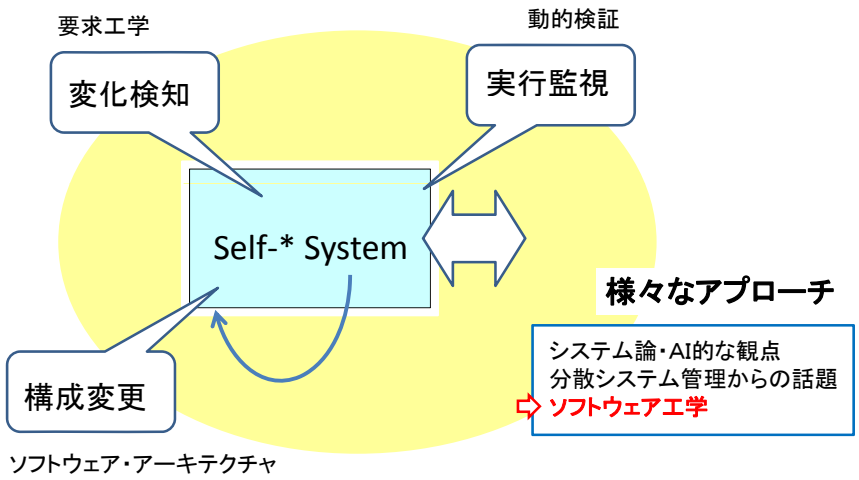
大学院特別講義（平成24年7月19日）

自己適応システム

ー最近のソフトウェア工学研究からー

中島　震
国立情報学研究所
総合研究大学院大学・情報学専攻

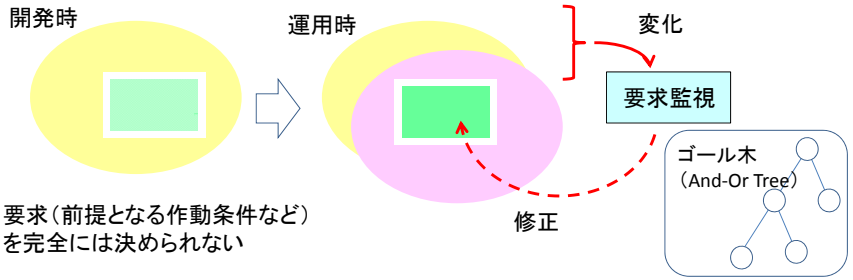
自己適応システム



内容

- 研究の流れ
  - 要求監視とモデルベース適応
- Webアプリケーションの事例
- 関連分野と今後の動向

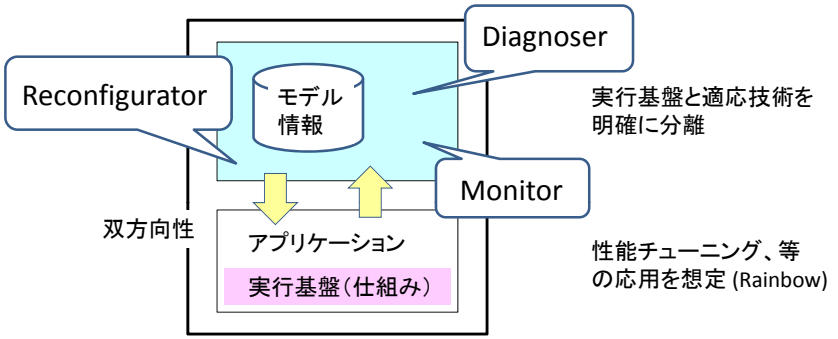
Requirements Monitoring



S. Fickas and M. Feather : Requirements Monitoring in Dynamic Environments, Proc. RE'95, 1995, pp.140-147.

A. Dardenne, A. van Lamsweerde, and S. Fickas : Goal-directed Requirements Acquisition, *Science of Computer Programs*, vol.20, pp.3-50, 1993.

