なぜ形式手法か(A)

SEC形式手法人材育成作業部会(編)

このモジュールについて

品質の高いソフトウェアの効率よい開発へ向け、形式手法の有用性を理解し導入に前向きに検討する姿勢の獲得を意図したモジュール

- 事前知識・経験
  - 現状のソフトウェア開発に対する問題意識
  - 形式手法の知識・スキルは不要
- 学習目標
  - 形式手法の概略レベルの知識を得る
  - 旧来の開発の課題への形式手法の有用性を理解し導入の検討を開始できる
- 主な学習項目
  - 形式手法の定義と成果例
  - 旧来のソフトウェア開発の課題と形式手法
  - 信頼性・安全性への期待、国際標準・規格、など
  - 慣習的他手法の限界と、その解決に有用な形式手法の特性

形式手法(Formal Methods)とは

- 形式手法とは、システム開発に用いられる手法で、数理論理学に基づく科学的な裏付けを持つ仕様記述言語を用いて設計対象を表現することにより、ある側面の仕様を厳密に記述し、開発の各工程で利用する手段の総称である
- 形式記述を行うことで曖昧さが取り除かれ、機械処理が可能になり、様々な可能性が開ける
- 形式手法の一分野であるモデル規範型の手法と、形式仕様記述言語、仕様の記述や検証のためのツールの活用により、厳密な仕様の記述や検証が可能となる上、形式手法の導入は、システム開発におけるステークホルダ間のコミュニケーションの活性化に寄与する

形式手法による仕様作成の成果例

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

	対象システム	適用工程	仕様規模	欠陥数	生産性	開発期間	備考
SCSK	証券業務	実システム UseCaseレベル 要求仕様	3+3万行 仕様+テストケース	0	2.5倍	45%	開発チームに業務知識無し
フェリカネットワークス	おサイフケータイファームウェア	実システム API外部仕様	7.4+6.6万行 仕様+テストケース	0	2倍	85%	開発チームに形式手法知識無し
産業技術総合研究所,オムロン	鉄道関係駅務システムデータ	既存システム 厳密仕様 作成実験	0.75+80万行 仕様+業務	欠陥発見数 29			「暗黙知は仕様の欠陥」

私たちの現実

- ソフトウェアの信頼性や安全性に対する関心が高まってきた
- これまで以上に効率良く開発することが求められてきている

- 普通の現場で開発に携わっているのは普通のチームとメンバ
- 現場にプレッシャーをかけても品質は高くならず、ストレスばかりがかかる

- 効率良く品質の高いソフトウェアを開発するための「銀の弾丸」はない

社会基盤としての計算機システム

- Safety Critical System
- Mission Critical System
  - 原子力発電所制御システム
  - 航空管制システム
  - 鉄道信号制御システム
  - 放射線治療システム
- 日常社会生活
  - 銀行オンラインシステム
  - ワープロ
- ディペンダビリティ（dependability）
  - 信頼性、安全性、頑健性、説明可能性、等々

ソフトウェアの不具合の例

- 鉄道改札機
- 東京証券取引所
- みずほ銀行
- 航空管制システム
- 搭乗手続きカウンタ
- 電話交換機
- 携帯電話リコール

etc.

情報技術セキュリティ評価基準ISO/IEC15408

経緯

- 欧米各国独自の情報技術セキュリティ評価基準では、相互認証は困難であったため、1999 年に国際標準として採択
- 日本では:2000年に、JISX5070　として採択

範囲

- セキュリティ製品（ハードウェア・ソフトウェア）およびシステムの開発や製造，運用
- 製品やシステムが備えるべきセキュリティ機能に関する機能要件と，設計から製品化に至る過程でセキュリティ機能が実現されていることを確認する保証要件を網羅した要件集
- セキュリティについての基本概念の説明とともに，評価対象となる製品やシステムのセキュリティ基本仕様を記述するセキュリティターゲット（ST）や，ST のベースとなる文書であるプロテクションプロファイル（PP）についても規定

ISO/IEC 15408 の構成と評価保証レベル

- 以下のパートから構成
  - Part 1. 総則および一般モデル
  - Part 2. セキュリティ機能要件
  - Part 3. セキュリティ保証要件
- Part 3 では評価保証レベル Evaluation Assurance Level (EAL) として，7 段階のレベルで保証要件が規定されている
  - EAL 1: 機能テストの保証
  - EAL 2: 構造テストの保証
  - EAL 3: 系統的テストおよび確認の保証
  - EAL 4: 系統的計画およびテスト，レビューの保証
  - EAL 5: 準形式的設計とテストの保証
  - EAL 6: 準形式的な設計の検証とテストの保証
  - EAL 7: 形式的な設計の検証およびテストの保証
- EAL 1～3 が一般向けで、EAL 4 が政府機関向け、EAL 5 ～ 7 が軍用・政府最高機密機関レベル

