特集
フォーマルメソッドの新潮流
1

Part I：歴史と概要

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―

荒木啓二郎 ● 九州大学大学院システム情報科学研究院

フォーマルメソッドの出自や過去の変遷，および，現状について概説し，実際のシステム開発における今後のフォーマルメソッド適用に関する効用やガイドラインについて述べる．

はじめに

フォーマルメソッド（formal methods）とは，システム開発において，開発対象を数学的に記述し，分析することによって，品質の高いシステムを効率よく開発するための方法である．我が国では，形式手法や数理的技法などと呼ばれることが多いけれども，本稿では，カタカナでフォーマルメソッドと記すことにする．

近年，ソフトウェアシステムの不具合が日常社会生活に及ぼす影響や，システムの安全性に関する国際標準やガイドラインへの対応などから，我が国においてもフォーマルメソッドに対して関心が高まっている．システムが正しく動作することを保証するために，信頼性の高いソフトウェアを開発するために，プログラムのテストを(半)自動化するために，などフォーマルメソッドに対してさまざまな期待が寄せられている．その一方で，フォーマルメソッドは難しい，現場への導入が困難そうである，導入の効果が分からない，などの多くの疑問や批判的な意見もある．

今回，本特集「フォーマルメソッドの新潮流」で紹介される我が国におけるフォーマルメソッドの具体的な適用事例により，フォーマルメソッドに対するより正確な認識が広まることを期待している．フォーマルメソッドの全般的な紹介については，中島による優れた長編の解説論文があるので[1]，詳しい内容はそちらに譲る．本稿では，紙面の制約から断片的かつインフォーマルに，フォーマルメソッドの出自を概観したあと，その特質について紹介し，フォーマルメソッドの今後，特に我が国における普及について述べる．まず，フォーマルメソッドの黎明期から20世紀末頃までを「過去」として，その歴史を振り返り，21世紀に入ったこの十年近くを「現在」として，現状を概観し，最後に，今後「未来」における我が国での実際の適用について論じる．

過去

ここでは，フォーマルメソッドの初期から20世紀末くらいまでの歴史を概観して，その位置付けや役割についての変遷の概略を示す．1960年代から1990年代にかけてのキーワードを以下に記す．

1960～70年代
- プログラミング言語の意味論
- プログラムの検証理論

1980年代
- 種々の形式手法の提案
- 適用事例研究

1990年代
- ツールの開発
- 実用システムへの適用実績

1970年代から1980年代にかけては，フォーマルメソッドの基礎となるプログラミング言語の意味論およびプログラムの正しさを数学的に証明するプログラム検証理論の研究が盛んであり，日本においてもSSE（Software Science and Engineering）という研究集会や本会のソフトウェア工学研究会やソフトウェア基礎論研究会などで多くの研究発表が行われた時期があった．その後，日本においては，この分野の研究が下火になってしまったが，海外ではヨーロッパを中心にして，基礎研究と応用研究とが息長く継続されてきた．

●フォーマルメソッドの多様性

Jonathan Bowenの Webページには100近い種類のフォーマルメソッドないしツールが列挙されている[2]．これらの多くは1970年代後半から1980年代にかけて研究や開発が始められた．それらの中のごく一部について，

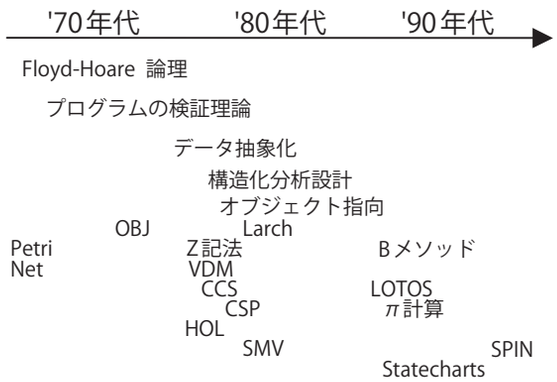

図 -1　フォーマルメソッドの概略年表

ソフトウェア工学におけるいくつかの主要な概念とともに大まかな年表を図 -1 に示す．以下，本稿で名前を挙げる手法やツールについては，Bowen のこの Web ページからリンクを辿って情報を得ることができる．