モデルベース適応



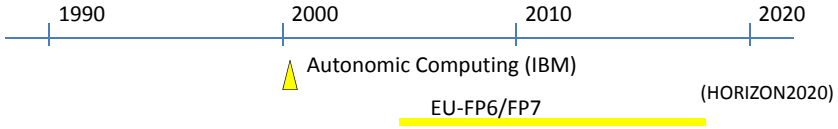
実行基盤と適応技術を
明確に分離

性能チューニング、等
の応用を想定 (Rainbow)

F. Dalpiaz et al : An Architecture for Requirements-Driven Self-reconfiguration, Proc. CAiSE 2009, 2009, pp.246-260.

D. Garlan et al : Rainbow: Architecture-based Self-Adaptation with Reusable Infrastructure, *Computer*, 37(10), 2004, pp.46-54.

3つの流れ



要求	要求監視 (Fickas) → GOREとの関係 確率過程(Ghezzi) Webサービスへの適用 (Ghezzi, Mylopoulos) 動的SPLE (Hallsteinsen)
アーキテクチャ	モデルベース適応 (Garlan) 動的ADL(Kramer)
実行監視	実行時検証(NASA-JPF) → 有限列の性質 統計モデル検査(CMU, INRIA)

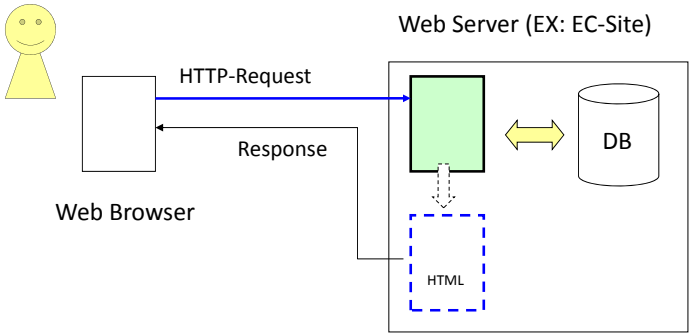
内容

- 研究の流れ
- Webアプリケーションの事例
  - ECサイトの例題
  - 自己適応システムの事例として
