

マルチコア環境におけるタスク設計検証

でたらめなアーキテクチャ
今のアーキテクチャ設計はなってない

2007/11/15
藤倉俊幸

はじめに

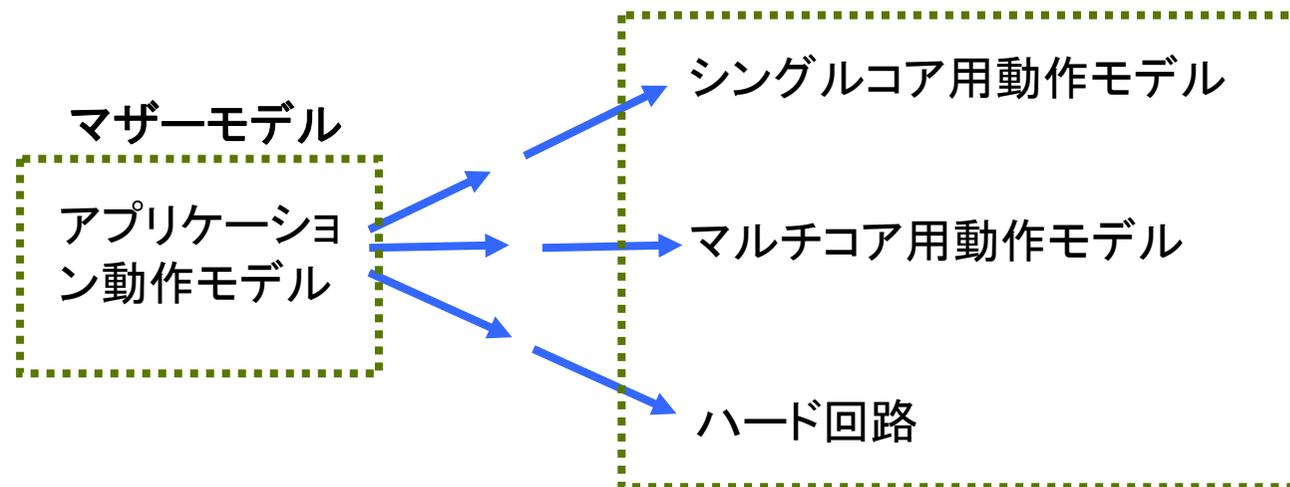
主題: マルチコア環境におけるタスク設計検証

今は旬なプラットフォーム
に関すること

アーキテクチャに関すること

アーキテクチャの視点からプラットフォームを見る。これが重要

実装モデル



たとえば、インタフェース誌、2008,1月号、数理的アプローチ 第9回

検証できるように分解する

■ たいていの場合、相互依存の問題がある

- ❖ ソースコードがなければコンパイルできない
- ❖ コンパイルできないとファイル分割の正しさは確認できない
- ❖ ファイル分割を決めないとソースコードを書けない

■ では、どうするか

- ❖ 反復する
 - ▶ 繰り返し開発
- ❖ より大きな流れの中で確認する
 - ▶ 要求から機能を見る
 - ▶ 全体から部分を見る
- ❖ 予測する
 - ▶ ルールがある
 - ▶ パターンがある

開発プロセス的

アーキテクチャ的

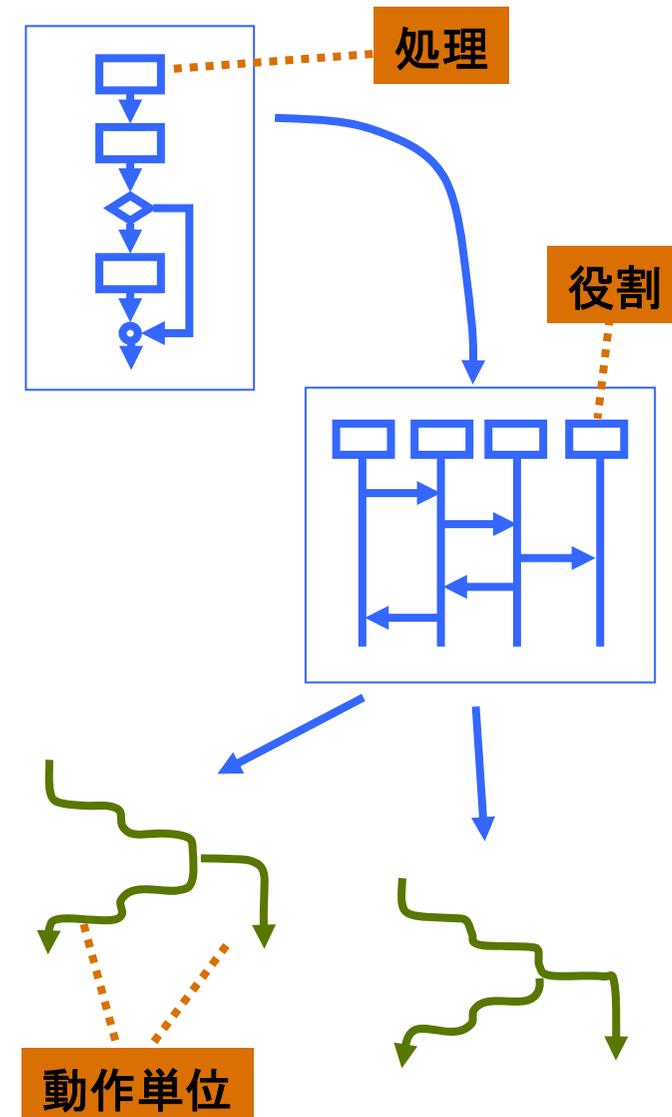
静的な構造を先に決める。どうすればコンパイルできるか知っている

形式的アプローチ

経験的アプローチ

アーキテクチャ的形式アプローチ

- フローチャートの時代
 - ❖ 分割されていない
 - ❖ アーキテクチャなしの時代
- シーケンス図の時代
 - ❖ 分割されたが、役割による分割
 - ❖ 静的アーキテクチャの時代
- 状態マシンの時代
 - ❖ ようやく動作を分割することができる
 - ❖ そして動作の検証が可能になる
 - ▶ ただし、未だに役割の単位と動作の単位の区別のつかない人がいる
 - ❖ 動的アーキテクチャの時代
 - ▶ 分割した状態マシンは、再合成することで検証できる。
 - ▶ 仕様を論理式で書いて、その真偽で検証できる
 - ❖ この枠組みの中でマルチコア対応をすれば安心



状態マシンを使用した検証の流れ

■マルチコアはターゲットアーキテクチャの一つ

