モデル検査手法の普及活動とその応用

青木利晃
北陸先端科学技術大学院大学
情報科学研究科

背景

- 形式手法への期待。
  - 信頼性低下に関する危機感。
  - 信頼性保証のためのコストの増大。
  - 経済産業省「情報システム信頼性向上に関するガイドライン」
  - 標準：機能安全(IEC61508), セキュリティ（ISO/IEC15408）
- よく見えない形式手法の全貌。
  - どうやら信頼性を向上させるものらしい。
  - とても広い分野なので見えにくい。
  - これが「形式手法」というものはない

形式手法・検証・モデル検査とは？

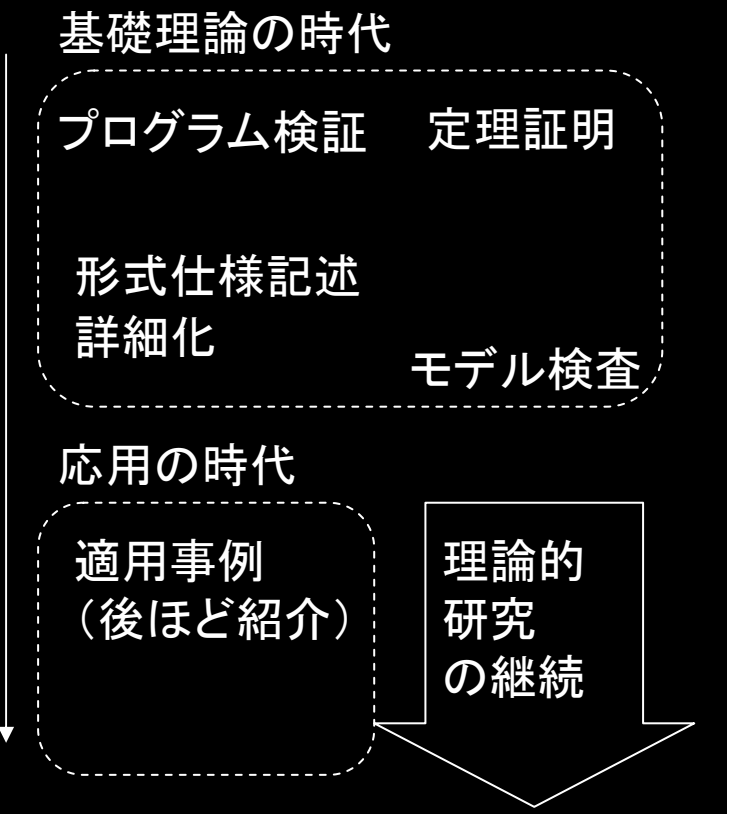
形式手法の特徴

- 形式手法・検証(Formal Methods/Verification)
  - 1960年代からヨーロッパを中心に研究・実践が行われている分野。
  - 数学や論理学に基づいて開発と検証を行う。
    - 科学的な根拠に基づいた開発と検証。
- 柱になっているコンセプト。
  - 厳密に記述して、正しさを保証する。
  - 厳密に詳細化を行って、正しさを保存する。

形式手法の歴史と変遷

- 最初の形式手法。
  - プログラム検証
    - プログラム（フローチャート）の正しさの証明。
- 上流工程の取り扱い。
  - プログラムの大規模化。
  - 形式仕様記述
    - 厳密に仕様を記述する。
    - 仕様の正しさを証明する。
- 上流工程とプログラムを接続。
  - 詳細化。
    - 正しさを保存したまま詳細にする。
- 数学・論理的問題の取り扱い。
  - 定理証明、モデル検査など。
  - 数学・論理的な証明の支援、および、自動化。

1960
1970
1980
1990
2000



形式手法の歴史と変遷

- 古典：検証(Verification)について。
  - テスト→妥当性(Validity)
    - 限られた場合（テストケース）について正しいかどうか検査する。
    - 誤りが無いかどうかは保証できない。
  - 検証→正当性(Correctness)
    - どのような場合でも正しいことを保証する。
    - 証明。
- 近代的：検証(Verification)について。
  - 正当性を保証するというよりは、
    - 科学的な手法を用いて誤りを検出する。
    - 科学的な根拠に基づいて、部分的な正当性を保証する。