- 関連分野と今後の動向

中島震：自己適応Webアプリケーションシステム：概念アーキテクチャと実現フレームワーク，コンピュータソフトウェア，2012年8月

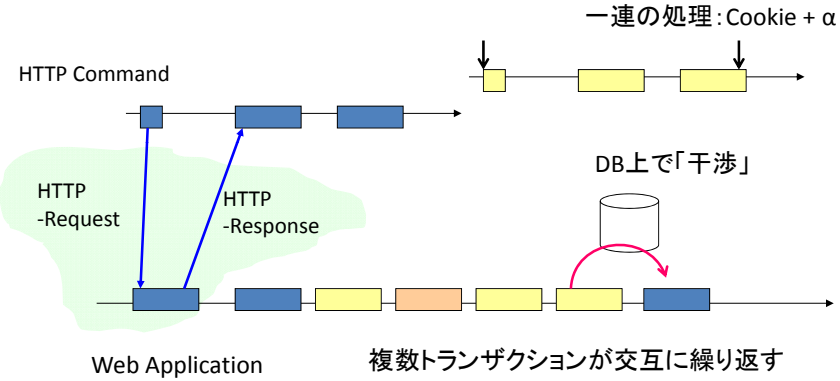
Web アプリケーション

オープンなシステム --- 様々なクライアント

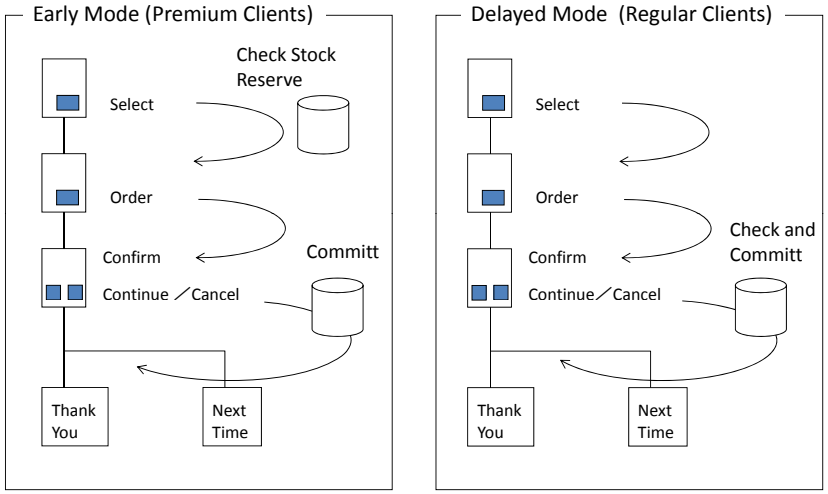


長時間トランザクション

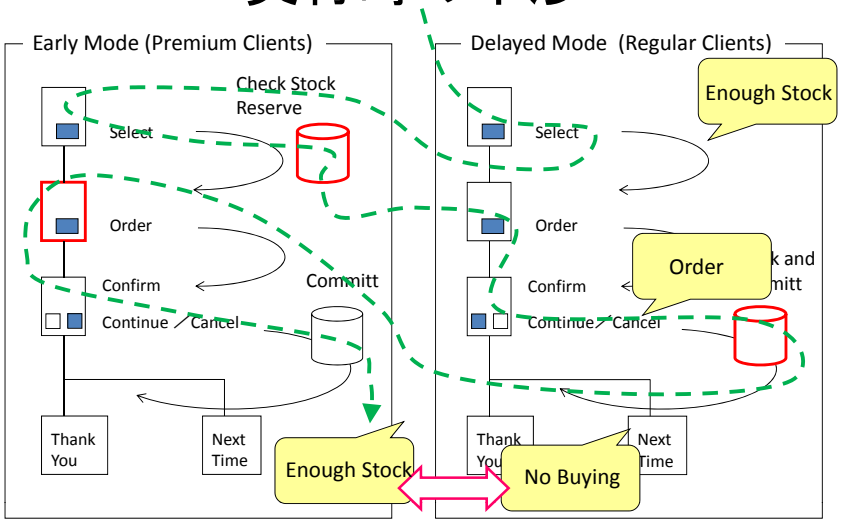
トランザクション＝HTTP コマンド列 (HTTP-Request/Response)



ECサイトの例



実行時の干渉

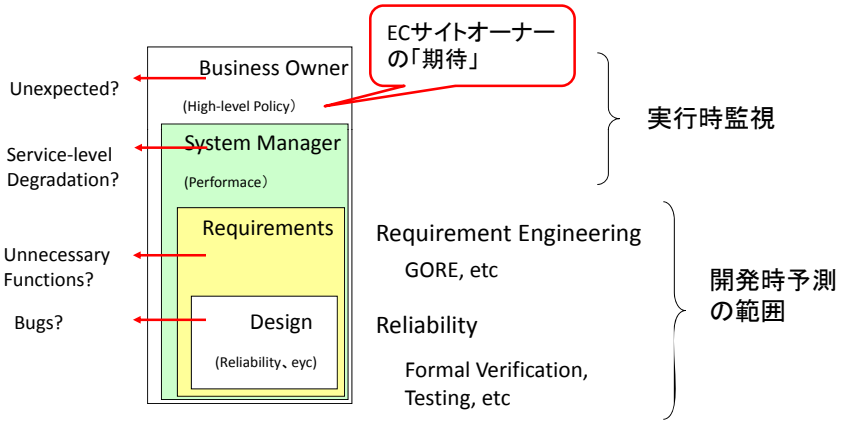


状況の整理

- 不具合
  - 一連の処理終了時の状態
    - 「Enough Stock」なのに「Cannot Buy」
  - EC-サイト･オーナーの期待に反する
    - 「Enough Stock && Want-To-Buy && not(Benefit)」
- システムのバグではない
  - 注文処理の方法に誤りはない
- 本当の理由
  - 2つのサービス機能が「干渉」
    - ビジネスゴールを達成 → サービスレベルの細分化
  - 特定のプレミア会員の振る舞い：要求分析時に予測不能
    - Reserve → Cancel

