National Institute of
Advanced Industrial Science
and Technology
AIST

AIST-PS-2008-012

モデル検査の教育プログラム構築に向けて

青木利晃 [1]，粂野文洋 [2,3]，木下佳樹 [4]，篠崎孝一 [5]，
高木理 [4]，高村博紀 [4]，田口研治 [2]，中原早生 [4]，
西原秀明 [4]，早水公二 [6]，本位田真一 [2]，渡邊宏 [4]

[1] 北陸先端科学技術大学院大学 (JAIST)
[2] 国立情報学研究所 (NII)
[3] 三菱総合研究所 (MRI)
[4] 産業技術総合研究所 (AIST)
[5] 関西電力 (KEPCO)
[6] メルコ・パワー・システムズ (MPS)

算譜科学研究速報
Programming Science
Technical Report

Research Center
for Verification and Semantics

独立行政法人 産業技術総合研究所 システム検証研究センター 発行
Published by Research Center for Verification and Semantics
National Institute of Advanced Industrial Science and Technology

モデル検査の教育プログラム構築に向けて

青木利晃 [1]，粂野文洋 [2,3]，木下佳樹 [4]，篠崎孝一 [5]，
高木理 [4]，高村博紀 [4]，田口研治 [2]，中原早生 [4]，
西原秀明 [4]，早水公二 [6]，本位田真一 [2]，渡邊宏 [4]

[1] 北陸先端科学技術大学院大学 (JAIST)
[2] 国立情報学研究所 (NII)
[3] 三菱総合研究所 (MRI)
[4] 産業技術総合研究所 (AIST)
[5] 関西電力 (KEPCO)
[6] メルコ・パワー・システムズ (MPS)

1　はじめに

数理的技法[1]とは，論理学や数学に基づいたシステム開発手法の総称である．幅広い技術や手法を指す名前であり，数理的技法に含まれるものと含まれないものを明確に分ける基準や標準などは存在しないが，形式仕様記述，定理証明，モデル検査などが中心的な手法として知られている．

その中でも，モデル検査 [6, 4, 17] は従来手法で困難な網羅的な検査も可能であることから近年特に注目されている．また数理的技法の中でも自動化の割合が高く前提知識が比較的少ないという点で利用者にとって有利である．現在，モデル検査を応用したツール (主要なものには NuSMV [21], Spin [25], UPPAAL [27], LTSA [18] などがある) が開発され，適用事例が着実に積み上げられつつある．

その社会情勢を反映してか，モデル検査 (もしくは数理的技法) の教育活動が行われるようになった．それらは社会人または技術者向けのものであることが多く，大学等研究機関，業界団体や公的機関，または民間企業の主催で各自行われている．

著者らは数理的技法による検証に関する教育を各者独自の立場から進めており，開講実績，書籍出版，対外発表などで，それぞれが一定の実績をあげてきた．互いの活動の比較や協同の可能性を探るため，集まって「数理的技

[1] Formal Methods．形式手法，形式的技法とも呼ばれる．

法の教育に関するワークショップ」を開催し，それぞれの取り組みを紹介した (2007 春)．その後も定期的にワークショップを開催し，議論を重ねている．

そのワークショップは，数理的技法を社会へ普及拡大するための技術者教育の仕組みについて研究することを目的とし，具体的には，

- 各者のカリキュラムの詳細比較を文書にまとめる．
- 数理的技法の知識を体系づけて明確化する．
- 体系化した数理的技法の知識に基づいたカリキュラムを作成する．これをリファレンスとして各者のカリキュラムの見直し，著者ら以外の機関が作成したカリキュラムとの比較に役立てる．
- カリキュラムに沿った学習の到達点として技術者認定制度の整備や普及活動も視野に入れる．

ことで合意した．

しかし，数理的技法のカバーする範囲は広く，またカリキュラム作成から普及活動までを一度に進めるのも膨大な作業となることから，短期的な目標としてモデル検査に範囲を絞り，以下の活動を行うこととした．

- 著者らの持つカリキュラムのうち，モデル検査に関係する部分を詳細に比較し，文書の形にまとめる．
- モデル検査に関する知識を明確にし，知識体系として文書化する．
- 知識領域に基づいて，モデル検査の教育カリキュラムを作成する．更に各者が既に持っているカリキュラムと比較して位置づけを明確化する．

知識体系について説明を加えておこう．一つの分野に関して系統的な教育を行おうとするとき，その分野で扱われる知識，つまり概念，用語，技術，それらの間の関連，重要度など，を把握したうえでカリキュラムを作成すべきである．分野に関するこれらの知識を得るための端緒として，分野全体を体系化し必要に応じて参照できるようにまとめたものを知識体系 (BOK: Body of Knowledge) と呼ぶ．著者らが作成しようとしているものはモデル検査に関する知識体系 MCBOK (ModelChecking Body of Knowledge) である．

本論文では，この活動について計画を述べ，既に得られた成果について述べる．第二章では活動の背景となる，モデル検査とその教育についての現在の状況を産学官それぞれの視点から述べる．更に関連する活動として，国内外の数理的技法の教育に関する研究コミュニティや，システム開発一般について既存の教育カリキュラムを紹介し，活動の位置づけを明確にする．

第三章では先にあげた三つの活動 (現行カリキュラム比較文書，モデル検査知識体系，新たな教育カリキュラム) のそれぞれについての計画の詳細を

説明する．最初に作成するものについて簡単に説明し，次いで作成方針と方法を述べ，必要な作業をあげる．

第四章では，既に得られている結果として，現行カリキュラムの比較について述べる．

更に補足として，個々の活動の詳細や資料を付録に記載した．

2　本研究計画の背景

モデル検査は汎用的な技術であり，抽象的な設計からソースコード，また巨大システムの制御から組込みシステム，電子回路までを対象にすることができる．モデル検査の扱われ方も状況によって様々であり，モデル検査の教育プログラムを構築する際には現状を概観しておくことが有益である．

本章では，いわゆる産・学・官，つまり産業界，学界，公的研究所・試験所などの公的機関，の三者についてそれぞれの見地からモデル検査教育活動の背景を述べる．モデル検査自体が現在どのような意味を持つか，教育プログラムの重要性，実際の教育活動などについて説明する．

詳細については以下の各節で説明されるが，産学官三者の活動を図式化したものが図 1 である．理論という形で整理された知識や先進的な手法を持つ学界から，技術適用の場である産業界への知の移転が現在様々な形で行われており，公的研究所・試験所などの公的機関も新技術の適用を推進するために環境整備や実際の教育活動を進めている．また産業界は実例や経験をもとに独自の教育活動を行っている．

現在これらの活動は大学・企業それぞれの機関で独立に進められている．モデル検査の知識領域を明確にし，標準カリキュラムを提示することで，それぞれの教育活動が協調され，より推進されることが期待される．

2.1　学界から

数理的技法に関する基礎理論としては，コンピュータ科学における論理学や離散数学といったものがある．これらの基礎理論については，従来から，大学の情報系の学部・研究科のカリキュラムに組み込まれている．一方で，その応用として，形式仕様記述法，プログラム検証法，モデル検査などの応用手法があるが，これらについて，カリキュラムに組み込んでいる大学は少ない．数理的技法を実践するためには，これらの応用手法まで学ぶことが必要である．よって，数理的技法を産業界に普及させるためには，将来，産業界の一員となる学生に大学で教育を行う，もしくは，社会人において教育を行うことが望ましい．