●モデル指向フォーマルメソッド

この時期は，フォーマルメソッドの中でも，いわゆるモデル指向と呼ばれる Z や VDM が主要な位置を占めていた．これらは，開発対象となるシステムを抽象的なモデルとして，論理学と集合論を基にした数学的実体を用いて表現し，その機能や性質を記述し分析する．この抽象的な記述では，システムが何をするか（What）に注目して，実現の詳細に依存しないシステムの本質的性質を抽出して厳密に記述することを目的とする．この抽象的な記述は，システムが何をするか，どういう効果をもたらすか，というシステムの機能に関する仕様を表しており，形式仕様（formal specification）と呼ばれる．モデル指向のフォーマルメソッドでは Floyd-Hoare 論理における事前条件（precondition）と事後条件（postcondition）および不変条件（invariant）とによって記述された仕様を用いるため，入出力間の関係やプログラムの実行による状態の変化を常に取り扱うプログラマにとって親しみやすい手法であるということができる．

この抽象的な形式仕様を段階的に具体化して，計算機上で実行可能なプログラムを作成するのが段階的詳細化（stepwise refinement）によるプログラム開発である．具体化する際の各段階において，具体化された記述が，具体化される前の記述において成り立つ性質を保存することを保証すれば，最終的に得られたプログラムが当初の仕様を満足することがいえる．図 -2 には，フォーマルメソッドに基づく正しいプログラムの開発方法の典型としての段階的詳細化による開発法を示す．

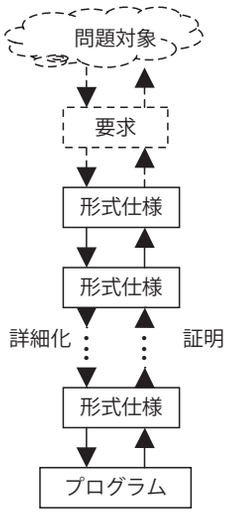

図 -2　段階的詳細化によるプログラム開発

●性質指向フォーマルメソッド

モデル指向とならんで，性質指向のフォーマルメソッドと呼ばれるものがある．Larch や OBJ などの代数仕様に基づく方法がそれに含まれる．そこでは，システムが有する性質を公理として記述する．プログラミング言語の重要な概念である抽象データ型の代数仕様記述に関する研究が活発に行われた．

性質指向のフォーマルメソッドは，並行動作プロセスの振舞いに関する仕様記述においても展開された．通信システムの記述のために開発された LOTOS（Language of Temporal Ordering Specifications）は，代数仕様記述言語 Act One と，並行動作システムの仕様記述と論証のための代数理論である CCS（Calculus of Communicating Systems）とを組み合わせたものである．

●モデル検査手法

モデル検査（model checking）もこの時期に提案されたが，当初は LSI の回路設計検証や通信プロトコルの検証に適用されて有効性を示した．なお，モデル検査の創始者である Edmund M. Clarke，E. Allen Emerson，Joseph Sifakis の 3 名に 2007 年度の ACM のチューリング賞が授与された．

●適用事例の蓄積

その後，1990 年代は，それまでの多種多様なフォーマルメソッドの提案とツール開発や事例研究の成果が花開いて実を結び始めた時期であると見なすことができる．その背景には，たとえばヨーロッパにおける ESPRIT ProCos（Provable Correct Systems）プロジェクトのように，1989 年から 1992 年の第一期に続いて，1992 年か

1. フォーマルメソッドはソフトウェアが完全であることを保証できる.
2. フォーマルメソッドはすべからくプログラムの証明である.
3. フォーマルメソッドは絶対的な安全性が要求されるいわゆるセーフティクリティカルシステムにのみ有効である.
4. フォーマルメソッドは高度に訓練された数学者を必要とする.
5. フォーマルメソッドは開発コストを増加させる.
6. フォーマルメソッドはユーザには受け入れられない.
7. フォーマルメソッドは現実の大規模ソフトウェアには使われない.

図 -3　フォーマルメソッドの七つの社会通念 [3)]

1. 初期の段階での誤りの発見に有効である.
2. 開発対象システムについて深く考えさせることに寄与する.
3. いかなる応用分野にも有効である.
4. 数学を基礎としているもののプログラムよりはずっと理解しやすい.
5. 開発コストを減少させる.
6. 顧客が購入しようとしているものの理解を助ける.
7. 産業界における実用プロジェクトに用いられて成功している.

図 -4　フォーマルメソッドの七つの事実 [3)]

ら 1995 年までの第二期と長期にわたるヨーロッパ諸国にまたがる戦略性を持った共同プロジェクトの果たした役割が大きい.

