自己適応ソフトウェア設計の研究動向

国立情報学研究所
鄭　顕志　　石川　冬樹

概要

- 自己適応ソフトウェアに向けては従来より，様々なコミュニティーにて取り組み
  - ソフトウェア工学
  - プログラミング
  - 人工知能
- ソフトウェア工学分野では近年，体系化の試みがなされ，トレンドとして盛んな研究
- サーベイを実施
  - SEAMS，RE@runtime，MODELS@runtimeを中心に

自己適応ソフトウェアのための自己適応設計に関する研究動向，コンピュータソフトウェア Vol. 31 No.1 2014

自己適応（Self-Adaptive）

DARPA Broad Agency Announcement:

- Evaluates its own behavior and changes behavior when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible

2nd SefSAS (Dagstuhl Seminar on Software Engineering for Self-Adaptive System) Report:

- Able to modify their behavior and/or structure in response to their perception of the environment and the system itself, and their goals

自己適応（Self-Adaptive）

- Zave/Jacksonのモデルに基づく説明 [Baresi, 2010]

$$S, D \models R$$

  - ソフトウェアの仕様 S は，想定する実行環境（ドメインの性質） D の下で動作したとき，要求 R を満たすように決定されなければならない

適応（Adaptation）：D が D’ へと変化したときに，R を充足するよう新たな S‘ を定義し，それに応じソフトウェアを修正する

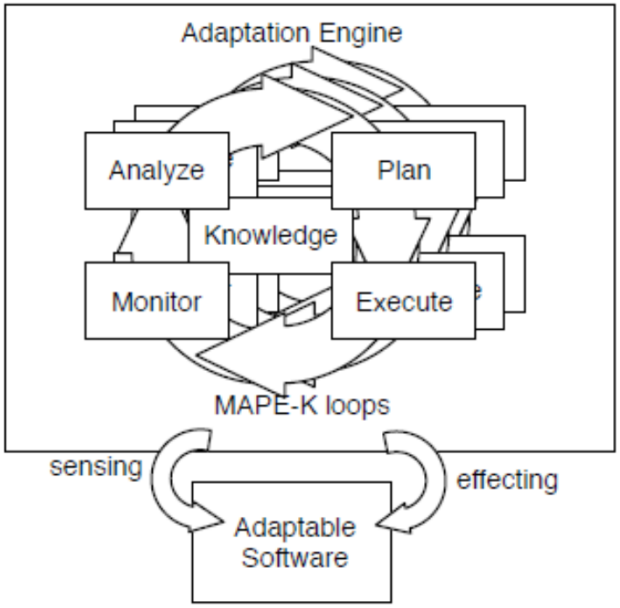
- 人手： 適応保守（Adaptive Maintenance）
- 自動： 自己適応（Self-Adaptation）

※ R の変化への対応は進化（Evolution）とされる

自己適応ソフトウェアの構成

- 外的アプローチが適切：アプリケーションロジックと適応ロジックを分離
- 外的アプローチにおける適応ロジックの代表的なモデル：*MAPE-K*
  - Monitor　　$S, D' \models^{?} R$
  - Analyze　　$S, D' \not\models R$
  - Plan
  - Execute　　$S', D' \models^{\checkmark} R$

（これらのためにKnowledgeが必要）



モデリング・知識の内容

- R： 帰納的要求・非機能的要求
- D： 様々な環境情報
  - ネットワーク構成，利用可能なサービス群，ゴール成立に関するドメイン仮定など
- S： 構造・振る舞い
  - ソフトウェアコンポーネントの構成，サービス呼び出しフロー，ゴールを実現する機能など
- 適応動作
  - コンポーネント配置の変更，サービスバインディングの変更，代替え機能への変更など

補足：@runtimeアプローチ

- S/D/R（のモデル）を実行時にソフトウェア自身が扱っていく必要がある
  - 従来は開発時に，人が扱っていた

  ※ 事前にすべてを想定することが難しいことを前提としようという意義もある [Sawyer, 2010]
- XXX@runtimeアプローチとして扱われている
  - MODELS@runtime： 実行時のモデル駆動によるS'からソフトウェアの生成など
  - RE@runtime： 実行時の R に関する推論
  - Verification@runtime： 実行時の S'（変更された振る舞い）の検証

Monitorにおける課題・研究

- 課題： 実行中のシステムに負荷をかけないための観測制御
- 取り組み例
  - 特定側面に特化したフレームワーク化（作り込み）
  - ゴールモデルと実行時イベントとの紐付け
  - 観測データの精度と，観測のためのオーバーヘッドとのトレードオフ調節
  - 変化があった部分のみの差分更新

Analyzeにおける課題・研究

- 課題： 要求充足の機械的な判定
- 取り組み例
  - ゴールツリーと各ゴールの状態における遷移のモデル化，実行時の充足判定
  - 不確しさを考慮した仕様記述言語（例：as XXX as possible）
  - ファジーゴールの表現や，ファジーゴールの充足割合に対する要求の表現

Planにおける課題・研究

- 課題： 様々な抽象度での選択肢のモデル化と，要求を満たす選択肢の決定
- 取り組み例
  - 開発時における，ゴールモデルやダイナミックプロダクトラインの分析を通した，状況に応じたゴール達成方法の決定
  - プランニング・制約充足による実行時の仕様探索

Executeにおける課題・研究

- 課題： 適切なタイミングでの実行中ソフトウェアの変更
- 取り組み例
  - ソフトウェア変更のためのAPI（アーキテクチャーレベルでの「リフレクション」など）
  - 安全に変更可能な状態を識別し，新たな仕様に対応する状態モデルに移行
  - 変更反映途中でも要求を満たし続けるよう，リフレクションAPIの実行順序を決定

議論

- 今後考えるべきこと？
  - 適応ロジック間の挙癒合
  - 適応ロジックの汎用モデリング
  - 設計から分析へ焦点を