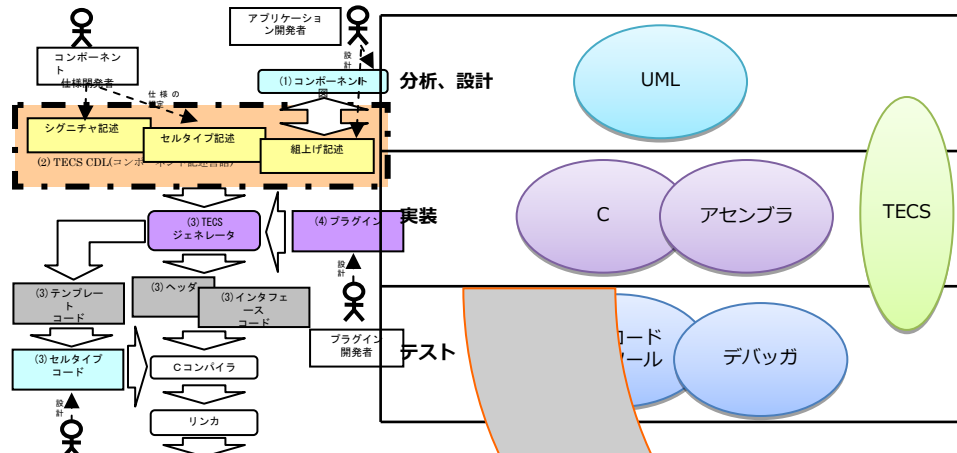


TECS適用事例と そこから見えてきた醍醐味

TOPPERS コンポーネントWG
株式会社ヴィッツ
鵜飼 敬幸

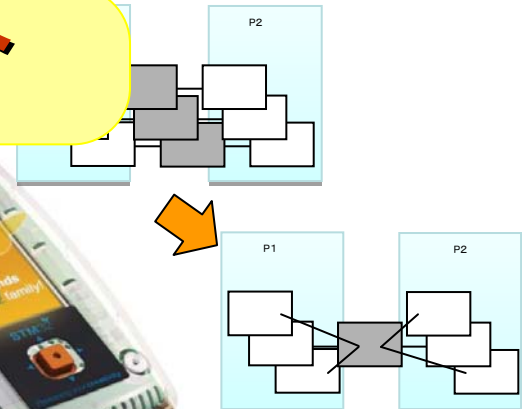
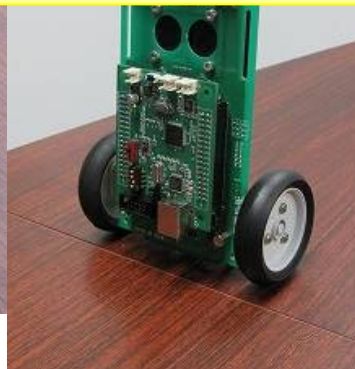
お話すること



TECS概要

TECSの醍醐味

適用事例紹介



TECSとは

TECSとは

- **TECS(TOPPERS Embedded Component System)**

- 各種のソフトウェアモジュールを部品化し、必要な部品を組み合わせることによって大規模な組込みソフトウェアを効率的に構築するための技術
- マルチコアプロセッサシステム/分散システムにおける遠隔呼出し(RPC : Remote Procedure Call)にも対応

- **開発目的**

- 大規模、複雑化する組込みソフトをコンポーネント構造にすることで見通しをよくする
- コンポーネントモデルを規定することで、組込み用ソフト部品の仕様定義を容易にし、流通を促進する
- 組込み向きの分散フレームワークなどフレームワークを実現するためのフレームワークを実現する

