

# リアルタイムシステムと上流設計ツール

日本ラショナルソフトウェア株式会社  
ソフトウェア・エンジニアリング・スペシャリスト  
荒井玲子



# 自己紹介

会社： 日本ラショナルソフトウェア株式会社  
所属： プロフェッショナルサービスグループ  
ソフトウェア・エンジニアリング・スペシャリスト  
荒井玲子 rarai@rational.com

## 経歴：

1991年よりオブジェクト指向開発に携わる  
オブジェクト指向の研究開発、教育、コンサルティング  
リアルタイム系システムへのコンサルティング...  
複写機系システム

組込みオブジェクト指向ワーキンググループ活動参加中

# 設計と設計ツール おおいなる勘違い

- 設計ツールを使用すると...(期待派)
  - UMLが覚えられる
  - オブジェクト指向設計ができるようになる
  - ツールが設計してくれる
  - ソースコードは書かなくていい
  - 生産性があがる
- 設計ツールは...(懐疑派)
  - お絵描きレベルでしょう
  - 使ってもリアルタイムシステムの制約には対応できない
  - 工数がかかるだけ。自分でコードを書いた方が工数がかからない

# ツールから受けられる利点

- 期待する利点を得るには、ツール以外の要素も必要

<<ドメイン知識>>

ドメイン特有の性質

制約の認識

<<オブジェクト指向設計・モデリングノウハウ>>

柔軟なアーキテクチャ設計

柔軟なメカニズム設計

<<設計ツール>>

動的振舞の検証

手作業工数減少

UML表記による記述

<<プロセス>>

インクリメンタルな開発

# 上流設計に求められるもの

- 柔軟なアーキテクチャ設計と検証
  - 制約を吸収する仕組み
  - 独立性の高いコンポーネントへの分割
- キーとなるメカニズムの設計と検証
  - 特に異常系の振る舞い、割り込みからの復帰メカニズム
- 要求が確定していないときの設計アプローチは？
  - ドメインに存在する“モノ”主体のアプローチも現実的
- 支援ツールに対する要件は？
  - 設計モデルの表現力があること
  - モデルのシミュレートができること
  - モデルの論理チェックができること

# ソースコード自動生成について

- ツールがソースコードを自動生成できることで得られる利点
  - より多くの工数を、アーキテクチャ設計およびモデリングに使うことができる
    - 開発費用の削減
    - 上流での振舞確認による品質向上
  - 仕様変更への対応がモデルベースでできる
    - システム拡張性の確保



- 前提条件:
  - 生成されたソースコードが100%信用できること
  - アーキテクチャ中心のシステム設計であること
  - モデル要素には、高いコネクティビティ性があること
    - 例：パフォーマンスの調整がコンポーネントが提供する I/Fのパラメータ設定できる
    - 例：サービスの追加がコンポーネントの入れ替えまたは追加レベル (plug-in) で対応可能である

# 現在の課題

- コードレベルのチューニングが必要
  - 特に組み込み制約を扱う部分
  - パフォーマンス、OS依存制約
    - 副作用：コード自動生成ができるためのモデルを作成してしまう可能性がある
- モデル要素にコンポーネント性がない
  - 仕様変更に対応できるかどうかは、モデル（またはソースコード）の細部がどうなっているかを調べなくてはならない
    - ただし、アーキテクチャによって、ある程度は回避できる課題
  - 実装言語のコンポーネント性が不十分
    - 使用するクラスの変更影響が大きい
- ソースコード自動生成よりも、アーキテクチャや設計モデルの振る舞い検証
  - 振舞い検証には、シミュレーションが有効
    - メカニズムの設計品質確保
  - 自動生成されるソースコードは、設計の副産物と考えた方が無難

# もっとも実りのない議論とは？

- ツール比較
  - ツールにはそれぞれ得意とする分野、支援するフェーズがあり、ツールごとの機能比較をしても意味がない。
  - 開発プロジェクトがどのフェーズで、何を目的にしてツールを使いたいのか、ということを確認していないと、ツールは選択できない。
- 似ている過去の比較議論
  - オブジェクト指向言語比較
    - どの言語が一番オブジェクト指向言語なのですか？
  - オブジェクト指向方法論比較
    - どの方法論が一番よいのですか？
    - どの方法論がリアルタイムシステムに適しているのですか？
      - 参考：リアルタイム向けのオブジェクト指向方法論にはROOM(*Real-Time Object-Oriented Modeling*)があります



# まとめ

- 上流工程に設計ツールは有効
  - ただし、設計ツールの使用から得られる利点を受けるには、ツールの他に、「設計ノウハウ」、「メイン知識」という2つの要素が必要。ツールだけの導入では、得られる利点に限界がある
- 上流設計のキーポイントは、キーとなるメカニズムとアーキテクチャの検証
  - アーキテクチャ中心のシステムで、拡張性、変更容易性を確保
    - 上流設計で確保
    - 制約による最適化は2段階目の設計 (James Rumbaugh)
- 設計ツールの選定は、開発のどのフェーズで何を目的に使いたいのかを明確にしてから
  - 「銀の弾丸はない」というのは、支援ツールでも同じ

# 補足資料：手法、方法論、ツールの関係

- ひとつの手法にたいして、方法論は複数あり、各方法論は、方法論提唱者の解釈に依存している
- 方法論には、支援するツールがあり、ツールは実装の都合に制約されている