代表的な適用事例

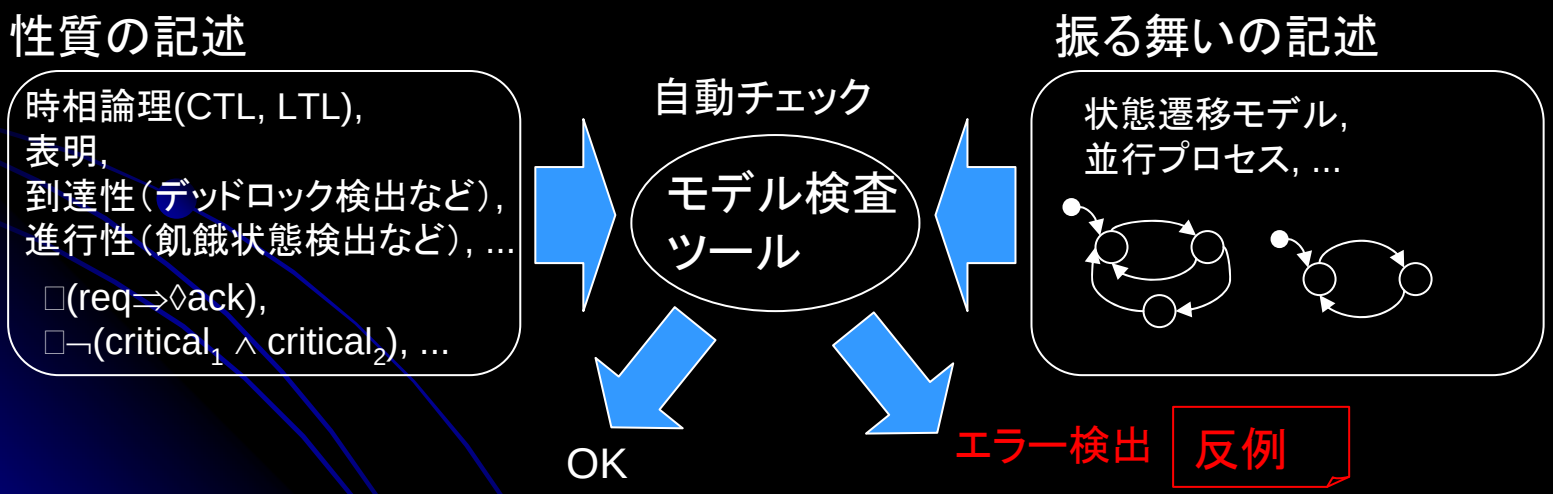
- 多くの適用事例
  - Edmund M.Clarke and Jeannette M.Wing：Formal Methods: State of the Art and Future Directions, ACM Computing Surveys, 1996.
    - 回路設計、データベース、プロトコル、標準、航空宇宙、企業インフラ、などなど、相当数。
  - 最近、日本でも事例が出てきた。
    - フェリカ、コピー機、デジタルスピードメーターなど。

期待する性質の保証手法

- 数学・論理的問題を解くための手法。
  - モデル検査手法：考えられる状態をすべて自動的に探索する。
    - 有限状態で特徴づけられるプロセスの並行動作にまつわる性質など。
    - 記述能力が低いが、自動的。
  - 定理証明手法：対話的に証明する。部分的に自動化を行う。
    - 述語論理で書かれた式の正しさなど。
    - 記述能力が高いが、対話的、もしくは、部分問題の自動化。
- 仕様・設計・プログラムに関する任意の性質の保証は、自動的にはできない。
  - 性質を限定、対象を限定、対象を抽象化することにより、自動化する。
  - 定理証明手法などにより、直接的に取り扱う。

モデル検査概説

- モデル検査手法：考えられる状態をすべて自動的に探索する。
  - 有限状態で特徴づけられるプロセスの並行動作にまつわる性質など。
  - 代表的なツール：NuSMV, Spin, LTSA, UPPAAL, etc.
- モデル検査の「モデル」≠UMLなどの「モデル」
  - 論理学では、与えられた性質をp、対象となる振る舞いをMとすると、「Mに対してpが成立する」ことを、「MはpのモデルMである」と呼ぶ。
  - モデル検査：振る舞いが記述した性質のモデルであることを検査する



時相論理(Temporal Logic): 時間に関する性質を取り扱うことができる論理
反例（Counter Example）: エラーに導く実行方法

教育と導入

普及活動

- モデル検査の普及活動、教育活動。
- 形式手法理解の入り口としてのモデル検査手法。
  - 多くの前提知識を必要としない。
  - 簡便なツールが存在する。
  - 注目されている組込み分野で有効。
  - 限界がある→ほかの手法に興味が湧く。
- 情報分野の実問題を解決するには？
  - 企業から問題の詳細を出してもらう。
    - すぐには難しい。失敗。
  - 企業内に形式手法を理解している人をつくる。
    - メリットは、最終的には、大学にかえってくる（共同研究など）。はず。。。

普及活動

- これまでモデル検査手法の普及活動を行ってきた（200名以上）。
  - 試行
  - 日本科学技術連盟セミナー
  - 人材養成プログラム（QUBE、TopSE）
  - 社会人向け授業（JAIST組込みシステムコース、一部）
  - 社内研修