●啓発普及

IEEE Software 誌では，1990 年 9 月号においてフォーマルメソッドの特集が組まれ，フォーマルメソッドの特質を的確に著したとして評価の高い Anthony Hall の論文 [3)] をはじめとして，フォーマルメソッドの具体的な適用事例も紹介された．Hall は，当時の欧米においてフォーマルメソッドに対する必ずしも根拠のない社会通念を列挙し，それらに対する反論を述べることによってフォーマルメソッドに対するより正確な認識を読者に与えた．Hall が掲げた社会通念を図 -3 に示す.

これらの社会通念は，誤解に基づいており，これらに対する反論がフォーマルメソッドに関する事実を表しているというのが Hall の主張である（図 -4 参照）.

また，フォーマルメソッドの実用に関する網羅的な調査が行われたり [4)]，フォーマルメソッドの概説論文も出版された [5)]．1999 年には，フォーマルメソッドの世界大会（World Congress on Formal Methods）と称された大きな国際会議がフランスのツールーズで開催された．それまでのフォーマルメソッドの基礎研究ならびに実用研究の集大成をまとめて発表する場として，この分野の先駆者である重鎮たちがほとんど出席し，多くのツール展示も行われ，多数の参加者でにぎわった.

現在

●フォーマルメソッド利用のレベル

前述のように多種多様な理論や手法やツールが提供されているフォーマルメソッドであるが，その利用に関して近年では以下のような 3 段階が示されている.

レベル 0：形式仕様記述

数学的な記法を用いて厳密な仕様を記述する．この記述を基にしてプログラムを開発する．証明や分析までは行わない.

レベル 1：形式的開発および検証

プログラムの性質を証明したり，詳細化により仕様からプログラムを作成する.

レベル 2：機械支援による証明

定理証明器や証明支援器を用いてプログラムの性質を証明する.

レベル 0 の形式仕様記述だけを行う場合でも，フォーマルメソッドの効用は大きい．対象システムの機能や構成を厳密に記述しようとすることによって，Hall の 2 番目の言明がいうように，問題対象の理解が深まり，見通しが良く分かりやすいシステム記述が得られる．仕様記述の不十分さや曖昧さに起因する多くの問題が開発の早い段階で明らかとなり，システムの品質と開発効率を向上させることに寄与する．本特集にも掲載されている我が国におけるモバイル FeliCa IC チップの開発プロジェクトは，このレベルの例である.

レベル 1 は，安全性やセキュリティにかかわる高信頼性システムの開発に用いられることが多い．レベル 2 は，システムの不具合が致命的な影響を及ぼす絶対的な信頼性が要求されるシステムに対して，特に核心部分について適用される．B メソッドを用いたパリの地下鉄 14 号線やフランスのロワシー国際空港のシャトルにおける制御システムの開発においては，段階的詳細化により Ada プログラムを作成し，その過程で数万にのぼる証明責務（proof obligation）と呼ばれる条件式を証明支援システムを用いて証明した．これらの例は，レベル 1 とレベル 2 が組み合わさったものと見なすことができよう.

●モデル検査ブーム

最近，我が国ではモデル検査への関心が高い．前述のように，従来はハードウェアの設計検証や通信プロトコルの検証に利用されて，その有効性が示されていたが，近年，実用的な自動検証ツール，ならびに，不具合検出

1. 上流工程での問題点発見と早期解決に効果がありそう.
2. 導入するのが難しそう.
3. ツールを用いた自動検証機能が必要.
4. 多様な大規模プロジェクトへの適用事例を知りたい.
5. フォーマルメソッド導入の効果が分からない.
6. どの手法・ツールを使ったらよいか分からない.
7. 他の開発工程との関係が分からない.