基礎理論とは異なり，数理的技法の応用手法の教育を行うためには，実践的な事例に基づく必要がある．現在のところ，ツールなどの環境は整備され

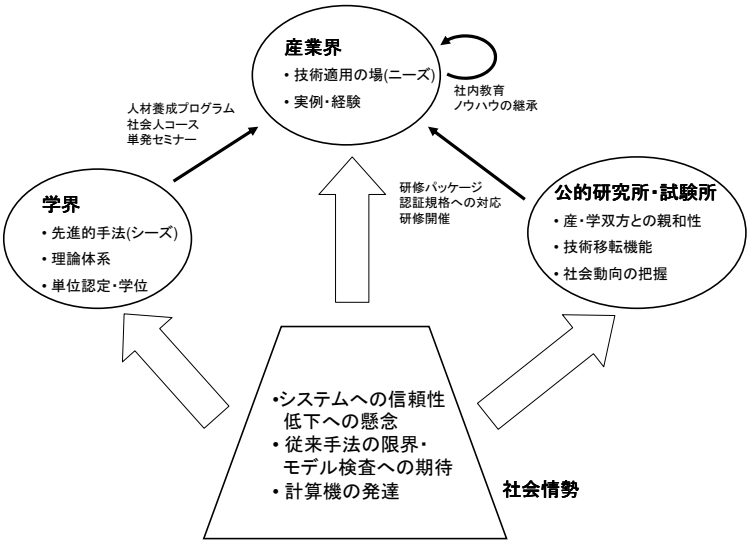

図 1: モデル検査教育活動の俯瞰図

つつあるが，教育用の実践的な事例が不足している．よって，数理的技法の実践的な教育を行うためには，その素材である事例，および，教科書などの整備が必要である．

一方で，現在，数理的技法の中でも，モデル検査に焦点を当てた教育活動が広まりつつある．現在の日本における大学発のモデル検査の教育活動を以下に紹介する．

それぞれの活動の実施概要については補足 A に示した．

- 大学での教育活動.

  モデル検査に関する基礎理論として，言語理論や論理学などが挙げられるが，情報系の大学では，従来から，これらの基礎理論について講義は行われている．モデル検査に特化した講義は，一部の大学では提供されているようではあるが，まだ少数であるように見受けられる．

- 人材養成プログラムにおける教育活動.

  文部科学省は，科学技術振興調整費において振興分野人材養成ユニットとして，教育に関する競争的資金を供給している．これらの内いくらかの事業に関しては，社会人の教育を対象としている．現在，モデル検査に関する教育を行っているのは，国立情報学研究所のトップエスイーと九州大学の QUBE プロジェクトである．

- 社会人コースにおける教育活動.

少子化問題に起因する教育対象数の減少が問題になってきており，社会人を対象としたコースを設置している大学が増えてきた．上の人材養成プログラムの社会人教育と異なるのは，これらのコースでは，実際に大学に入学をして単位を取得することにより修士や博士などの学位を取得することである．また，コースへの入学をしなくても，単科目で履修を行うことも可能である．この場合，正式な単位が授与されるため，他の大学で単位の読み換えを行える場合がある．北陸先端科学技術大学院大学のように，そのような社会人コースにおいて，モデル検査に関する授業を提供しているものもある．

- 一般公開セミナーにおける教育啓蒙活動.

  技術や活動の紹介として，大学の教員が講師となっている一般に公開されたセミナーが開催されている．最近では，モデル検査に関するセミナーも多く見られるようになってきた．例えば，北陸先端科学技術大学院大学と日本科学技術連盟が共催しているセミナーや，ソフトウェア科学会が開催しているチュートリアルなどがある．

2.2 産業界から

システム開発の大規模化と工期短縮の流れは，様々な形で品質への影響を与えており，システムを開発する企業は様々な問題を抱えている．例えば,「複雑なシステムでは，開発仕様を完全に規定することそのものが難しく，開発が進んだ段階になってからも度々，仕様変更が発生して，手戻りにより工数や開発スケジュールが台無しになる」,「期間短縮のためにシステムを分割して並行開発を行うと，全体動作やインターフェースに関する理解が異なっており，統合テストで問題が発生する」,「既存のシステムを再利用して期間短縮を図ると，以前は見られなかった想定外の動作によって問題が発生する」,「テストに必要な時間が短縮されると，必要なテストを全て行うことが難しく多少のリスクを認識しながらテストを効率化するが，運用してから想定外の問題が発生する」といった状況がある．テストを中心にした従来手法だけでは，品質と工期の確保が限界に達しており，生産性と品質の向上を目的とした新しい技術としてモデル検査が注目され始めている．モデル検査は，システムが取り得る全ての状態を網羅的に自動検査する手法であり，従来のテストでは実現できなかった全数検査を自動的に実現できることから，開発の効率化と品質向上に期待が掛けられている．

海外では 80 年代終わりから，モデル検査を治水システムや火星探査機のソフトウェア検証に活用した事例 [25]，航空機の空中衝突防止装置 TCAS II の要求仕様に適用した事例 [2]，さらにプロトコルやアーキテクチャ等に数多くの検証事例 [7] が発表されており，最近の例では，マイクロソフト社が

Windows XP のデバイスドライバー検証ツールにモデル検査に基づいた解析エンジン SLAM を開発して利用している [3].

国内でもここ数年，さまざまなシンポジウムや研究会で，組込みシステムの開発にモデル検査を用いた事例 [23]，モデル駆動開発にモデル検査を組み合わせた事例 [20]，さらには，モデル検査作業の支援ツールを開発した事例 [24] 等が発表され始めており，既に数十件の検証事例を有する民間企業も現れている．大学の社会人コースや公開セミナーでの受講ばかりではなく，一部の企業では，社内教育の一部としてモデル検査に関する研修を行っている．

これらの事例から，企業内でモデル検査に興味を持つ技術者が自主的にモデル検査を活用する状況が，急速に拡がりつつあると共に，今後，組織として体系的かつ継続的に適用する段階への移行時期が近づきつつあると考えられる．

モデル検査は，モデル検査器を使った自動検証が行なえることから，数理的技法の中では一般のシステム開発技術者にも判り易く使い易いという特長がある．この特長を活かして，理論面の学習を終えてから実践に入る道筋だけでなく，実際の問題に対して様々な技術レベルでモデル検査を活用しながら，疑問点が出てから学習する，あるいは専門家に相談するような方法が取れれば，短期間で企業への体系的な導入が実現可能である．そのためには，モデル検査全体で必要となる知識の項目や，実践において知っておくべき項目／レベルを明確にしておく必要がある．また，モデル検査を企業として導入するには，自らは実践しなくても導入を決定する立場の経営層に，モデル検査の全体像を明確に示す必要がある．

2.3 官から

現在システムの信頼性に関わる規格・標準の策定が進められており，関連して数理的技法が注目されている．その一つは機能安全に関する国際規格 IEC61508 [10] である．この規格ではシステムの安全度水準 (SIL:Safety Integrity Level) を四段階にわけ，各水準に対してそれを満たすシステムを設計開発する際の要求事項をあげている．安全が最も高く要求される水準 SIL4 では数理的技法のソフトウェアに対する適用が強く推奨されており，産業界への数理的技法導入の動機のひとつとなっている．また，情報セキュリティに関する国際標準 ISO/IEC15408 [11] においても，セキュリティ要件実装の評価保証レベル（EAL：Evaluation Assurance Level）を七段階に分けて規定しているが，EAL5, 6, 7 では（半）形式的記述言語を利用することが要件とされている．

一方，従来手法で扱える規模を越えた製品開発，社会構造の変化などの要因から高度な開発技術とそれを身につけた人材が必要とされている．それを象徴するキーワードとして,「組込みシステム」が近年多用されるようになっ