開発＝予測

ステークホルダからみたディペンダビリティ

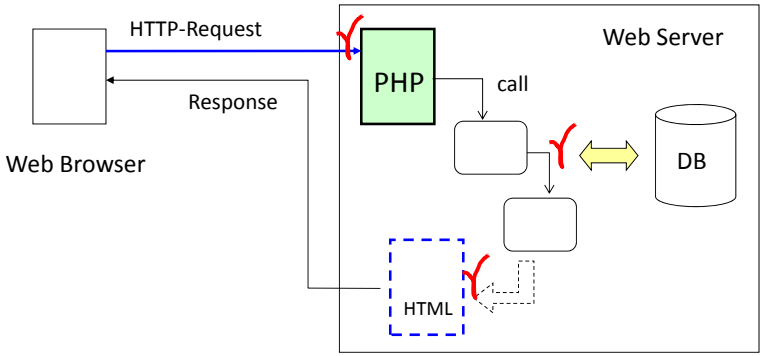


自己適応の考え方を導入

- 実行監視
  - システムの実行履歴 → 好ましくない状況に陥るか？
  - 監視性質の表現法 → 有限状態オートマトン
    - トランザクションは「開始」と「終了」が明確
- 機能変更
  - 機能の縮退 → もっとも簡単な変更
    - プレミア会員向けサービスを一時的に中止
  - 機能の復活
    - 好ましくないアクセスがないことをタイムアウトで検知

監視ポイント

PHP ページ = 変換子 [(HTTP, DB) → (HTML, DB)]



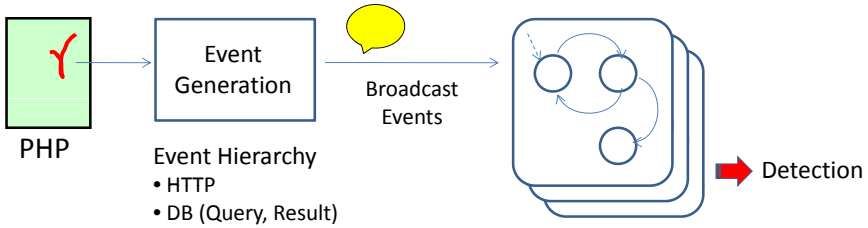
監視オートマトン

- 対象
  - Webページの遷移列：どのWebページが参照されたか？
  - PHP関数の起動列：どの関数が起動されたか？
- 表現力
  - いずれも有限状態オートマトンで表現可能

F. Ricca and P. Tonella : Analysis and Testing of Web Applications, Proc. 23rd ICSE, pp. 25-34, May 2001.

K.M. Olender and L.J. Osterweil : Cecil: A Sequencing Constraint Language for Automatic Static Analysis Generation, IEEE TSE, 16(3), pp. 268-280, 1990.