図 -5　日本におけるフォーマルメソッドに対する認識

時の反例の提示によるデバッグ支援などのキャッチフレーズにより大きな関心と期待が寄せられている．しかしながら，状態爆発や抽象化したモデルにおいて検証した内容の意味などについての課題も存在する．我が国においては，産業技術総合研究所のシステム検証研究センターがモデル検査に関する研究と実用化，ならびに教育普及に関して精力的に活動している．

●我が国における状況

本特集に掲載されているように我が国においてもフォーマルメソッドの実用的な適用事例がある．また，講習会やセミナなどもしばしば開催されており，おおむね盛況のようである．我が国においても，ようやく Hall の七つの社会通念の議論ができる状況になってきたといえよう．ただし，必ずしも正しく認識されているわけではないのは Hall の場合と同様である．我々がこれまでに開催してきたフォーマルメソッドに関する講習会やセミナなどにおいて，参加者からよく聞かれた意見を図 -5 に示す．

フォーマルメソッドの啓発普及のための講習会やセミナに参加した後での参加者の意見なので，システム開発の上流工程において，要求や仕様を明確化し明示的に記述することによって開発効率やシステムの品質を向上させることに効果がありそうであるということは，参加者に伝わっているようである．しかし，講習会やセミナの中で，ヨーロッパを中心にさまざまな分野において実用システムの開発に適用された実績が多数あることに言及はしても，個別の事例を具体的に紹介してはいない．そのため，導入の方法や効果に対する評価も含めて，それらの具体事例の詳細を知りたいという要望が多い．

フォーマルメソッドに関する資料はたくさんある．書籍や学会の解説論文やインターネット上の Web ページから膨大な資料に辿り着くことができる [2), 4), 5), 6)]．ただし，日本語の資料が少ないので，入門書や解説書はもとより，適用事例報告や導入案内などを整備し充実させる必要がある．また，フォーマルメソッドのコミュニティを立ち上げ育成するための活動も行って，問題意識，知識，スキル，経験などの共有を図ることも必要である．

●フォーマルメソッドの効用

ここで，フォーマルメソッドの効用について少し述べる．フォーマルメソッドの効用には，直接的なものと間接的なものがある．

直接的な効用としては，開発対象システムに関して，種々の成果物が得られる．開発されるシステムを構成するプログラム，仕様書，設計書，利用マニュアルなど各種の文書や資料を高品質に効率よく生産することに役立つ．また，証明されたり確認されたりしたシステムの性質や特性も成果物に含まれる．

間接的な効用は，直接的な効用で挙げた成果物のように目に見える形で現れるわけではないが，むしろ直接的な効用よりも大きい．Hall が 2 番目に挙げているように，フォーマルメソッドを適用しようとすると，おのずと開発対象に対する理解が深まり，問題の構造や関連する概念が明確になる．それを厳密に表現し記述しようとすることによって，開発者間の認識や理解の共有を図ることができるようになる．開発者間同士の，また，開発者と顧客やシステム利用者との間のコミュニケーションの齟齬も少なくなる．Hall が 6 番目に挙げているように，顧客にとっても利点があるということになる．

レベル 0 の形式仕様記述の段階だけでも，大きなメリットがあるのである．しかも，レベル 0 では，レベル 1 やレベル 2 と比べると，適用するためのコストも低く，かつ前述のように適用効果も高い．フォーマルメソッド適用経験者は，例外なくこの間接的効用について言及する．ただし，この間接的な効用は，定量的に示すことが難しい．そのため，フォーマルメソッド適用経験者が，この効用を強調して説明しても，なかなか納得してもらえない．筆者は，開き直って，「やってみなければ分からない」などと嘯くこともある．

●連携統合化

前述のように多種多様なフォーマルメソッドおよびツールが存在するが，ヨーロッパでは相互連携の気運が出てきた．元来，ヨーロッパの研究者は，自分自身のアイデンティティを大事にして，個別の研究を深めて他との相違を主張する傾向があった．それが前述の多種多様なフォーマルメソッドの発展の一因であると考えられる．ところが，近年，システム開発という多面性を有する複雑な対象に対して，単一の理論や方法論では部分的にしか解決できないという認識のもとに，EU 統合という状況も推進力となって理論や方法論の融合や統合が進められている．

上述のようにヨーロッパの研究者たちが自己のアイデンティティを重視するということは，他人のアイデンティティも尊重するということであり，狭いヨーロッパ

の中で自分のことも他人のことも互いによく知っている．その彼らが戦略的に協調するのであるから，その影響は大きくないわけがない．英国計算機学会の FACS 分科会[7]，欧州フォーマルメソッド学会（FME）[6]では，フォーマルメソッドに関する研究，実用化，教育などに関する活動を長年にわたって活発に行ってきており，現在のフォーマルメソッド隆盛の原動力となってきた．また，英国の Grand Challenge や EU の RODIN（Rigorous Open Development Environment for Complex Systems）および DEPLOY（Industrial Deployment of System Engineering Methods Providing High Dependability and Productivity）などの国際的な共同プロジェクトを推進して，ヨーロッパ各国ならびに他の地域との連携を行ってきている．我が国からも，これらの国際共同プロジェクトに参加し貢献することが望まれる．