電子機器の機能安全に関する国際規格IEC61508

- リスクの事前評価=「絶対安全は存在しない」
- リスクの軽減=許容範囲内におさめる
- IEC61508:2000
  - Functional safety of electrical/electronic/programmable electronic safety-related systems
- JISCO508:2000
  - 「電気・電子・プログラマブル電子安全関連系の機能安全」
- 安全尺度 SIL (Safety Integrity Level)

SIL	信頼性	連続稼動設備における1時間あたりの故障発生率
1	90%以上	10^{-6}以上10^{-5}以下
2	99%以上	10^{-7}以上10^{-6}以下
3	99.9%以上	10^{-8}以上10^{-7}以下
4	99.99%以上	10^{-9}以上10^{-8}以下

IEC61508関連の規格

現格種別	境格名称	対象
EN規格	EN61548	一般(計装)
	EN50126	鉄道
IEC規格	IEC6151	プロセス産業
	IEC61513	原子力
	IEC62061	産業機械
	IEC62304	医療
	IEC62800	モータ
ISO規格	ISO/WD26262	自動車
指針類	EEMUA	アラーム指針
	DFT STAN	防衛(英国)
	海軍規格(英国D	海軍(英国)
	MISRA-SA	自動車(英国)
	計量ソフトウェア指針	モータ
	EMCガイドライン	IEE(英国)

形式手法の推奨

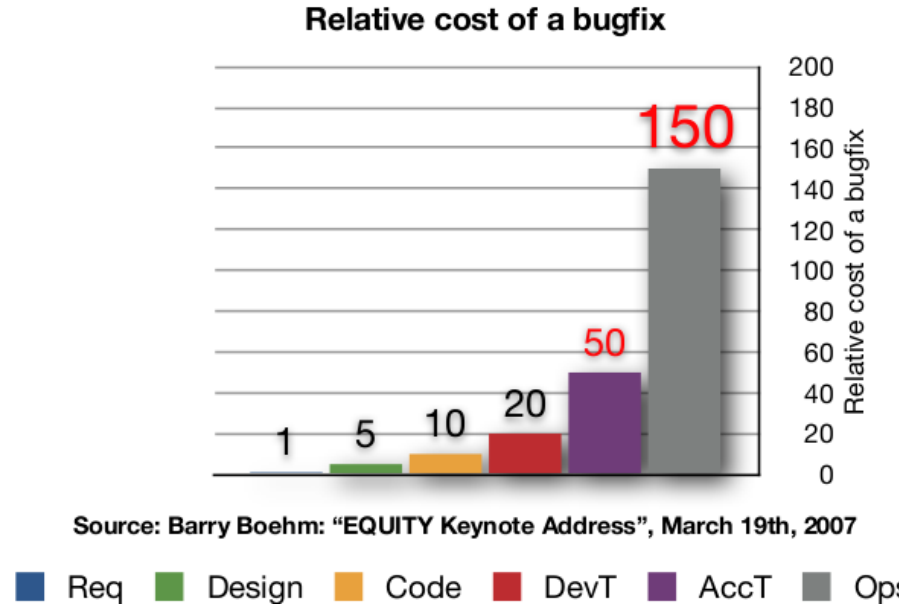
表 3　IEC61508 における形式手法の推奨例（R: Recommended, HR: Highly Recommended）

Software safety requirements specification					
Technique/Measure	規格の参照先	SIL1	SIL2	SIL3	SIL4
1 Computer-aided specification tools	B.2.4	R	R	HR	HR
2a Semi-formal methods	表 B.7	R	R	HR	HR
2b Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	—	R	R	HR
Software design and development: detailed design					
Technique/Measure	規格の参照先	SIL1	SIL2	SIL3	SIL4
1a Structured methods including for example, JSD, MASCOT, SADT and Yourdon	C.2.1	HR	HR	HR	HR
1b Semi-formal methods	表 B.7	R	HR	HR	HR
1c Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	—	R	R	HR
2 Computer-aided design tools	B.3.5	R	R	HR	HR
3 Defensive programming	C.2.5	—	R	HR	HR
4 Modular approach	表 B.9	HR	HR	HR	HR
5 Design and coding standards	表 B.1	R	HR	HR	HR
6 Structured programming	C.2.7	HR	HR	HR	HR
7 Use of trusted/verified software modules and components (if available)	C.2.10 C.4.5	R	HR	HR	HR