た.「組込みシステム」に関する教育研修は，大学・民間・公的研究機関など既に様々な場で行われており，情報処理推進機構・ソフトウェアエンジニアリングセンター (IPA/SEC) により組込みスキル標準 [13] が策定されている．但し現状ではスキル標準の中では数理的技法に言及しておらず，組込みシステムの教育研修でも数理的技法を扱うものは殆どない (扱っているものも，ほぼ大学発の活動に含まれる)．

しかし数理的技法を組込みシステムに適用した事例は年々蓄積されており，有効性が次第に明らかになってきている．また，経済産業省は，2006,2007 年度に組込みシステム，信頼性向上，生産性向上，高度 IT 人材育成などをキーワードとして盛り込んだ情報産業に関する幾つかの施策をあげている．特に 2006 年に発表した「情報システムの信頼性向上に関するガイドライン」 [16] では,「自然言語による要求仕様作成作業の誤りを極力排除し，ソフトウェア設計の一層の精度向上を図るため，形式手法（仕様記述言語による仕様記述とモデル検証）及びツール等の適用可能性及び効果等を評価の上，積極的に活用を検討することが望ましい.」としている．さらに，SEC においても，高信頼性システム開発手法に関する検討会を立ち上げ，数理的技法を中心とした技術的な事項に加えて，教育・体制の整備，数理的技法の適用箇所・目的の絞込みなどの検討を行った [14]．2007 年 11 月には，数理的技法を適用して高信頼性システムを開発した事例の紹介を中心とした公開フォーラムを開催している．

数理的技法とその教育は以上の社会的要請と施策の方向性に合致しており，その活動を広く展開していく好機であるといえる．

旧通商産業省の研究所を母体とする産業技術総合研究所では数理的技法の研究と普及を目的として 2004 年度にシステム検証研究センターを設置した．産業界との共同研究を通して数理的技法導入についての成果を着実にあげている一方，技術者を対象としたモデル検査の研修教材を作成している．また産業技術総合研究所は 2008 年 7 月に組込みシステム技術連携研究体を発足させる．この研究体は数理的技法を含むシステム検証を実際の製品開発に適用する設備を提供する他，企業技術者を対象とした教育プログラム「組込み適塾」(「モデル検査」の科目を持つ) を関西経済連合会組込みソフト産業推進会議と共同で運営する (補足 A を参照)．

2.4 関連研究

最初に計算機科学全般の教育に関連する国際会議について述べることにより，どのような学会で教育に関する課題が議論されているかを俯瞰する．計算機科学に関する学会の研究グループとしては，ACM における SIGCSE (Special Interest Group on Computer Science Education) が大きく，様々な教育関連の国際会議を支援している．例えば，以下の国際会議がある．

- SIGCSE (Technical Symposium on Computer Science Education)
- ITiCSE (Annual Conference on Innovation and Technology in Computer Science Education)

また, 大きなソフトウェア工学の会議, 例えば ICSE (International Conference on Software Engineering) においては，特別トラック Education Track が開設され，教育に関する様々な問題点が議論されている．ソフトウェア工学教育に限ると，CSEET (International Conference on Software Engineering Education and Training) がある．1980 年代後期から 1990 年代初期にかけてソフトウェア工学教育は SEI の教育グループにより行われ，SEI の支援により CSEET は 1987 年から始められたという経緯がある [15].

数理的技法の教育に関しては, 以下の会議が定期的に行われている．

- Teaching Formal Methods ワークショップシリーズ ([5, 19, 26])

2008 年には，Formal Methods in Computer Science Education (FORMED) 2008[9] が開催された．これらの数理的技法の教育に関する会議において，これまで数理的技法の BOK に関する論文が発表されたことは無いが,Oliveira により FME-SoE (Formal Methods Europe, subgroup on Education) の一部として行われたものがある [22]. 本論文では，数理的技法に関する授業の内容に関するアンケート結果を基に，網羅的な知識の分類を行っているが，残念ながら体系として完成されたものでは無い．彼との私信によると，FME-SoE においては議論が行われたそうだが，結局，合意に至るのは難しいとの結論に至ったようである．

次に BOK について概観する．ソフトウェア工学教育においては，SWEBOK (Software Engineering Body of Knowledge) [1] が ACM と IEEE Computer Society の合同タスクフォースにより策定されている．BOK は計算機科学以外の世界においても策定されているが，ここでは計算機科学に関連するものだけを取り上げる。計算機科学に関連した BOK において，非常に成功している例の一つが PMBOK (Guide to the Project Management BOK) である．PMBOK は Project Management Institute により開発されたプロジェクト管理技法に関する知識体験である．PMBOK を基にしたプロジェクト管理技術の認定が世界的に実施されており，日本においても行われている．非常に範囲を限定した BOK も開発されており，例えば W. Humphrey により提唱された PSP (Personal Software Process) に特化した BOK も存在する．

BOK は大きな学問単位で策定される場合もあるし，非常に狭い分野，例えば一つの方法論だけで策定される場合もある．しかし，その目的は教育カリキュラム作成のための基礎部分としての知識体系の定式化であり，技術者認定のための技術レベルの評価基準の決定である．ただし，大きな学問単位において策定された場合，内容が一般的すぎてより詳細な技術分野において利用するのには適してない，という欠陥がある．本プロジェクトの動機も，

SWEBOK が我々のプログラムで実施している数理的技法，特にモデル検査に関するプログラムのガイドラインに全く利用出来ないことに起因する．本件については，付録においてより詳細に説明が行われる．

数理的技法自身は非常に広範囲の学問であり，本プロジェクトの目的であるモデル検査はその一部にすぎない．本プロジェクトは将来，策定されるであろう FMBOK (Formal Methods Body of Knowledge）の一部を構成するものになることが期待される．

3 作業計画

第一章に述べたように，本研究はモデル検査の教育プログラムを構築することを目的とし，

- NII, AIST, JAIST, MPS のカリキュラムのうち，モデル検査に関係する部分を詳細に比較し，文書の形にまとめること，
- モデル検査に関する知識領域を明確にし，知識体系 (BOK) として文書化すること，
- 知識領域に基づいて，モデル検査の教育カリキュラムを作成すること，更に各者が既に持っているカリキュラムと比較して位置づけを明確化すること，

の三つの作業を行う．各作業の詳細は以下の節で述べられるが，その前に全体的な計画を述べておきたい．

つくられる教育プログラムはモデル検査の関わる様々な領域・段階に対応することが望ましい．しかし著者らは既に各自で教育活動を進めているので，最初にその内容を確認する必要がある．各自のカリキュラムを詳細に比較し文書化することで本研究の出発点と方向性を明確にすることができる．

モデル検査は多くの分野に関係し，その知識も広い範囲に分布している．様々な領域・段階に対応できるカリキュラムを定めるために，モデル検査自身の知識領域を確認しておく必要がある．研究の第二段階としてモデル検査に関する知識を体系づけて整理し，文書としてまとめる．ただし，全ての知識を体系化することは原理的に不可能であるから，現状における一種の手引書 (Guide to BOK) として作成する．

上記の作業の後に，モデル検査教育カリキュラムを作成する．先に作成した知識体系をもとに，科目毎に講義内容や時間配分など詳細を記したリストを製作する．更に，学習者の目的や背景知識に沿った履修例を提示する．

以下，計画されている各作業の日程を述べる (図 2).