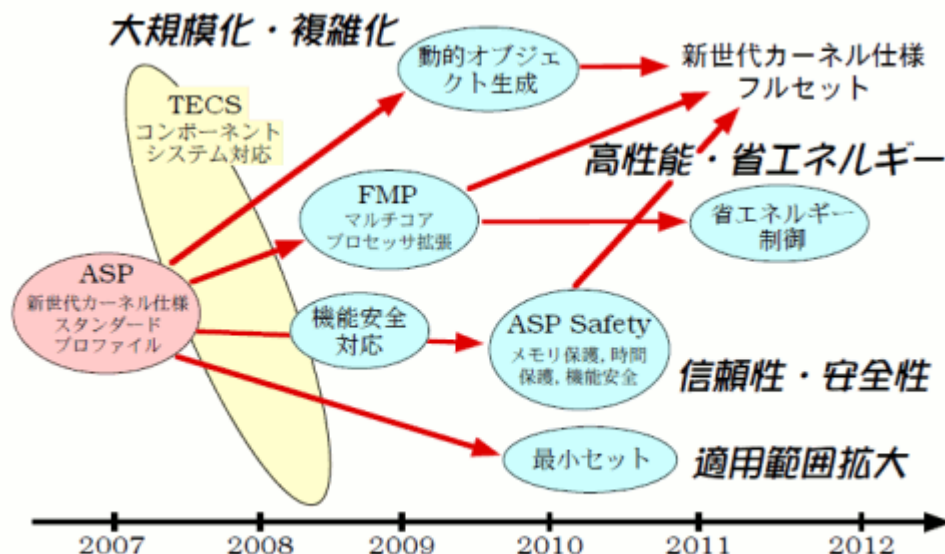
コンポーネント仕様WGのあゆみ

2004年 1月	コンポーネント仕様 WG 開始
2006年 3月	TECSと命名（方向性がほぼ固まる）
2006年10月	セキュリティコンポーネントIPA未踏に採択
2007年12月	組込みZINE連載開始(現CodeZine)
2008年 6月	会員向けリリース開始
2008年11月	キャッツとヴィッツがサポートを表明
2009年 5月	一般向けリリース開始 組込みプレスに記事を掲載
2010年 3月	NEP公開講座でセミナー実施
2010年 7月	同上
2010年 6月	トランスペアレントRPC会員向けリリース
2010年12月	STM32 Primer2 を用いた教育教材一般公開

