


InsideSolution

**GSPベースのタスク設計手法 v2.0**  
**GSP研究会**  
**2009.03.06**

eSOL/RCS  
 藤倉 俊幸

http://www.esol.co.jp


InsideSolution

**eSOLについて**  
**イーソルグループの事業構成**

**組込み事業**

<b>組込みソフトウェア製品</b> <small>Embedded Products</small>	組込みソフトウェア開発をトータルにバックアップするOS、統合開発環境、ミドルウェアなどソフトウェア製品を提供
<b>ソフトウェア受託開発</b> <small>Solution Engineering</small>	30数年の技術の蓄積を背景としたOS、アプリケーション、開発場所にとらわれない組込みに特化した受託開発
<b>コンサルティングサービス</b> <small>Research &amp; Consultation</small>	大規模組込みソフトウェアを開発するための開発工程、開発手法、開発ツールなどに資するコンサルティングサービス
<b>技術者教育と派遣サービス</b> <small>eSOL em@ex</small>	組込みソフトウェア技術者の教育サービスと、その教育によって育てた技術者の派遣サービス（株式会社イーソル エンベックスがご提供します）

**特機事業**

<b>流通システムソリューション</b> <small>Logistics Engineering</small>	縦横機能性に特化したドットプリンタやハンディターミナルの開発・販売、開発受託、ソフトウェア開発支援ツールを提供
--	---

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.


InsideSolution

**Research & Consultation Services(RCS) 部**

**リサーチ&コンサルテーション サービス**

「実プロジェクトを止めずに、プロセスを改修。増われたノウハウを捨てることなく、新技術を導入。現実解を追求し、何でも足すが得も引かない。」イーソルのコンサルタントは、これらを実現します。

**RCSが提供する4ステップ**

**サービスメニュー**

**アセスメント**  
Assessment  
 アセスメントを行い、その結果をもとに強みの発掘や強みを活かす提案を行います。

**教育「ノウハウ」**  
Education  
 足りない技術力と知識を補います。  
アーキテクチャ    開発プロセス    設計・検証

**コンサルテーション**  
Consultation  
 「技術はあるのに、うまく回らない」「効果が長えない」……そんなお客様を救えます。

**自律**  
Self-study  
 お客様自身によるアセスメントと開発が可能です。

**研究**  
Research  
 お客様と一緒に新しいアイデアや手法を作り上げます。

**効果のイメージ**

**攻めのコンサルRCS**

1) 知識の出し惜しみはしません 2) 型にはまった考え方はしません 3) 評論家ではなく相談役です

RCSアーキテクチャアセスメントを一度お試しください

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.


InsideSolution

**タスク設計と検証**

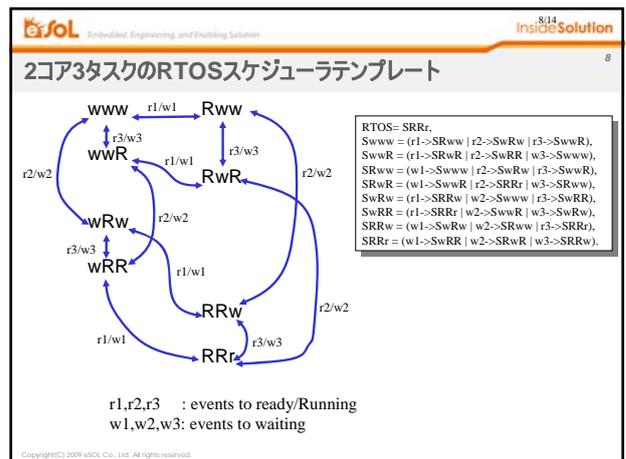
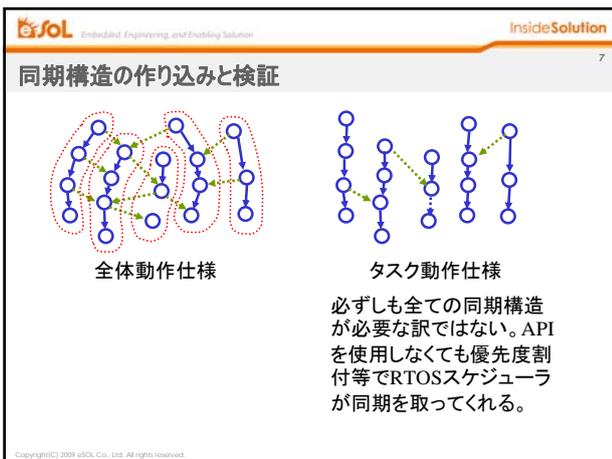
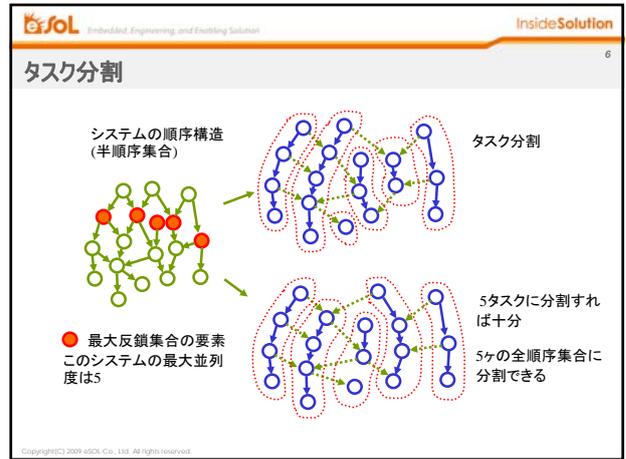
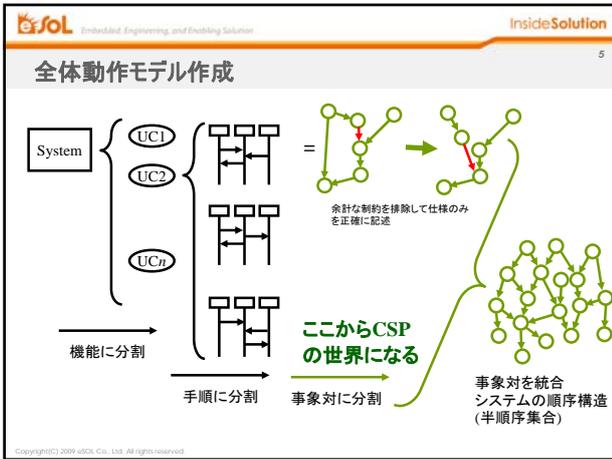
**■ マルチタスクシステム**