活動計画が具体化し，ロードマップができあがった 2007 年 12 月を起点とする．質の高い Guide to BOK，質の高いカリキュラムを作成するには，早

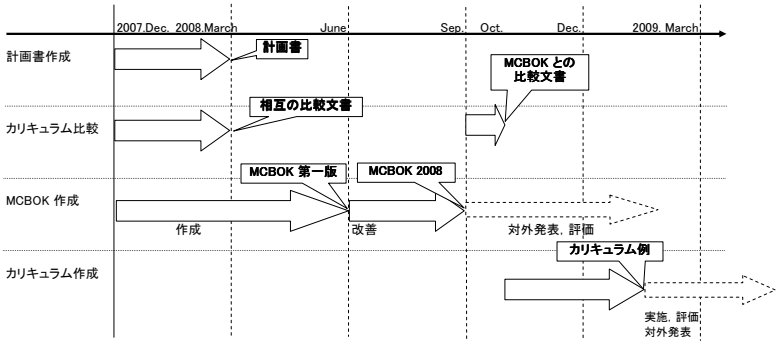

図 2: ロードマップ

い時期から外部の意見を求めることが重要であろう．そのために今回の活動計画を周知させる必要がある．2008 年 3 月までに，活動計画を文書化して公開する (本論文)．既に NII, AIST, JAIST, MPS 四者のカリキュラム比較が終わっており，計画書に追加して記す．

これと並行しながら BOK の作成を行う．2008 年 3 月までに作成方針，方法などを決定し，その後本体を作成する．2008 年 6 月末には第一版を作成し，その後既存のカリキュラムと比較しながら改良していき，9 月末には「モデル検査 BOK 2008 年版」として完成させる．

次に，10 月末までに，著者たちの持つカリキュラムを，完成した BOK を元に再度比較分析する．その結果はその後つくられる予定の，カリキュラム例にも生かされるはずである．

その後，2009 年 3 月までカリキュラムを作成する．この期間のうち，前半に実際のカリキュラム作成を行い，後半にはその検討や他組織のカリキュラムとの比較を行う予定である．作成したカリキュラムを使った教育活動と，その結果をもとにしたカリキュラムの評価，改善はその後の課題として残す．

3.1 カリキュラムの比較調査

現在の状況を把握するために，我々が既にモデル検査教育のために使用しているカリキュラムの詳細を比較する．結果は第四章で報告する．比較に際して，以下で述べるような観点が必要だと思われる．

- 特徴・指針
- 受講者対象・教育にかける期間

講義・研修の中で扱われるトピックやその構成などカリキュラムの詳細は全体的な方針に沿って並べられているはずなので，まず各カリキュラムの全体的な設計を確認する．また受講者対象や期間などの開催形態が内容に影響を与えることも考えられるので，それらを確認したうえで詳細の比較を進める．

実際の内容を比較する際には以下のような点に注目する．

- 扱うトピック
- 使用ツール
- 教科書・資料

各カリキュラムがカバーしている範囲を確認し，カリキュラム同士の重なりをみる．扱うトピックは多岐にわたることが予想されるため，遷移モデル，時相論理，並列プログラムの扱い，検査プロセスなど (ツールの扱いを含めて) 8 つに分類して比較した．扱い方・深さについても言及する．

データとしては， NII, AIST, JAIST, MPS の四者のカリキュラムが集まった．NII のカリキュラム (トップエスイー) に関しては，直接の比較対象として挙がらなかったがモデル検査に関係する内容を持つ講座を参考としてあげた．比較の材料には，各者が用いているテキスト，カリキュラム文書・シラバスを用い，更にワークショップの際の資料を参考にした．

カリキュラム同士の比較によって，必ず教えなければならないトピックに関する知見，共通の指針などが明確になると思われる．これらは後の活動内容であるカリキュラム例作成に役立つであろう．

3.2 モデル検査知識体系の提示

本プロジェクトにおいては，モデル検査に関する知識体系を策定し，提示するのが最終的な目的である．本章においては，BOK を作成するための方法論について述べる．既に様々な BOK が，異なる分野において，様々な目的で作成されているが，その作成方法について述べた文献はほとんど無く，作成のための方法論が確立されていないと言っても過言では無い．

作成に際して最初に考慮する必要がある事項としては，まず最初に何をどのような目的でどのように作成するのかを明確にすることである．

- ゴールの明確化
- 範囲の明確化
- 方法論の明確化

本プロジェクトのゴールと範囲については，既に他の章において明確化されている．

方法論については，用語をどのように分類するか，得られた用語とその分類についていかに検証を行うか，についての考慮を行う必要がある．用語の分類は基礎となる分類法，基準により異なる．一般的な方法論については，[1] における ”Appendix A Knowledge Area Description Specifications for the Ironman version of the guide to the software engineering body of knowledge” において概説されている．用語は知識領域（Knowledge Area）として整理されるが，そこでは，以下の事項に関して標準と要求が述べられている．

- 知識領域内の用語の分類
- 記述方法
- 参照資料の選択
- 関連分野の知識領域の同定
- 知識領域の記述フォーマット

ここで，用語の分類に関して述べられていることを要約すると，以下のようになる．

- Associate editor による各 KA の担当責任分担
- 用語を選択する際の中立性，一般性への考慮
- 複数の候補の提案
- 用語の範囲の考慮（他の分野への横断性，直交性への考慮）
- 用語分類の制限（多くても 2 から 3 階層）
- 用語の樹状の分類

本資料は具体的な方法論まで言及していず，また知識が他の分野へ横断している場合や，直交している場合にどのように取り扱うかについては，結論が出ていないと述べている．

知識領域の同定方法としては，その分野・技術・知識を示すキーワードをいかに探すかといったことが重要になるが，例えばウェッブによる検索や，その分野において代表的，標準的な教科書から用語を抽出する，といった方法が一般的である．しかし，このような作業を自動的に行うといった方法論や技術が開発されていないのが現状である．

本研究プロジェクトの研究課題の一つとして，上記のような方法論を参考にしつつ，BOK 作成の工学（Engineering making BOK) といった，新たな方法論の提案があることをここに明記しておく．

3.3　カリキュラムの提示

教育一般に言えることであるが，教育目標を明示して，それに導く体系的なカリキュラムを策定することは重要である．この共同研究では，モデル検査に関して，そのようなカリキュラムを提案することが目的である．

提案するカリキュラムでは，大学などの教育機関における教育だけでなく，産業界へモデル検査技術を普及させることも意図としている．特に，後者のためには，単に理論から応用まで順番に教育する手法だけでは不十分である．産業界には様々な学問的背景を持った人材が居るため，学習者の習熟度や前提知識に応じた履修方法を示す必要がある．さらに，モデル検査を実践することだけを考えると，必ずしも理論から応用まで一貫して習得する必要はない．そこで，習得する知識に応じた履修方法も明確にすべきである．

本カリキュラムは，MCBOK に基づいて以下の手順で作成する予定である．

1. カリキュラムに含める知識と含めない知識を分類する．

   MCBOK に含まれる個々の知識に関して，２つの観点から議論が必要である．１つ目は，その知識に関する技術や理論が明確になっているかどうかである．例えば，モデル検査の適用に関しては，並行プロセスやプロトコルへの応用は古典であり適用法が明確であるが，それを用いた開発プロセスなどについては不明瞭な部分がある．後者の知識は，現時点では，事例として示すのは良いが，定着した手法として教育するのは避けるべきである．むしろ，研究対象として捉えるべきである．２つ目は，その知識を教育することに意義があるかどうかである．カリキュラムの目的にも依存するが，あまりに高度であったり，極めてマニアックであったりするものについては，教育対象に含めるべきか検討が必要である．