イベント生成

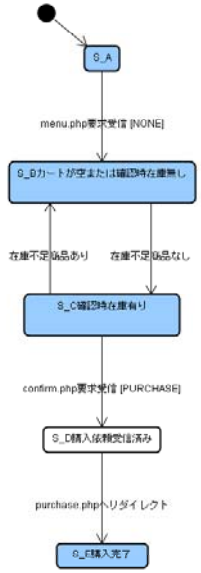


Event Hierarchy
- HTTP
- DB (Query, Result)
- HTML
- Transaction

一般的には複数監視オートマトン

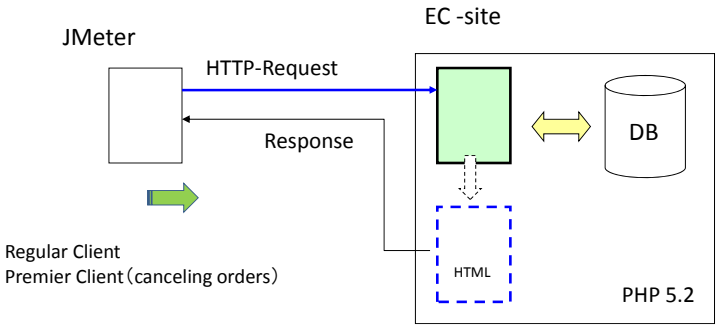
implicit consumption

クライアント・セッションごとの実行履歴を監視



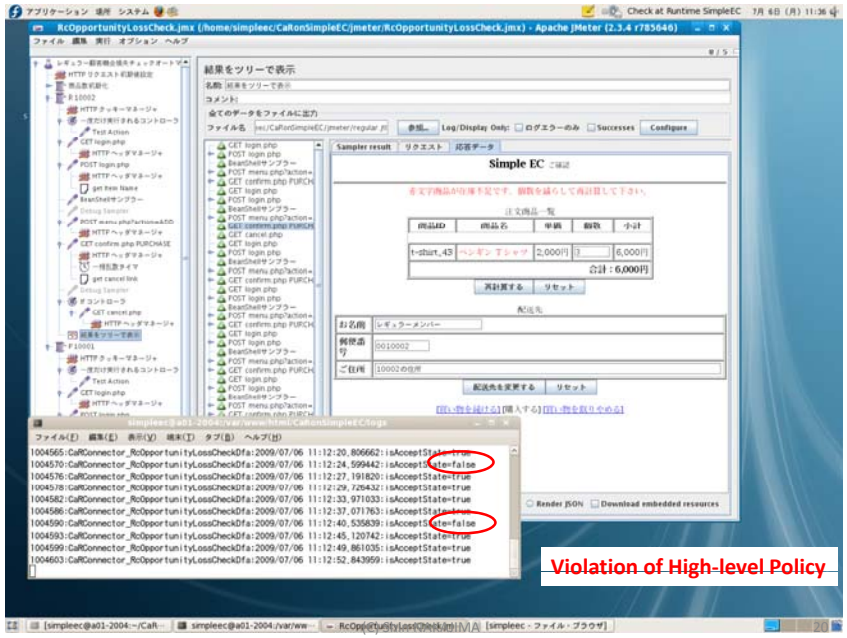
監視オートマトン（記述例）

コンセプトデモ