未来：今後の適用に向けて

ここでは，フォーマルメソッドの今後の発展に関するおおまかな方向性を簡単に述べたあとに，今後，フォーマルメソッドの適用を，特に我が国において，広めるために参考となるフォーマルメソッド利用のガイドラインや留意点を紹介して，最後に本稿のまとめを述べる．

●フォーマルメソッド研究開発の方向性

ここでは，フォーマルメソッドの今後の発展の方向について述べる．当然のことながら，理論の発展やツールの開発は進み，実用システム開発への適用も行われていくことは間違いない．ツールに関しては，いくつかの異なる手法の統合化および分析や検証の自動化が進む．フォーマルメソッドでは，分析的に対象の性質を明らかにしたり証明したりすることが多いが，逆に，構成的にシステムの全体的なアーキテクチャに基づいて，部分から全体を合成することを支援する方法やツールについての研究開発も求められる．実時間組込みシステム開発からのニーズにより，時間概念の取扱いや連続系と離散系とのハイブリッドシステムも重要な対象となろう．また，記述の対象もプログラムや仕様記述より抽象度が高い，ユーザや顧客の要求を直接取り扱う方向にも発展するであろう．

●フォーマルメソッド適用のガイドライン

多種多様なフォーマルメソッドがあるために，図 -5 の 6 番目に掲げているように，どれを使ったらよいのか分からない，それぞれがどういう分野や問題に適しているのかが分からないという指摘をしばしば受ける．数あるフォーマルメソッドの中から主なものについてだけでも，それぞれの特徴や適用分野などを調べたり，教えてもらったりした後で，さて，自分はどれを使おうかということを検討するのでは時間がかかりすぎるであろう．

1. 汝，適切な表記法を選ぶべし．
2. 汝，形式化を行うべし，されど過ぐること勿れ．
3. 汝，コスト予測をすべし．
4. 汝，フォーマルメソッドの指導者を身近に持つべし．
5. 汝，従来よりの開発法を棄つること勿れ．
6. 汝，十分に文書化すべし．
7. 汝，自らの品質標準を危うくすること勿れ．
8. 汝，独善となる勿れ．
9. 汝，テストすべし，またテストすべし，さらにテストすべし．
10. 汝，再利用すべし．

図 -6　フォーマルメソッドの十戒[8]

フォーマルメソッドを適用して有用な成果を得ようとすれば，まず最初に，自らの問題自体を理解する必要がある．自分たちの目的はいったい何なのか，条件や制約としてはどのようなものが存在するのか．そして，それに適うフォーマルメソッドは何かを選択するのである．

また，前述のフォーマルメソッド利用のレベルに関して，いきなりレベル 2 で利用するということは現実的ではない．レベル 0 から段階を踏んで，より高度な利用に進むのが自然である．レベル 0 でも，前述のように，問題対象の理解を深めたり，関係者間での認識や経験の共有を促進したり，コミュニケーションの齟齬を小さくしたりするなど大きな効果が得られる．

これは，数学的な道具建てを用いて，開発対象となる問題領域やシステムを厳密に記述しようとすることによってもたらされる効果である．その効果は，間接的なものであり定量的に示すことは難しいけれども，実際に適用経験を持つ人は，ほぼ一様にその有効性に気付く．フォーマルメソッドとは書くことであり，書くことによる生活改善運動であるという主張をここに提示しておこう．

●フォーマルメソッド利用上の注意

フォーマルメソッド利用に際しての留意点として，Bowen と Hinchey はフォーマルメソッドの十戒なるものを提示した[8]．図 -6 にその十戒を掲げる．

前述のように，自分たちの目的や制約などが明らかになったとしても，種々のフォーマルメソッドの中からどれを選べばよいのか，それによって，どのような効果が得られるのか，これらの十戒は直接答えてくれない．どれが適切な表記法なのか，過剰にならない形式化とはどのようなものか，従来の開発法との関連はどうすればよいのか，などなど，実際に適用しようとすると分からないことばかりであろう．しかし，前述のように適用事例の資料はたくさんあるのだから，Polya の教えに従って[9]，