（出典: IEC61508 Part1 Annex A より作成）

なぜ形式手法か？理論的根拠

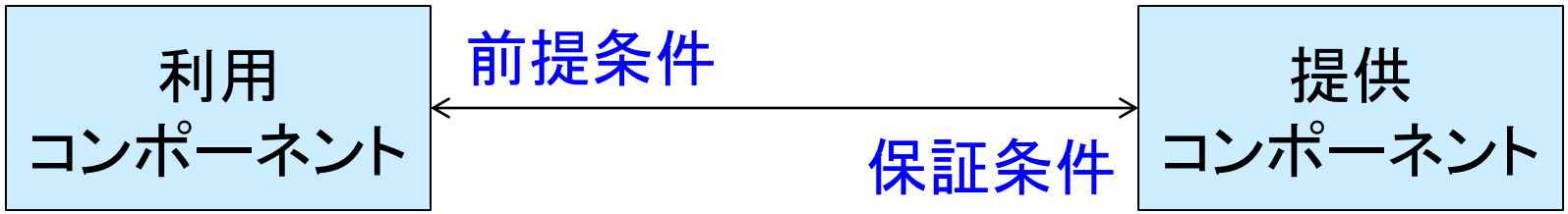
- プログラムは数学系だから
  - しかも，CやJavaのプログラムは「検証が困難な数学系」
  - 要求仕様記述工程からプログラミング工程までのどこかで，数学系に変更しなければならない
  - 検証しやすい数学系で記述した方が，品質を上げやすい
- 形式手法では、上流工程で仕様を検証
  - 欠陥修正コストが10分の1から200分の1で済む([Boehm:07]他)
- 旧来の手法では、検証が困難
  - レビューだけでは限界がある