2. 知識を教えるのに必要な時間数を決める．

   個々の知識に関して，その広さや深さを検討して，それを教えるのに割くべき時間を割り当てる．

3. 知識の依存関係を明確に定義する．

   知識には，前提条件や発展学習目標がある．それらの依存関係を検討して，教える順序に関する制約を明確にする．

4. 講義毎に知識をグループ化する．

   １つの講義の時間数を定義し，それに収まるように知識をグループ化する．

5. カリキュラムを作成する．

まずは，講義毎に記述するカリキュラムのフォーマットを作成する．目的，単元，講義方法，時間配分，評価方法など，カリキュラムに示すべき内容を定義する．そして，そのフォーマットに基づいて，4 でグループ化した知識を教育するためのカリキュラムを作成する．

6. ガイドラインを作成する．

   カリキュラムの全体像を示すガイドラインも作成する．このガイドラインには，学習者の習熟度や，習得すべきスキルに応じた履修方法，教育コスト (時間) を明示する．

7. 評価．

   作成したカリキュラムを評価する．現在行われているモデル検査の教育活動に，部分的に組み込んで実施したり，各機関が共同して提案カリキュラムに沿ったセミナーや授業を開催することが考えられる．そして，評価のためのアンケートを作成し，それらを実施後，調査を行い，教育効果について定量的な評価を行う．

カリキュラムは以上の手順に沿って，本共同研究に参加する各機関が分担して作業を行い，定期的に開催する「数理的技法の教育に関するワークショップ」で議論しながら作成する．また，FME の Wiki [8] に,「Formal Methods Courses around the World」という数理的技法に関するコースのサーベイがある．以上のカリキュラムを策定する作業の多くの部分で，このサーベイにあるコースを参考にすることができる．

最終的には策定根拠を持つ，体系的に整理されたカリキュラムを提示することができると考えている．これにより，知識を教えるための道筋とコストが明確になることが期待される．

4 カリキュラム相互比較

共同研究参加組織におけるモデル検査の教授カリキュラムのうち,

- 国立情報学研究所 (NII) におけるトップエスイーにおける講義
- 産業技術総合研究所システム検証研究センター (AIST/CVS) で開発している CVS 教程
- 北陸先端科学技術大学院大学 (JAIST) 組込み大学院
- メルコ・パワー・システムズ株式会社 (MPS) 社内教育

に対して教授内容の比較を行なった．なお比較は 2007 年秋の時点の資料に基づいている．

4.1 各カリキュラムの概要

4.1.1 NII/トップエスイー

国立情報学研究所 (NII) が行なっているトップエスイープロジェクトのモデル検査の講座.

比較に用いたカリキュラムデータ

トップエスイーの Web ページ `http://www.topse.jp/` に掲載さているシラバス, およびトップエスイー向けの教科書のうち次の二冊によった.

- ソフトウェア科学基礎
- モデル検査による設計モデル検証

受講対象, 期間

- ソフトウェア科学専攻の修士課程終了程度の知識をもつ社会人
- 1 講座 12 週

カリキュラム

トップエスイープロジェクトのうち, モデル検査に関係のあるものには次の講座がある.

- 基礎理論
- 形式仕様記述 (基礎編・応用編)
- 設計モデル検証 (基礎編)
- 設計モデル検証 (応用編)
- 並行システムのモデル化と検証
- 性能モデル検証 — 時間オートマトン, UPPAAL
- 実装モデル検証 — Java PathFinder, Assertion

これらのうち, 基礎理論, 設計モデル検証 (基礎編), 設計モデル検証 (応用編) を比較の対象とした. また, 性能モデル検証については概要だけまとめておく.

基礎理論

トップエスイー講座を履修するために必要な予備知識を提供.

第 1-2 週　命題論理, 一階述語論理
第 3 週　集合, 帰納的データ構造, 関係, 写像, 列, 帰納法による証明
第 4 週　並行プログラムの特徴, 同期機構
第 5 週　時相論理学: LTL, CTL, CTL* 意味論, 公理体系
第 6 週　時相論理による様々な性質の定義とその記述方法
到達可能性, 安全性, 活性 性質のパターン記述
第 7 週　モデル検査アルゴリズム 時相論理と妥当性の証明方法
第 8 週　モデル検査の実装: 探索空間の表現, BDD, 状態数削減方法
部分順序還元手法, On-the-Fly モデル検査
第 9 週　有限オートマトン, 正規表現, Buchi オートマトン
第 10 週　モデル検査とオートマトン, LTL とモデル検査, 時間モデル検査
第 11 週　抽象化: 抽象解釈, 静的解釈, 述語抽象化, データ抽象化,
近似化, 過大近似, 過小近似
第 12 週　ツール紹介: SPIN, LTSA, SMV, UPPAAL

設計モデル検証 (基礎編)

- ネットワーク家電の制御ソフトウェアを題材とした産業ソフトウェアの設計モデル検証問題を扱う
- SPIN – Promela
- Assert による検査
- 同期通信
- メッセージ通信 – Sender/Receiver の例
- 停止性の検査 – End ラベル, デッドロック
- progress ラベルによる進行性の検査 – 飢餓状態
- 公平性
- 複雑な性質の記述 – Never Claim
- 時相論理 (LTL)
- LTL 式の never claim への変換
- UML モデルから Java 実装, テスト・デバッグ
- UML モデルから検証モデル記述, シミュレーション
- 検証モデルの検証, 反例分析, モデル修正
- Java プログラムのテスト・デバッグ, 設計モデル修正
- 実習を通じて, 検証モデル記述, 抽象化, 検証式記述, 反例分析のノウハウを修得設計

設計モデル検証 (応用編)

- SMV, LTSA を用いての実習
- 基礎編の例題に対して, SMV および LTSA を用いて, モデル記述, シミュレーション, 検証, 反例分析を行なう

性能モデル検証

- ネットワーク家電の制御ソフトウェアを題材とした産業ソフトウェアの性能モデル検証問題を扱う
- UPPAAL を使用
- UML Profile for SPT による設計とテスト・デバッグ実習
- UPPAAL によるオーディオプロトコル例題の検証

4.1.2 AIST/CVS CVS 教程

産業技術総合研究所システム検証研究センター (AIST/CVS) で開発している, モデル検査研修コース (初級・中級・上級) を比較の対象とした.

比較に用いたカリキュラムデータ

CVS 教程の教科書, スライド (ドラフト分も含む) によった.

受講対象, 期間

初級コースの受講対象者は,

- プログラミングの経験を持つ方
- ソフトウェア開発に従事する技術者
- システム検証に興味を持つ専門学校生, 大学生, 大学院生

であり, 中級は初級修了者, 上級は中級終了者である.
期間は, 集中コースで, 一日 6 時間, 初級で 4 日, 中・上級で 3 日である.

カリキュラムの特徴

- ツールに依存しない, Radical な知識の導入を目指す.
- 実際は SPIN または NuSMV で実習を行なう.
- 初級編 (6 時間 × 4 日) - 初学者・学生が対象
  - モデル検査の概要と作業を理解
  - ツールの基本操作を理解
- 中級編 (6 時間 × 3 日) - 初級編修了者が対象
  - 典型例を扱うのに支障がない程度の理解
  - 合成と抽象化
- 上級編
  - 検証作業を効率よくすすめるための知識
    (モデル検査の動作原理, 適用技術)
  - 中規模演習

CVS 教程 (初級編)

- モデル検査とは
  状態, 遷移, 次の瞬間, 非決定的/決定的
  安全性, 活性 Gϕ, Fϕ
- モデル検査の基礎
  状態と遷移を定めて状態遷移系を書く
  検査する性質: 正しい性質/反例のある性質
  遷移系を目でみて検査
  ツールによるモデル検査
    (モデルの記述, 検査式の記述, 検査の実行, 反例の解析)
- 検査式の記述
  直列回路, 並列回路により, And/Or の導入
  Until operator
- 並行システムと排他制御
  並行システム/セマフォ/Critical Section
  検査: 同時に CS に入らない
  検査: 各プロセスとも CS に入ることが可能
  公平性の問題
- LTL
  パス, パス上の真偽
  様相記号
  Semantics
  LTL 式 を書く演習
- ソースコード検査
  簡単な C プログラム (整数が割りきれるか判定) のモデル化
  テスト技法とモデル検査の違い
- 演習: 自動販売機 (少し規模の大きい演習)