普及活動比較

	アンケート数	理解度	有効性	実践できる
産総研（初級）	75	59（79%）	70（93%）	N/A
産総研（中級）	23	7（29%）	24（100%）	19（79%）
NII（基礎）	36	22（61%）	N/A	25（69%）
NII（応用）	7	5（71%）	N/A	5（71%）
JAIST（日科技連）	81	74（91%）	76（94%）	71（88%）

普及活動

- 教育目標。
  - 応用法を考えることができるくらいの知識をつける。
    - 対象となる問題はドメインやプロジェクトにより異なる。
  - 有効そうで典型的な問題を対象。
    - 奇をてらわない。
- 演習中心のセミナー。
  - PCを持ってきてもらう。
  - 約100個のサンプル。
  - 並行性と非決定性の問題に焦点を当てる。
    - スケジューリング、資源管理など。

普及活動

- ツールの使い方を理解するのは容易。
  - ２～３日間のセミナー。
    - ２日間では少し時間不足気味。
  - 受講者のバックグランドにより理解度には差があるものの、全く理解できていない人はいない。
  - 検証能力に驚く人と、導入は難しいと思う人に2分化。
    - 属している部署や対象ドメインにより異なる。
    - 技術自体は正しく理解できている（判断はできているから）。
- 並行性や非決定性の問題を再認識。
  - 並行性に関するテストをセミナーの前後で実施（試行）
  - 注意点がわかるようになる。

普及活動

- 普及活動の普及活動が必要。
  - これまで1人で行ってきたが、1度に20人くらいの指導が限界。
  - 普及活動を行える人を育成する必要がある。
  - アプローチを検討中。

使い方の検討

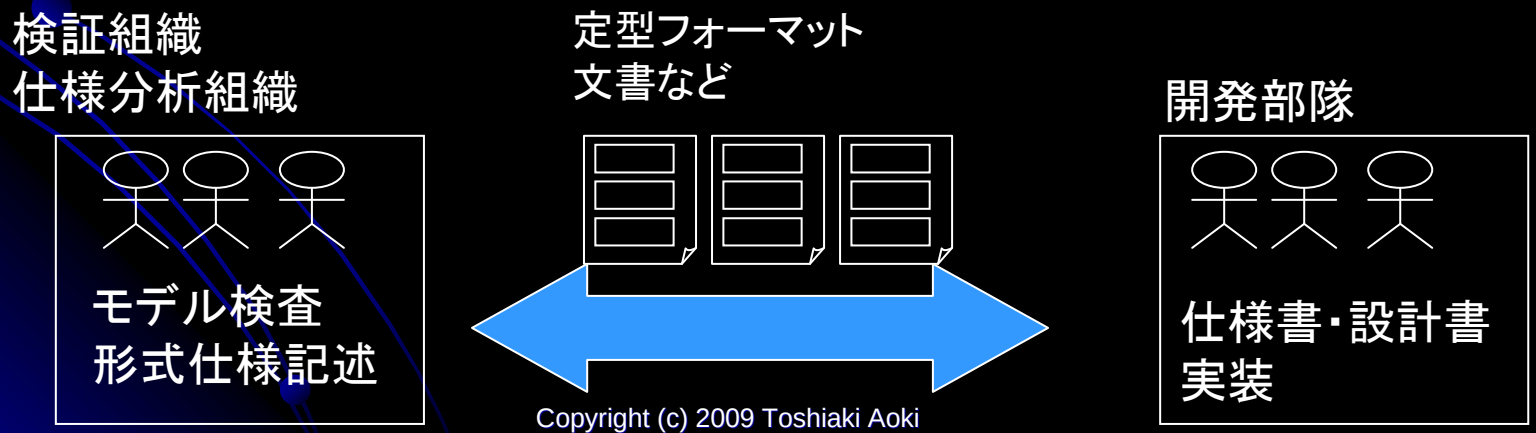
- 技術の特性を考慮して、適材適所。
  - 形式仕様記述
    - おそらく、開発プロセスに組み込まなければ意味がない。
  - モデル検査手法・定理証明手法
    - 開発プロセスに組み込んで使う。
      - 設計モデリングやソースコード解析への適用。
    - アドホックに使う。
      - 各々の開発者が、電卓のように使う(デザイン電卓)。
      - 考えにくい部分を補う。
- 最初から、大風呂敷を広げて導入しない方が良いような気がしています。
  - アドホックに使うことから始めるのがよさそうな印象。

方法論の整備(技術の組み合わせ)