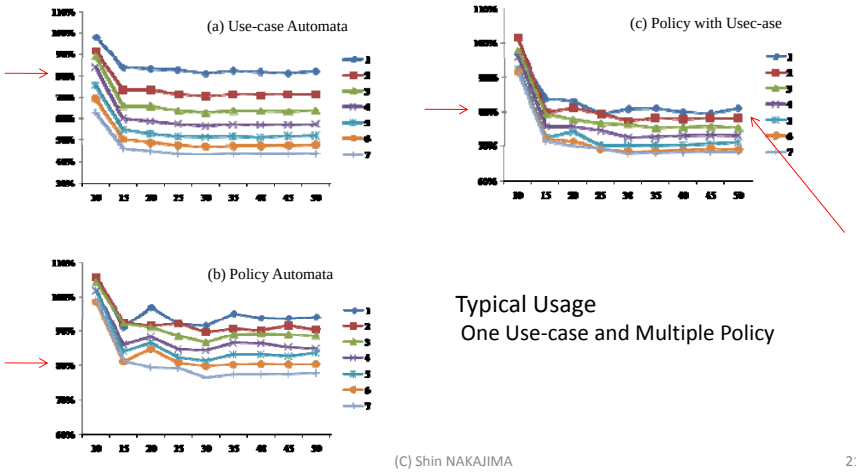


Regular Client
Premier Client（canceling orders）

Detecting Anomalies
Performance（runtime overhead of automata）



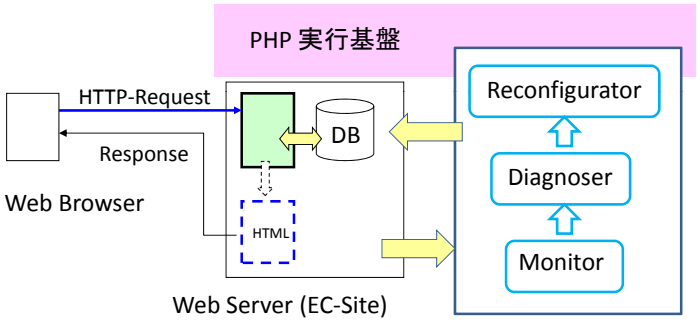
実行性能の傾向



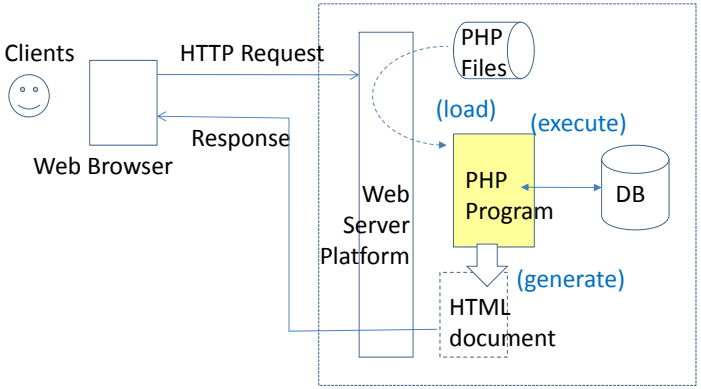
Typical Usage
One Use-case and Multiple Policy