似た問題はないか，その結果を使うことができないかを検討すればよい．十戒の 10 番目がいうように再利用を図るわけである．また，フォーマルメソッドの実際の適用において，十戒の 4 番目は，特に有益である．フォーマルメソッドの特定の手法を選択する際や，適用途中にこれでよいのか確認する際の先達の助言や励ましは何よりもありがたく有益なものである．

これらの十戒は 1995 年に提示されたものであるが，Bowen と Hinchey は 10 年後でも通用すると主張している[10]．ただし，4 番目は，「汝，フォーマルメソッドの指導者とドメインエキスパートの両方を当初から持つべし」として，開発プロジェクト自体とフォーマルメソッドとの連携を強調している．

おわりに

本稿では，フォーマルメソッドの特質や適用に際してのガイドラインなどについて，欧米，特にヨーロッパのフォーマルメソッドコミュニティの資料を中心に紹介した．我々自身の経験に基づいて，我々独自の内容をここに提示してもよいのだが，結局，似たような内容になってしまうので，既存の資料を引用したり参照したりすることにした．とはいえ，前述のように，話を聞いたり読んだりしただけでは，分からない．実際にやってみることが重要だ．まずは，レベル 0 から，簡単な例題や身近な問題を題材にして，やってみることを強く勧めたい．

FeliCa IC チップの開発プロジェクトにおける事例は，彼らが自分自身の問題を十分認識しており，欲張ることなく適切な目的を設定したことが一番の成功の要因であると評価している．さらに，彼らは，フォーマルメソッドはシステム開発上の部分的な手法であると位置付けて，既存の開発法の中にうまく取り込んで，継続的に利用し続けている．このような事例を種々の分野で蓄積して，経験や知見を共有し再利用できるコミュニティを形成することが我が国において必要である．そのためには，欧米で実施しているような国家や EU レベルでの長期的な戦略のもとでの大型プロジェクトによる支援や活性化が望まれる．

最後に，フォーマルメソッドに対する適切な日本語の名称を考案することが必要であると考える．形式手法という名称からは，誤解を受けたり敬遠されたりする傾向がある．数理的技法というのも何のことなのかよく分からない．よい案があれば，ぜひともご提案いただきたい．正式手法ではどうかという意見も一部にはあるが，システム開発標準としてフォーマルメソッドを用いることを規定するようになれば，まさしく正式な手法と称すべきものである．早く，そのような時代がくることを期待する．

謝辞　日頃から有益な情報や有意義な議論をいただくソフトウェア技術者協会フォーマルメソッド分科会（SEA sig-fm），VDM 研究会，IPA/SEC 高信頼性システム開発手法調査検討会の皆様に感謝する．

参考文献

1) 中島　震：ソフトウェア工学の道具としての形式手法―彷徨える形式手法―，ソフトウェアエンジニアリング最前線 2007, pp.27-48 (2007).
2) http://vl.fmnet.info/
3) Hall, J. A. : Seven Myths of Formal Methods, IEEE Software, Vol.7, No.5, pp.11-19 (1990).
4) Craigen, D., Gerhart, S. and Ralston, T. : An International Survey of Industrial Applications of Formal Methods, Technical Report NIST GCR 93/626 (Vols. 1 and 2), U.S. National Institute of Standards and Technology (1993).
5) Clarke, E. M., Wing, J. M., et al. : Formal Methods : State of the Art and Future Directions, ACM Computing Surveys, Vol.28, No.4, pp.626-643 (1996).
6) http://www.fmeurope.org/
7) http://www.bcs.org/server.php?show=conWebDoc.1215
8) Bowen, J. P. and Hinchey, M. G. : Ten Commandments of Formal Methods, IEEE Computer, Vol.28, No.4, pp.56-63 (1995).
9) 柿内賢信（訳），Polya, G.（著）：いかにして問題をとくか，丸善 (1954).
10) Bowen, J. P. and Hinchey, M. G. : Ten Commandments of Formal Methods ... Ten Years Later, IEEE Computer, Vol.39, No.1, pp.40-48 (2006).

（平成 20 年 3 月 25 日受付）

荒木啓二郎（正会員）
araki@csce.kyushu-u.ac.jp
九州大学教授．工学博士．IEEE Fukuoka Section Chair，VDM 研究会会長，ICFEM2008 大会委員長，IFM'99, FM2003, ICTAC2004 のプログラム委員長など．Formal Methods Europe, ACM, IEEE など各会員．日本学術会議連携会員．