CVS 教程 (中級編)

- 遷移系の合成
  同期合成・非同期合成・通信合成・共有変数を考慮した合成
  演習: 排他制御のアルゴリズム
    Peterson/Dekker/ベーカリー アルゴリズム
- 抽象化
  模倣写像
  抽象化写像
    保存定理と偽反例

データ抽象化

述語抽象化

スライシング

演習: ベーカリーアルゴリズムの抽象化

- CTL による検査式の記述法

  計算木と時間の表現

  CTL 式の定義

  CTL 式の真偽

  よく用いられる CTL 式, 同値条件

  CTL と LTL

CVS 教程 (上級)

- LTL モデル検査のオートマトンによる実現
- CTL モデル検査の BDD による実現
- 有界モデル検査

4.1.3 JAIST 組込み大学院

北陸先端科学技術大学院大学 (JAIST) 組込み大学院の「ソフトウェア検証手法」の講義. この講義は, モデル検査の他に, 定理証明とプログラム検証についてもふれている. ここでは, モデル検査の部分を比較の対象とした.

比較に用いたカリキュラムデータ

JAIST 組込み大学院, 2006 年度「ソフトウェア検証手法」の講義のスライド集によった.

なお, 青木による, 日科技連でのセミナーのスライド集も参考にした.

受講対象, 期間

- 社会人学生を対象とした, 大学院生 (修士, 博士)
- 集中講義
- モデル検査, 定理証明, プログラム検証を含めて 15 講義時間
- なお, 日本科学技術連盟のソフトウェアモデル検査入門は 3 日間 (各 6-7 時間程度)

ソフトウェア検証技法 - モデル検査

- 有限状態モデル – 並行性, 非決定性を含む
- 非決定性
- 並行性

  並行に動作する状態遷移モデルの合成

  メッセージ通信 – 同期通信と非同期通信

- SPIN – Promela
- Assert による検査
- 同期通信
- メッセージ通信 – Sender/Receiver の例
- 停止性の検査 – End ラベル, デッドロック
- progress ラベルによる進行性の検査 – 飢餓状態
- 公平性
- 複雑な性質の記述 – Never Claim
- オートマトン – 有限オートマトン, Buchi オートマトン
- 時相論理 (LTL)
  排他制御, request-acknowledge, starvation-free, 公平性
- LTL 式の never claim への変換
- 時相論理 (CTL*, CTL)
- 状態爆発と抽象化
- 抽象化
  抽象解釈, データ抽象化, 述語抽象化
- 抽象化の理論
  抽象化写像, 保存定理
- プログラムに対する具体構造
- 排他制御, 交互実行, Dekker, Peterson
- スケジューリング – Sleep/Wakeup, 優先度, μITRON RTOS
- セマフォの応用 – Reader-Writer
- 一般探索問題への応用 – 運搬問題, ライツアウト, ナイトの交換
- 状態遷移図への応用 – 変換のバリエーション

4.1.4 MPS 社内教育

MPS (メルコ・パワー・システムズ株式会社) が社内で実施しているモデル検査セミナー (基礎・応用・実践) を比較対象とした.

比較に用いたカリキュラムデータ

KEPCO/MPS にて開発したテキスト

- 基礎コーステキスト
- 応用コーステキスト
- モデル検査器ガイドブック

受講対象, 期間

受講対象者は下記の通り

- 基礎コース:社内のプログラマー/設計者
- 応用コース:基礎コース修了者
- 実践コース:応用コース修了者

期間は下記の通り

- 基礎コース:12 時間
- 応用コース:40 時間
- 実践コース:80〜160 時間 (題材の難易度による)

カリキュラムの特徴

- 現場の技術者が受講後すぐに使えることを目指す.
- 3 つのコース全てで実習重視.
- 実践コースでは実際に開発したソフトウェアを題材にする.
- モデル検査ツールは NuSMV に限定.
- KEPCO/MPS にて開発したモデル検査支援ソフトウェア (NuSMV 用 GUI) による効率化なモデル作成と反例解析の実習がある

基礎コース

- モデル検査の概要
- 状態遷移系
- モデル化の方針
- モデル検査専用言語
- 時相論理
- CTL 式の作成方法 (基礎)
- モデル検査器の操作方法
- 反例解析の方法
- 練習問題, 演習課題

応用コース

- フローチャートからのモデル化の手法
- 状態遷移系とフローチャートとの整合性の取り方
- 時間の概念とモデル化
- プログラムカウンタの付け方
- CTL 式の作成方法 (応用)
- フローチャートの基礎検査
- モデル検査支援ソフトウェアの操作方法
- ラベル付けアルゴリズム
- 演習課題

実践コース

- ソースコードのモデル検査
- 検査対象の選定
- 検査範囲の絞り込みと抽象化
- NuSMV のモジュールの設計
- モデル検査支援ソフトウェアの操作方法
- 実システム適用のノウハウ
- 検査結果報告書の作成方法
- 実システムのソースコードによる実践課題

4.2 カリキュラム比較

UPPAAL 等を用いた実時間モデル検査の実習については, NII の「性能モデル検査」でのみ扱われているので比較の対象には含めなかった.

4.2.1 ツール

Assert 文, End Label などだけを用いて, 時相論理式を用いない検証を, シミュレーション機能による検証とよぶことにする.

- NII

  UML を仕様記述のベースにする

  SPIN, SMV, LTSA

  各ツールの違いを修得

  シミュレーション機能による検証 (Assert, End Label) の実習.
- CVS 教程

  SPIN または NuSMV で演習

  ツール非依存のモデル検査技術の修得を目指す

  シミュレーション機能による検証の解説は無い
- JAIST

  SPIN

  シミュレーション機能による検証 (Assert, End Label) の実習.

  Never Claim の記述の実習.
- MPS

  NuSMV に限定

  各ツールのライセンス形態の説明有り

4.2.2 状態遷移モデル

- 比較項目
  状態・遷移・通信
  プログラムに対応する状態遷移モデル
  モデルの性質の状態遷移による記述 (Never Claim)
- NII
  オートマトン
  すべてカバー
- CVS 教程
  通信を中級で扱っている
  Never Claim については上級で扱う予定
- JAIST
  すべてカバー
- MPS
  状態・遷移 (通信はふれていない)
  プログラムに対応する状態遷移モデルは詳しく説明

4.2.3 並列プログラム

- 比較項目
  非決定性
  相互排除
  セマフォ
  メッセージ通信
  同期通信
  非同期通信
- NII
  すべてカバー
- CVS 教程
  いろいろな例題を通じて, 非同期通信以外はすべてカバー
- JAIST
  すべカバー
- MPS
  非決定性についてふれている

4.2.4 時制論理

- 比較項目
  LTL, CTL, CTL*
  安全性, 活性, 排他制御, 公平性の記述
  LTL/CTL の比較 (記述の得て不得手がある)
- NII
  すべてカバー
- CVS 教程
  CTL* 以外すべてカバー
  LTL 式と CTL 式の変換の問題についてふれている
- JAIST
  すべてカバー
- MPS
  LTL, CTL*に関係するもの以外はすべてカバー

