なお，SAT/SMT ソルバは充足解の探索の支援機能を有してはいるものの，単体では充足可能性問題の判定のみを行うことに注意が必要である。誤った論理式を与えれば当然誤った結果が出力される。そのため，実際の装置に適用する場合，仕様から検査のための論理式を自動生成する機能が必要と考えられる。

任意の数の閉そくを扱った場合からも分かるように，SMT ソルバで全称記号等を扱う場合，処理能力が落ちる。従って元々が結線論理で構築されている閉そく装置や連動装置などでは，SAT/SMT ソルバは有効であるが，ブレーキパターンの計算や線区データベースなど，集合や自然数を扱う場合は更に詳細な検討が必要で，B メソッドのような定理証明手法の方が有効と考える。

仕様通りのプログラムを作るという点においても B メソッドは有効である。その得失を考慮した上で検証やコード生成を行う必要がある。

6．おわりに

単線自動閉そく装置を例に B メソッドを用いた定理証明および SMT ソルバを使用した自動検証の適用を試みた。B メソッドにおいては，変数が多くなると証明に手間がかかるが，集合や自然数を扱う場合などは有効であり，検証済みコードが得られる利点がある。一方，SMT ソルバでは自然数等の扱いは苦手だが，2 値変数等に限れば変数が多く複雑であっても対応可能である。今後も他のシステムへの適用を試みるとともに，様々な検証手法の適用についても試みることで，鉄道信号システムに用いられるソフトウェアの安全性向上に寄与したい。

文　献

1）寺田夏樹：鉄道信号システムへのフォーマルメソッドの適用，鉄道総研報告，Vol.16, No.7, pp.15-20, 2002
2）寺田夏樹：段階的詳細化による鉄道信号へのフォーマルメソッド適用法，鉄道総研報告，Vol.21, No.11, pp.41-46, 2007
3）Abrial J-R, *The B-Book Assigning programs to meanings* Cambridge University Press , 1998.
4）梅村晃広：SAT ソルバ・SMT ソルバの技術と応用 , コンピュータソフトウェア , Vol.27, No.3, pp.24-35, 2010
5）継電連動装置標準結線図 , 日本鉄道電気技術協会 , 1987
6）継電連動装置（CTC 関連及び ARC）標準結線図 , 日本鉄道電気技術協会 , 2000
7）J. Fitzgerald, P. G. Larsen, *Modelling Systems -- Practical Tools and Techniques in Software Development*, Cambridge University Press, 1998.
8）A Boralv, Case Study: Formal Verification of a Computerised Railway Interlocking, Formal Aspects of Computing, Vol.10, pp.338-360, 1988.
9）http://yices.csl.sri.com/
10）川村正，藤井英明，土田勝紀，高橋和子：鉄道信号システムの連動装置の形式的検証向けモデル化と検証環境構築，電子情報通信学会論文誌 , Vol. J88-D-I, No.12, pp.1727-1739, 2005