形式手法とアーキテクチャ

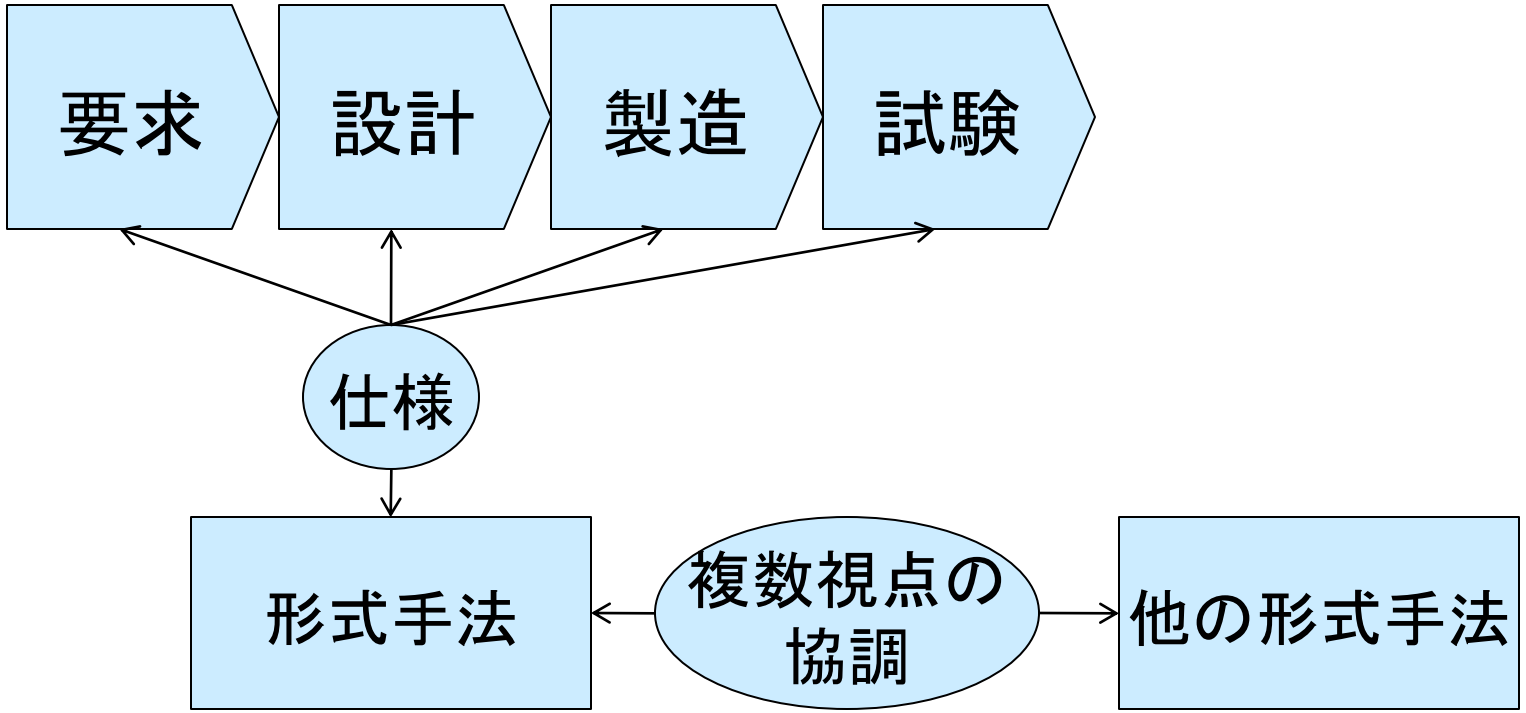
- アーキテクチャの特性と相互作用に対して形式手法を活用できる

検証対象	形式手法の活用方法の例
特性	構造モデルで構成要素を定義 構成要素の操作と状態に基づいて，振舞いをモデル検査
相互作用	利用コンポーネントの前提条件に対して，提供コンポーネントの保証条件による充足性を検証



開発プロセスと形式手法

- どの開発工程に形式手法を適用するか？
- 適用工程と他の開発工程との関係をどうするか？
- 適用工程内で，形式手法と他の手法の関係をどうするか？
- 複数の形式手法を用いる場合，どのように組み合わせるか？



ソフトウェアの機能性・信頼性の確認方法

- ソフトウェアのレビュー
- テスト
- 検証、証明

ソフトウエアのレビュー

- 開発の各工程ごとに成果物を検査・評価すること
  - ウォークスルー(walk-through)
    - 開発者が進行役となって実施
    - エラーの早期発見が目的
    - 修正の検討と確認はしない(開発者の責任)
    - 資料を事前配付
  - インスペクション(inspection)
    - モデレータが進行役となって実施
    - エラーの早期発見と修正の検討が目的
    - 修正結果まで追跡確認(モデレータの責任)
    - ウォークスルーよりは形式的
    - 資料を事前配付
  - ラウンドロビン(round-robin)
    - レビューメンバが持ち回りでレビュー責任者を務める

テストとは?

- 「プログラムのテストは、バグの存在を示すには極めて有効であるが、バグが存在しないことを示すには絶望的に無力である」[E. Dijkstra]
- 不具合の検出
- 信頼性に対する確信度
- 品質指標を用いたテスト項目の作成
  - 要求品質
    - 顧客に依存
  - 品質指標で危険の掘り下げ
    - 品質特性の分類
  - テスト項目
    - 漏れなく、顧客が認める評価項目

ソフトウエア検証 (Software Verification)

- 「納得できる正当性の証明が、必要な確信のレベルに到達する唯一の方法であるように思われる」[E.Diijkstra]
  - 公理系に基づく証明
  - 証明の記述、チェック、理解、評価
  - 「プログラムが大規模・複雑になればなるほど、検証が有効・重要になる(何故なら、テストではいよいよ解決できない)」
- 形式手法の基礎
- 理論体系の構築
  - 対象=システム、性質
  - 理論と現実
- 証明することの副産物
  - 概念の明確化
  - 曖昧さのない議論と記述

形式手法適用レベル

[http://en.wikipedia.org/wiki/Formal-methods]

- 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し、証明や分析までは行わずに、この記述を基にしてプログラムを開発する
- 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

もっと詳しい形式手法の説明

- 数理的な道具建/方法に基づくシステムの開発方法
  - 記述、分析、設計、実現、検証、保守
- 30年以上の歴史
- 多種多様な理論、方法論、ツール例えば、J.Bowenのwebでは、100種類以上
  [http://formalmethods.wikia.com/wiki/Formal-methods]

形式手法の特質

- 形式手法に対する認識
  - J. A. Hall:
    Seven Myths of Formal Methods,
    IEEE Software,Vol.7,No.5,pp.11-19,1990.
  - J. P. Bowen and M. G. Hinchey
    Seven More Myths of Formal Methods,
    IEEE Software,Vol.12,No.4,pp.34-41,1995.
- 欧米では基本的素養
  - 相互理解、議論/分析の道具建て
  - 実用の段階
  - 記述と分析の多面性
  - 種々の理論･ツールの併用、統合

サマリー

品質の高いソフトウェアの効率よい開発へ向け、形式手法の有用性を理解した上で、形式手法の導入に前向きに検討する姿勢の獲得を目標に、主に以下を学習

- 形式手法の定義と成果例
- 現状のソフトウェア開発の課題と形式手法
  - 信頼性・安全性への期待、国際標準・規格、など
  - 慣習的他手法の限界と、その解決に有用な形式手法の特性