自己適応の導入

監視オートマトンが不具合検出 ⇒ コンポーネント入れ替え

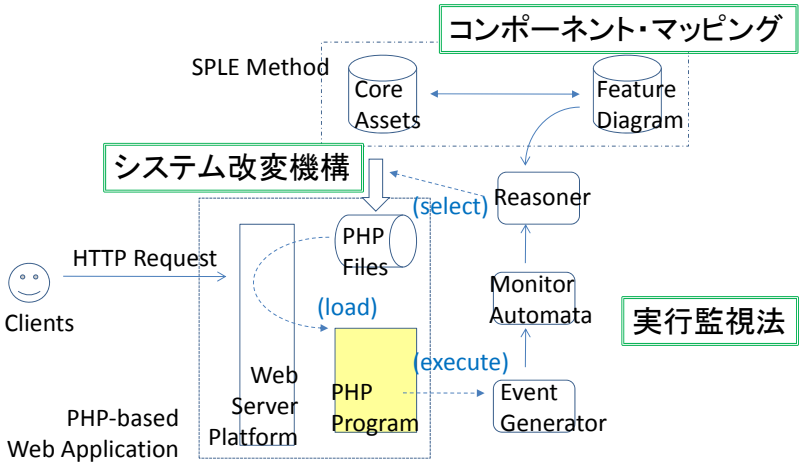


PHPベース・アプリケーション

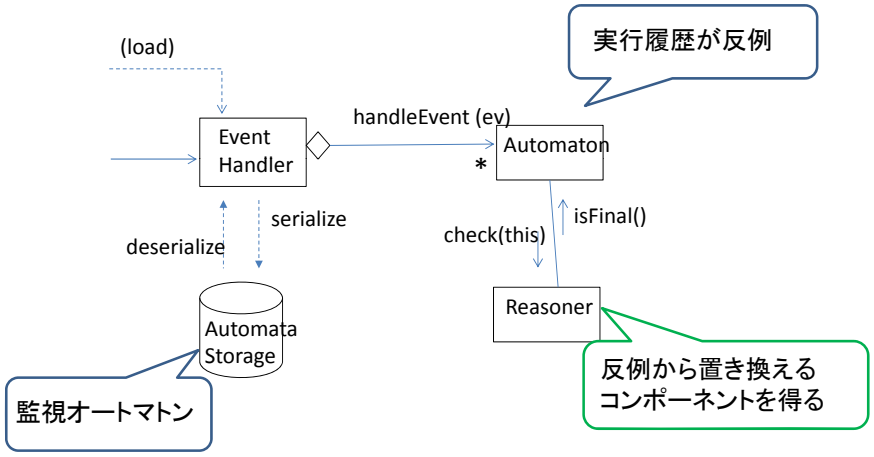


PHP-based Web Application

自己適応アーキテクチャ

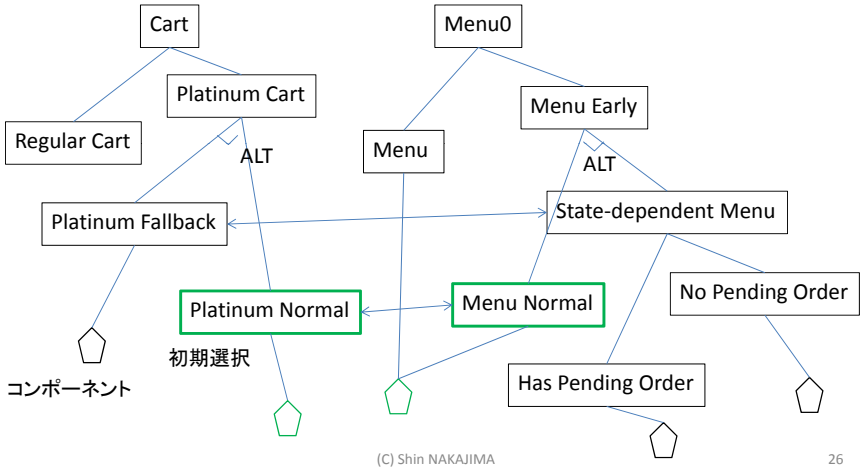


不具合検知の制御フロー



コンポーネント・マッピング

フィーチャ・ダイアグラムの表記法を利用



内容

- 研究の流れ
- Webアプリケーションの事例
- 関連分野と今後の動向
  - 動的検証との関係
  - ITリスクとの関係

監視する性質

- 開発段階 → 兆候検査
  - 2つのステップ
    - 検査性質の候補を求めること （← こちらは今後の課題）
    - 実行時に検査すべき性質であると確認すること
  - 開発時の設計情報に対するモデル検査
  - 非決定性・過大近似 → 見かけの不具合の可能性
- 実行時 → 監視オートマトン
  - 実行時検査で本当に発生するかを確認

実行時検証と実行監視

- Runtime Verification
  - 動的検証（プログラム・テスティング）
  - 振る舞い仕様を満たすか否かの検査： σ |= ψ

（有限長）

- 振る舞い仕様の表現法
  - （線形）時相論理（LTL） → 有界LTL
  - イベントパターン
  - 有限状態オートマトン

cf. 無限長の受理列 L(Buchi) ⊃ L(LTL)