- 形式手法を使うことを前提とした方法論の整備が必要。
  - 通常どおりに開発を行っていた→開発の中盤で突然モデル検査を使って検証をする→無理に決まっている。
  - 形式手法を適用できるように文書化、モデル化しなければならない。
  - 形式手法を適用できるように体制や管理をしなければならない。
- 他技術との連携。
  - 形式手法は、今の開発手法すべてにとってかわるものではない。
    - モデル検査手法や定理証明手法を導入しても、テストが全くなくなるわけではない。
    - 適用可能な部分が自動化、効率化される。
  - 適用できる部分と適用できない部分の見分けが必要。
  - 周辺技術との連携が必要。
    - オブジェクト指向モデリングとの関係は？アーキテクチャ設計との関係は？・・・

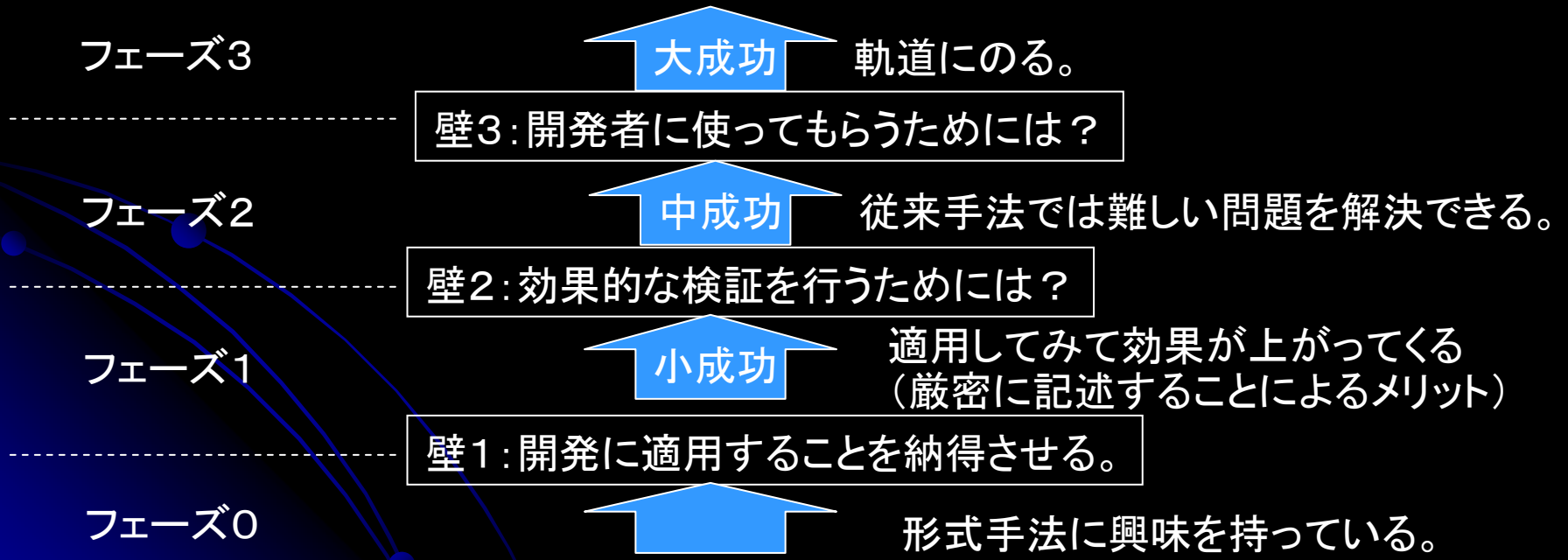
技術者教育

- 技術者全員が知らなければならない技術か、一部の人が知っていれば良い技術か。
- 組織構造に依存する。
  - 検証組織、仕様分析組織などを作るのであれば、連携する技術者とのインターフェースが確立されていれば良い。
    - AT&Tの例：５人の検証エンジニア。



教育と導入

- 積極的な企業や部署は評価や導入をはじめている。
- 導入の困難さ(感触)。
  - いくらかの壁が段階的にやってきそう。
    - 開発体制やドメインの特徴によるいろいろな解がありそう。



まとめ

- 形式手法
  - 古くから研究、実践がされている。
  - 研究領域から実践領域へ。
- ノウハウの蓄積とソフトウェア工学の観点からの整理の必要性。
  - 他技術との併用。
  - 適用箇所、適用工程。
  - 解決対象問題の明確化と解決法の蓄積。
- 最初の呼び水として、
  - モデル検査手法（Spin, SMVなど標準的なもの）
  - 形式仕様記述（VDM,Z, B）

  などから使ってみてはいかがでしょうか？
  - 成功したり、興味が出てくると、より、高度な手法に手が出るはずです。