4.2.5 遷移モデルの合成

- 比較項目
  同期合成
  非同期合成
  通信合成
  共有変数と合成
- NII
  検証記述としての合成の扱いはない
- CVS 教程
  すべてカバー
- JAIST
  同期合成, 非同期合成, 通信合成について説明している
- MPS
  同期合成, 非同期合成について説明している

4.2.6 抽象化

- 比較項目
  模倣写像
  保存定理
  抽象化写像
  データ抽象化

述語抽象化
スライシング

- NII
  基礎理論で原理を説明
- CVS 教程
  中級でかなり丁寧に扱う
- JAIST
  スライシングを除いて, ふれている
- MPS
  データ抽象化について説明している

4.2.7 モデル検査の流れ

- 比較項目
  検証モデル記述
  シミュレーション機能による検証
  検査式の記述
  検査
  反例解析, 反例のフィードバック
- NII
  すべてカバー
- CVS 教程
  シミュレーションについては扱っていない
- JAIST
  すべてカバー
- MPS
  シミュレーション以外はすべてカバー

4.2.8 モデル検査の原理

- 比較項目
  LTL モデル検査の原理
  CTL モデル検査の原理
  BDD を用いた場合の変数の順序による効率化
- NII
  基礎理論で説明
- CVS 教程
  上級で説明

- JAIST
  CTL に関してはふれていない
- MPS
  CTL のラベル付けアルゴリズムのみ説明

4.2.9 その他

- CVS 教程では, 有界モデル検査を扱っている.
- MPS では, 実システムのモデル検査の実習を行なっている.

4.3 比較表

比較した内容を簡単に表にまとめておく.

なお, 見出しの「NII MC1」は設計モデル検証 (基礎編), 「NII MC2」は設計モデル検証 (応用編) のこと, 「JAIST 組込」は JAIST 組込み大学院のことである.

	NII 基礎	NII MC1	NII MC2	CVS 初級	CVS 中級	CVS 上級	JAIST 組込	MPS 基礎	MPS 応用	MPS 実践
使用ツール										
Spin		○		○	○		○			
SMV			○	○	○			○	○	○
LTSA			○							
状態遷移モデル										
State Chart							○	○		
Kripke 構造				○	○		○			
Program の遷移モデル				○	○				○	○
Never Claim		○					○			
並列プログラム										
非決定性	○	○		○			○	○		
相互排除	○	○		○	○		○			
セマフォ	○			○			○			
メッセージ通信	○	○			○		○			
同期通信		○			○		○			
非同期通信		○					○			
時制論理										
LTL	○	○		○	○		○			
CTL	○				○		○	○	○	
CTL*	○						○			
LTL と CTL の違い					○		○			
LTL から CTL					○					
CTL から LTL					○					
論理式のパターン										
安全性	○	○		○			○	○		
活性	○	○		○			○	○		
排他制御	○	○			○		○			
公平性	○	○			○		○	○		
遷移モデルの合成										
同期合成					○					○

非同期合成					○		○			○
通信合成					○		○			
共有変数と合成					○					
抽象化										
模倣写像					○		○			
保存定理	○				○		○			
抽象化写像	○				○		○			
データ抽象化	○				○		○			○
述語抽象化	○				○		○			
スライシング	○				○					
モデル検査の流れ										
検証モデルの記述	○	○	○				○	○	○	○
シミュレーション		○	○				○			
反例解析		○	○				○		○	○
検査式		○	○				○	○	○	○
反例のフィードバック		○	○				○		○	○
LTL モデル検査の原理	○					○	○			
有限オートマトン	○	○				○	○			
Buchi オートマトン	○	○				○	○			
Kripke 構造の変換						○				
LTL 式の変換	○					○				
積オートマトン	○					○				
空語判定						○				
CTL モデル検査の原理	○					○				
ラベル付け	○					○				○
論理関数	○					○				
BDD	○					○				
変数の順序	○					○				
有界モデル検査						○				
時間モデル検査の原理	○									
時間オートマトン	○									
実システムのモデル検査										○
	NII 基礎	NII MC1	NII MC2	CVS 初級	CVS 中級	CVS 上級	JAIST 組込	MPS 基礎	MPS 応用	MPS 実践

A 補足

A.1 筆者の関係する教育活動の事例

- トップエスイー (国立情報学研究所)
  - URL: http://www.topse.jp/
  - 「産学融合先端ソフトウェア技術者養成拠点の形成（通称：トップエスイー)」．文部科学省の科学技術振興調整費による人材養成プログラムの一つ．モデル検査に関連した講義としては,「設計モデル検証（基礎編)」,「設計モデル検証（応用編)」、「実装モデル検証」、「性能モデル検証」、「並行システムのモデル化と検証」において，モデル検査の基礎から始まり，UML のモデルの検証，並行 Java プログラムの検証，実時間システムの検証，並行システムの検証などを，SPIN，SMV，LTSA，UPPAAL，FDR 等のツールを用いて講義を行っている．各講義は、12 コマ（１コマ 90 分）で行われている．
- QUBE(九州大学)
  - URL: https://qube.slrc.kyushu-u.ac.jp/
  - システム LSI 設計人材養成実践プログラム．文部科学省の科学技術振興調整費による新興分野人材育成事業の一つ．組込みソフトウェア設計技術コース モデル検査手法－状態遷移モデルとモデル検査－ (A-SW7) で，モデル検査のコースが提供されている．年１回，２日間開催.
- 北陸先端科学技術大学院大学 組込みシステム大学院コース
  - URL: http://www.jaist.ac.jp/
  - 社会人向けに設定された博士前期課程，博士後期課程のコース．講義は東京田町のサテライトキャンパスで行われる．ソフトウェア検証手法 (I477E) でモデル検査を含む講義が提供されている．1 コマ 90 分で，15 コマ開講 (2 単位).
- 北陸先端科学技術大学院大学/日本科学技術連盟共催セミナー
  - URL: http://www.juse.or.jp/
  - 北陸先端科学技術大学院大学と日本科学技術連盟が締結したソフトウェア産業界における人材育成に関する包括協定に基づく．モデル検査に関して３日間開催している．年１～２回開催.
- 日本ソフトウェア科学会チュートリアル
  - URL: http://www.jssst.or.jp/
  - 日本ソフトウェア科学会が主催しているソフトウェア科学技術に関するチュートリアル．これまでに，Spin, UPPAAL などに関するチュートリアルが開催された.
- モデル検査研修コース
  - URL: http://unit.aist.go.jp/cvs/training-course/training-course-top.html

- 産業界の技術者向けに，モデル検査の技能を教えるコース．教材開発のための試行という位置づけで開催していた．初級編 4 日間，中級編 3 日間．

- 組込み適塾
  - URL: `http://www.kansai-kumikomi.net/tekijuku/`
  - 関西経済連合会組込みソフト産業推進会議と産業技術総合研究所が共催する．組込みソフトに関連するソフトウェア技術を一科目あたり半日または一日かけて講義する．アドバンスドトピックスに科目「モデル検査」(一日開催) を持つ．

A.2 既存カリキュラムにおける数理的技法の扱い

数理的技法を既存のソフトウェア工学のコースに統合する場合には，その前提知識や他のコースとの整合性など様々な解決すべき問題点がある．しかし，本プロジェクトにおいてはモデル検査に特化した知識体系を構築し，それを基にカリキュラムの策定を行うことを目的としている．

ソフトウェア工学教育においては，SWEBOK が ACM と IEEE Computer Society の合同タスクフォースにより策定されている．しかし，SWEBOK は一般的すぎ本プロジェクトの参考にはなれど，実際の知識体系の構築には全く不十分である．

SE2004 [15] においては，数理的技法に関するコースとして SE313 Formal Methods in Software Engineering が示されているが，これはいかに数理的技法に関するコースをソフトウェア工学のコースに当てはめるかを考慮したものになっており，数理的技法の教育者から見ると，非常に不完全なものであることが分かる．