ITリスク

ITリスク ＝ ［事象の発生確率］ x ［影響度の大きさ］

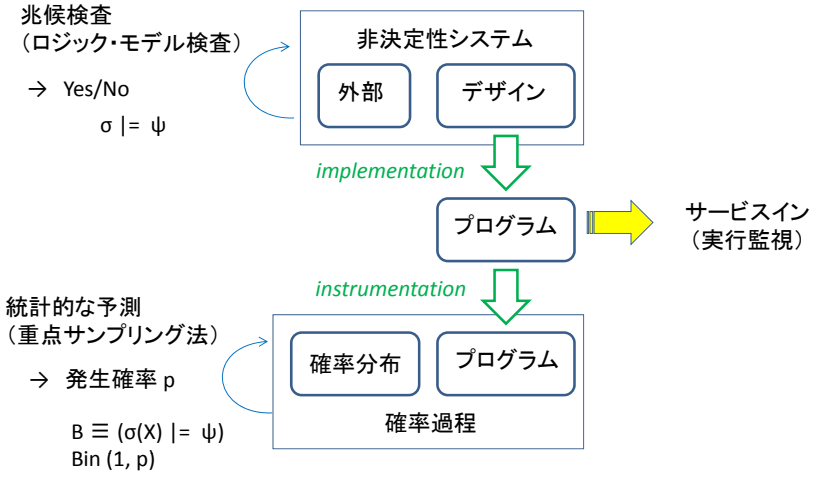
秩序ある複雑さ（Organized Complexity）

影響度 ＼ 発生確率	小さい	大きい
小さい	（放置？）	（自己適応？）
大きい	（自己適応？）	（不具合の修正）

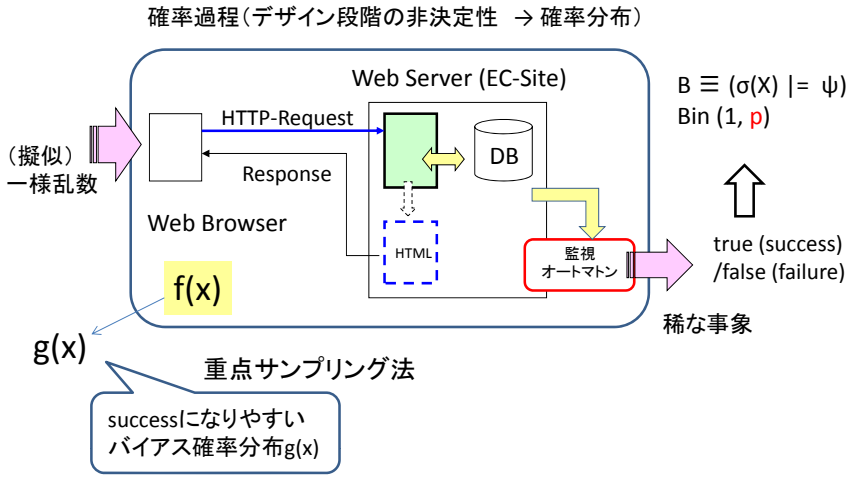
→ 判断の材料

中島震：柔らかな不具合の発生頻度予測，
日本ソフトウェア科学会大会，2012年8月

発生確率の推定

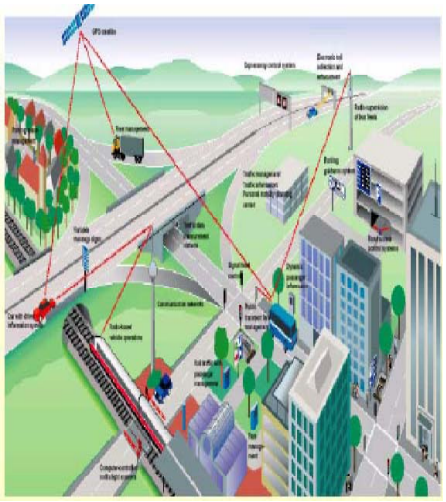


モンテカルロ法



EU-FP7 のビジョン：SoS

スマートハウス

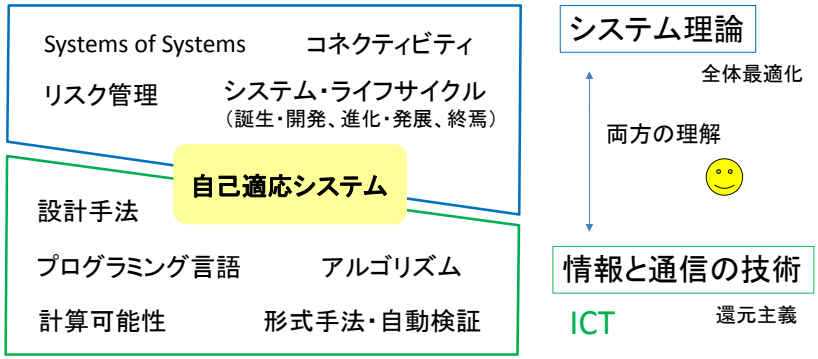


交通制御

ICT Proposers' Day 2011
19 - 20 May, Budapest

システム理論と「情報と通信の技術」

システム思考の重要性



まとめ

- 自己適応システム
  - 初期の重要な研究（要求監視、モデルベース適応）
  - 事例：Webアプリケーション
  - 関連する要素技術：実行時検証
- システム思考の重要性
  - 目指すビジョンの複雑さ
  - コネクティビティとリスク管理

http://researchmap.jp/nkjm/