- ❖ 組込みシステムでは、並列・並行動作が前提
- ❖ 並列・並行動作する単位をここではタスクと呼ぶ
- ❖ 従来は全てスキルの高いエンジニアが設計していた

**■ CSPを利用した場合**

- ❖ 全体モデルを作る → 動作に関する形式化と検証
- ❖ 分割する → タスク分割の最適化
- ❖ 同期構造を作る → 設計検証
- ❖ テストケースの生成 → 自動テスト、網羅率の把握

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.



InsideSolution

### 不要になる技術と新しく必要になる技術がある

並列度の高い、メニーコア、TPCORE、XMOS環境(CSP環境)でのRTOSを使用しない実現方法

CSPアプリケーションモデル

並列度の低い環境による並行動作での実現

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.

InsideSolution

### 外部仕様の抽出、テストへの応用

システムの順序構造 (半順序集合)

入力順序構造

出力順序構造

● 入力を伴う事象  
● 出力を伴う事象

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.

InsideSolution

### 全体モデルの作り方の例

■ 最も単純な繰り返しのモデリング

Seq1

$A = (a \rightarrow br \rightarrow END).$   
 $B = (ar \rightarrow cr \rightarrow b \rightarrow END).$   
 $C = (c \rightarrow END).$   
 $MA = (a \rightarrow ar \rightarrow END).$   
 $MC = (c \rightarrow cr \rightarrow END).$   
 $MB = (b \rightarrow br \rightarrow END).$   
 $||Seq1 = (A|B|C||MA||MC||MB).$

$LA = A; LA.$   
 $LB = B; LB.$   
 $LC = C; LC.$   
 $LMA = Ma; LMa.$   
 $LMc = Mc; LMc.$   
 $LMb = Mb; LMb.$

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.

InsideSolution

### 全体モデルの作り方

■ 一般の場合

Seq1

$A1 = (a \rightarrow br \rightarrow END).$   
 $B1 = (ar \rightarrow cr \rightarrow b \rightarrow END).$   
 $C1 = (c \rightarrow END).$   
 $Ma = (a \rightarrow ar \rightarrow END).$   
 $Mc = (c \rightarrow cr \rightarrow END).$   
 $Mb = (b \rightarrow br \rightarrow END).$   
 $(Seq1) = (A1 | B1 | C1 | Ma | Mc | Mb).$

全体モデル

$STM = Seq1S.$   
 $Seq1S = (Seq1)T1T2.$   
 $T1T2 = (t1 \rightarrow Seq2S | t2 \rightarrow Seq3S).$   
 $Seq2S = Seq2; T4.$   
 $T4 = (t4 \rightarrow Seq1S).$   
 $Seq3S = Seq3; T3.$   
 $T3 = (t3 \rightarrow Seq2S) \vee (t1, t2, t3, t4).$   
 $||SSTM = (STM).$

- 状態マシンのテンプレートを生成する
- そこに各シーケンス図から作ったモデルを代入する
- 全体モデルでオブジェクトモデルを合成したもので検査する

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.

InsideSolution 13

### タスク分割と検証の考え方

全体モデル BaseModel  
act<sub>1</sub>, act<sub>2</sub>, ..., act<sub>n</sub>

分割する

分割すると勝手に動くけどいいですか?

TaskA = BaseModel @ {actA1, actA2, ...}.  
TaskB = BaseModel @ {actB1, actB2, ...}.  
TaskC = BaseModel @ {actC1, actC2, ...}.

(TaskA || TaskB || TaskC) ⊆ BaseModel によって検証する

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.

InsideSolution 14

### 検証例

Initial  
Seq1

A B C  
Seq1

■ Cが勝手に動いてしまう危険がある

Check

-1 0 1 2 3 4 5 6 7

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.

InsideSolution 15

### まとめ

- CSPをタスク設計に利用する手法について説明した
  - ❖ タスク仕様を形式的に記述して、検証することが出来る
    - ▶ タスク分割の最適化が出来る
    - ▶ 同期構造の設計と検証が出来る
    - ▶ テストケース生成、ログ解析器を生成出来る
  - ❖ RTOSモデルと合成することで環境に合わせた設計・検証が可能になる
- 基本設計をLTSAでおこない、詳細設計を別のツールでおこなうことも可能
  - ❖ CならSPIN、アセンブラならNuSMV
- 状態爆発については、抽象化や問題の切り分けで対応する。そのためには上流工程からの開発プロセスの見直しが必要になる。
  - ❖ ここが、ネックになっている

Copyright(C) 2009 eSOL Co., Ltd. All rights reserved.