国内においては，情報処理学会情報処理教育委員会が J07 の策定を行っている。これは， CC2001-CC2005 を基に、 CS, IS, SE, CE, IT の五分野の教育体系を策定することを目的としている．形式手法に関する SE 領域の知識カテゴリーと領域に知識項目は以下にものである．

MAA はソフトウェアのモデリングと分析にあたり，MAA.md はモデリングの基礎である．CMP はコンピュータ基礎を意味する．

- CMP.fm　　形式手法
- MAA.md.1　モデリングの原則（分解，抽象化，汎化，投影／ビュー，明快性，形式的アプローチの利用など）
- MAA.md.2　事前条件，事後条件，不変表明
- MAA.md.3　数理モデルの仕様記述言語（Z や VDM）の紹介
- MAA.md.4　モデリング言語の性質
- MAA.md.5　モデルの文法と意味（モデルの表現の理解）
- MAA.md.6　明快性（前提が全くない場合，全ての前提を記述する場合）

CMP.fm は形式手法に関する知識分類であり，更に次のように分類される．

- CMP.fm.1 抽象機械の適用（SDL, Paisley など）

- CMP.fm.2 仕様記述言語および技法の適用（ADM, B, CSP, VDM など）
- CMP.fm.3 仕様からのソースコードの自動生成
- CMP.fm.4 プログラム導出
- CMP.fm.5 候補となる実装の解析
- CMP.fm.6 異なる実装と仕様とのマッピング
- CMP.fm.7 詳細化
- CMP.fm.8 正当性の検証

これらをもとに， J07 では形式手法のコースの設計を行った．2007 年度における成果は [12] にまとめられている．上記の説明でも明らかなように，モデル検査については扱われていない．

参考までに，SWEBOK における数理的技法の知識領域では無いが，深く関連する領域を以下に上げる．

- MAA (Software Modeling and Analysis)
  - Modeling Foundation (MAA.md)
- VAV (SOftware Verification and Validation)
- PRO (Software Process)
- QUA (Software Quality)
  - Safety Critical Systems (SAS.sfy)
  - Embedded and real-time systems (SAS.emb)

モデル検査は， MAA.af.2 正しさの分析（例，静的分析，シミュレーション，モデル検査，他），において言及されている．この事から考えても，関連技術分野との相関関係について， MCBOK の策定の際に考慮する必要があることが分かる．

参考文献

[1] Alain Abran, James W. Moore, Pierre Bourque, Robert Dupuis: Guide to the Software Engineering Body of Knowledge 2004 Version SWEBOK. IEEE (2004).

[2] Richard J. Anderson, Paul Beame, Steve Burns, William Chan, Francesmary Modugno, David Notkin, Jon Damon Reese: Model Checking Large Software Specifications. SIGSOFT FSE 1996: 156-166.

[3] Thomas Ball, Ella Bounimova, Byron Cook, Vladimir Levin, Jakob Lichtenberg, Con McGarvey, Bohus Ondrusek, Sriram K. Rajamani, Abdullah Ustuner: Thorough static analysis of device drivers. EuroSys 2006: 73-85.

[4] Béatrice Bérard, *et al.*: Systems and Software Verification: Model-Checking Techniques and Tools, Springer-Verlag, 2001.

[5] Paul Boca, Jonathan P. Bowen, David A. Duce, (editors): Teaching Formal Methods: Practice and Experience. Electronic Workshops in Computing (eWiC), BCS (2006)
`http://cms.brookes.ac.uk/tfm2006/`.

[6] Edmund M. Clarke, Orna Grumberg, Doron A. Peled: Model checking, The MIT Press, 1999.

[7] Edmund M. Clarke, Jeannette M. Wing: Formal Methods: State of the Art and Future Directions. ACM Comput. Surv. 28(4): 626-643 (1996).

[8] Formal Methods Europe web site. `http://www.fmeurope.org/`

[9] FORMED'08.: Formal Methods in Computer Science Education. `http://formed2008.inf.elte.hu/`

[10] IEC 61508.: Functional safety of electrical/electronic/programmable electronic safety-related systems. Bureau Central de la Commission Electrotechnique International, Geneve, (2000).

[11] ISO/IEC 15408.: Information technology - Security techniques - Evaluation criteria for IT security - Part1, Part2 and Part3. ISO/IEC 2005 (2005).

[12] 情報処理学会.: 学部段階における情報専門教育カリキュラムの策定に関する調査研究. 情報処理学会 (2008).

[13] 情報処理推進機構 ソフトウェア・エンジニアリング・センター：組込みスキル標準 2005 年版,
`http://sec.ipa.go.jp/download/files//report/200505/ETSS_skill_Version1.0.pdf`

[14] 情報処理推進機構 ソフトウェア・エンジニアリング・センター：高信頼ソフトウェア構築技術に関する動向調査,
`http://sec.ipa.go.jp/reports/20080606.html`

[15] The Joint Task Force on Computing Curricula, IEEE CS and ACM.: Software Engineering 2004. IEEE CS and ACM (2004).

[16] 経済産業省：情報システムの信頼性向上に関するガイドライン,
`http://www.meti.go.jp/press/20060615002/20060615002.html`

[17] Kim G. Larsen, Christel Baier, Joost-Pieter Katoen: Principles of Model Checking, MIT Press, 2008.

[18] LTSA web site: `http://www.doc.ic.ac.uk/ltsa/`

[19] C. Neville Dean, Raymond T. Boute (editors): Teaching Formal Methods, CoLogNET/FME Symposium, TFM '04, LNCS 3294, Springer (2004).

[20] 村井理恵，服部彰宏，野村秀樹，山本訓稔：Model Checking を適用した実践的非同期制御検証，ソフトウェアテストシンポジウム 2007 東京，Jan.2007.

[21] NuSMV web site: `http://nusmv.irst.itc.it/`

[22] Jose Nuno Oliveira: A Survey of Formal Methods Courses in European Higher Education. TFM 2004: 235-248.

[23] 篠崎孝一，水口大知，石井健志：組込みソフトウェア開発のイン-デザインモデル検査，ソフトウェアテストシンポジウム 2004, Jan.2004.

[24] 篠崎孝一，太田弘，早水公二，星野光勇，今村哲典，吉田雅昭：モデル検査支援ソフトウェアの開発，ソフトウェアテストシンポジウム 2007 東京，Jan.2007.

[25] Spin web site: `http://spinroot.com/`

[26] TFM'03.: Teaching Formal Methods: Practice and Experience. `http://cms.brookes.ac.uk/tfm2003/`

[27] Uppaal web site: `http://www.uppaal.com/`

モデル検査の教育プログラム構築に向けて
(算譜科学研究速報)
発行日：2008 年 8 月 28 日
編集・発行：独立行政法人 産業技術総合研究所 (システム検証研究センター)
同連絡先：〒560-0083 大阪府豊中市新千里西町 1-2-14 三井住友海上千里ビル 5F
TEL：06-4863-5025
e-mail：informatics-inquiry@m.aist.go.jp

An education program of model checking for engineers: Current state and an improvement plan (in Japanese)
(Programming Science Technical Report)
Aug. 28, 2008
(Research Center for Verification and Semantics (CVS))
National Institute of Advanced Industrial Science and Technology (AIST)
5F Mitsui Sumitomo Kaijo Senri Bldg., 1-2-14, Shinsenrinishi-machi, Toyonaka, Osaka 560-0083 Japan
TEL：+81-6-4863-5025
e-mail：informatics-inquiry@m.aist.go.jp

AIST01-J00022-85