

並行オブジェクト設計手法COMETの 組み込みシステム開発への適用

キヤノン株式会社

小林 秀典

kobayashi.hidenori@canon.co.jp

組み込み開発における問題点

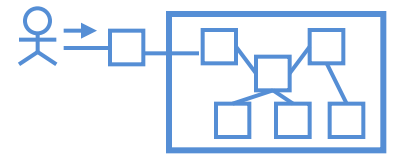
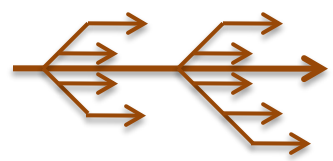
改変に強いソフトウェアを求める組み込みシステム開発と、オブジェクト指向設計技術が目指す方向性は合致している。しかし、組み込みシステム開発における確固たるオブジェクト指向設計技術は確立されてない。RTOSのタスク中心の世界とオブジェクトの世界のギャップは大きく、これらを結びつける手法が必要とされている。

COMETの拡張による解決

タスク設計を系統的に導出する並行オブジェクト設計手法COMET(*1)の組み込みシステム開発における適用性をケーススタディを通して検討する。組み込みシステム開発に対する弱点として挙げられたRTOSへのマッピングに関して、COMETを拡張することでオブジェクト指向設計技術適用の第一歩を得る。

並行オブジェクト設計手法COMETの適用検討

組み込みシステム開発 vs. オブジェクト指向設計



組み込みシステム開発

- 多機種の派生開発
- 短期間での差分開発
- タスク、実装重視の世界

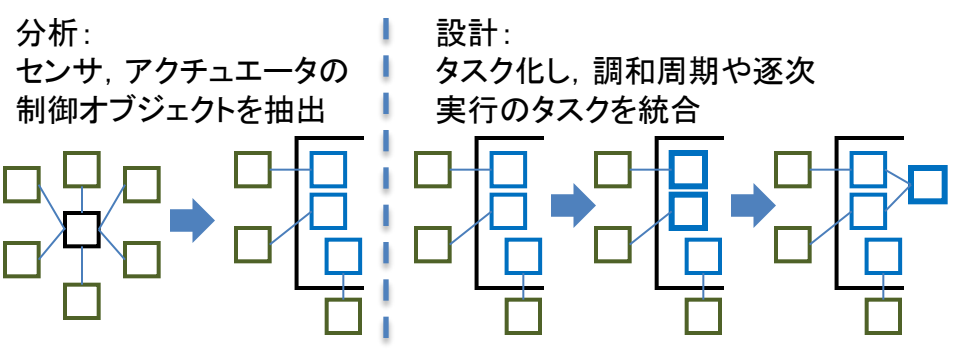
オブジェクト指向設計

- 再利用性に優れた設計
- 拡張に強い設計
- オブジェクト、UMLの世界

- 要求と手法の方向性は合致
- 実利用は限定的(確立されていない)
→ 並行性設計が特徴的なCOMETのケーススタディを行い、ギャップを確認

COMETの特徴

タスク設計を系統的に導出する明確な指針を提示

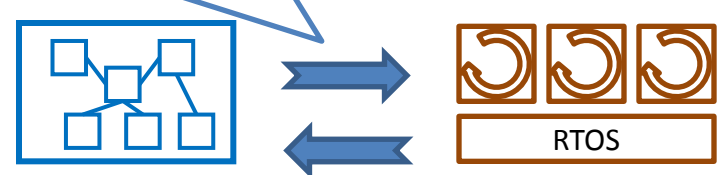


組み込みシステム特有の問題に関してはギャップが残った
・省電力化のためのサスペンド制御 ・RTOSへのマッピング

μITRON実装のためのCOMET拡張

μITRON実装に向けた課題

COMETの設計をRTOSの機能にどうマップするべきか
・同期機構には何を利用する? ・割り込みの扱いは?



RTOSを前提としてCOMETの設計はどうあるべきか
・RTOSの資源数は? ・RTOS上の実装制約の確認は?

COMETの拡張

- (1) 設計の拡張 (RTOS共通)
 - デバイス制御タスクはタスク処理部分と割り込み処理部分を分離
 - タイマオブジェクトは、要求される精度に応じてOS機能を利用するか、独自ドライバで制御するかを決定



- (2) 実装へのマッピング (μITRON前提)
 - コネクタはCOMETの同期・非同期の分類に従ってμITRONの機能を選択
 - オブジェクトに対する排他機構は優先度逆転の有無に着目して、μITRONの機能を選択

(*1) Concurrent Object Modeling and architectural design meTHod
Gomaa, H.: Designing Concurrent, Distributed, and Real-Time Applications with UML, Addison-Wesley, 2000