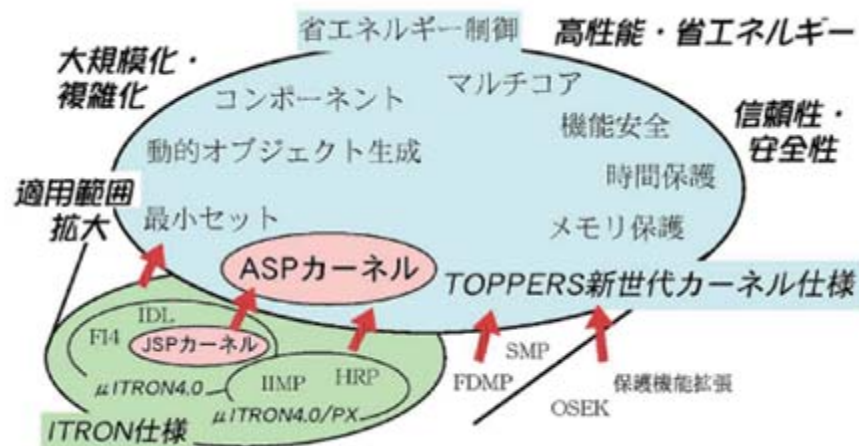
TECSの位置づけ

- TOPPERSプロジェクトでの位置づけ

(1)TECS対応



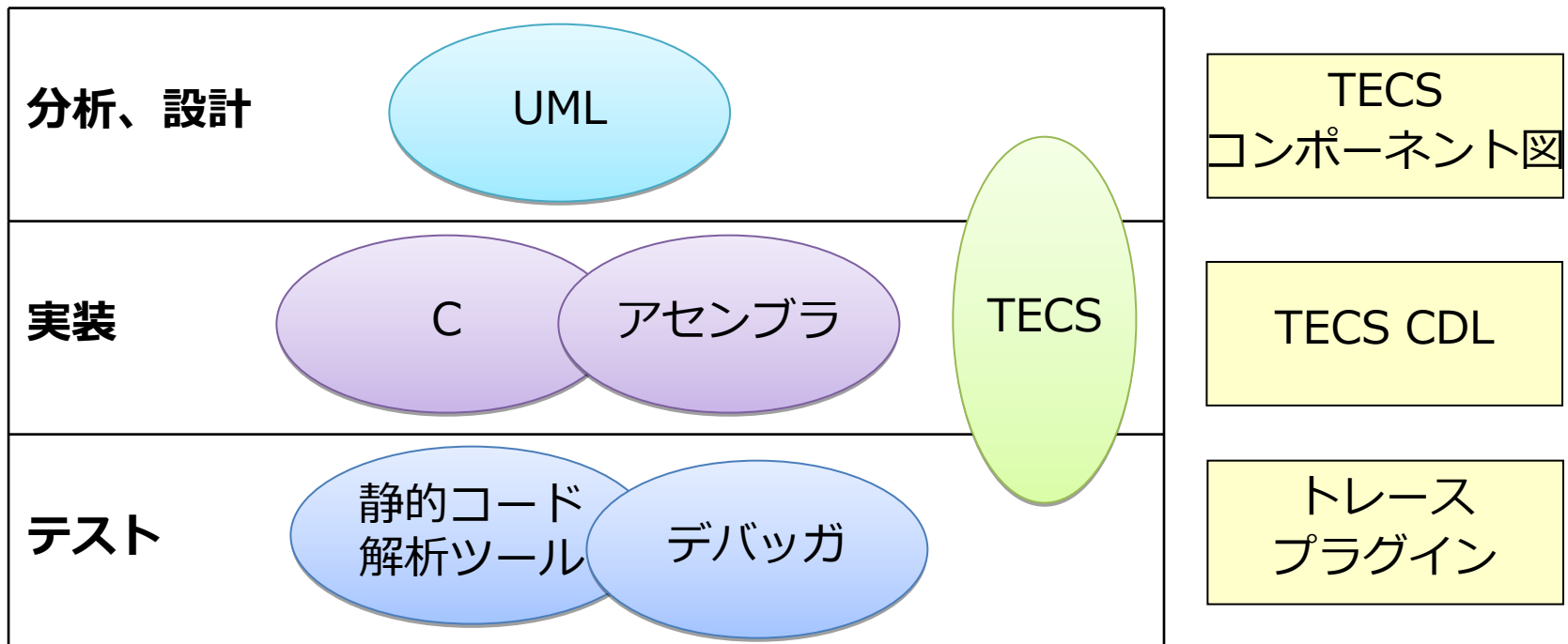
※ ASP以外のカーネルの名前は仮称



(2)新世代カーネル仕様

TECSの位置づけ

- ツールチェーンにおける位置づけ
 - TECSがカバーできる設計領域の範囲



TECSの仕様化方針

- **静的なコンポーネント生成と結合**
 - － 柔軟な組上げと最適化によるオーバーヘッドの最小化
- **すべてのものをコンポーネントとし、隠さない。**
 - － アロケータ、RPCチャネルなどを隠さない
- **インタフェース定義の曖昧さをなくす（型・入出力など）**
- **オブジェクト指向技術を直接的には取り入れない**
 - － C言語を基本とする（想定する利用者の知識）
- **ROM化コードの生成、ソースコードからビルド**
- **動作コンテキストの指定**
- **プラグインによるフレームワーク・フレームワークの実現**
 - － 分散フレームワーク、セキュリティフレームワーク等フレームワークを構築するフレームワークに使える

TECS仕様の構成

- **TECS コンポーネントモデル**
 - TECSによるソフトウェア部品の構成方法と使い方を規定
- **TECS コンポーネント図**
 - TECSコンポーネントモデルを図により表現する手段
- **TECS コンポーネント記述言語(TECS CDL)**
 - 正確なコンポーネントの定義とコンポーネントを組み合わせてシステムを組上げる記述言語
- **TECS コンポーネント実装モデル**
 - TECS コンポーネントを実装し、利用するモデルを定義
 - 仕様書に提供されているモデルは一つの例

参照実装

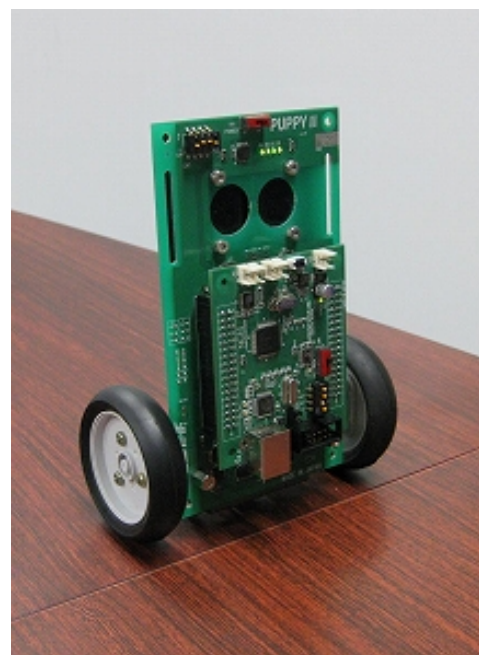
- TECSジェネレータ¹tecsgenの実装
- TECSジェネレータのプラグインの実装
 - トレースプラグイン
 - RPC プラグイン（分散フレームワークの実現）
 - （セキュリティプラグイン）…IPA 未踏ソフトウェア開発事業採択

¹単にジェネレータと書いた場合、インタフェースジェネレータのことを指します

適用事例紹介

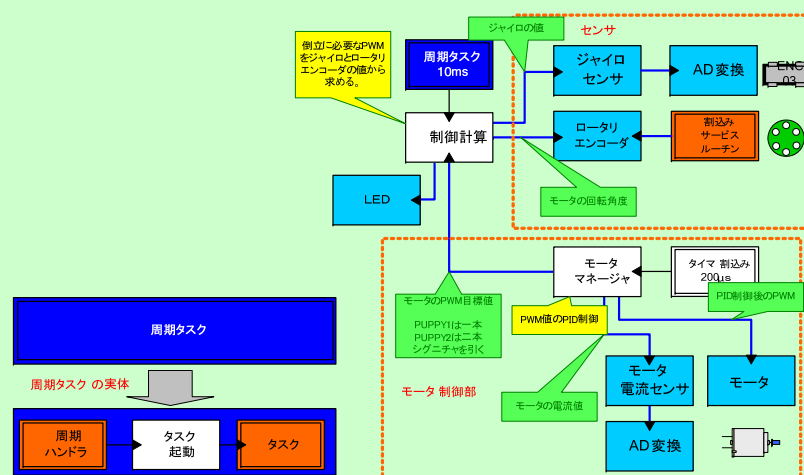
北斗電子 PUPPY / PUPPY2を用いた事例

名古屋大学組込みシステム人材育成プログラム
NEP で利用するPUPPY/PUPPY2を対象

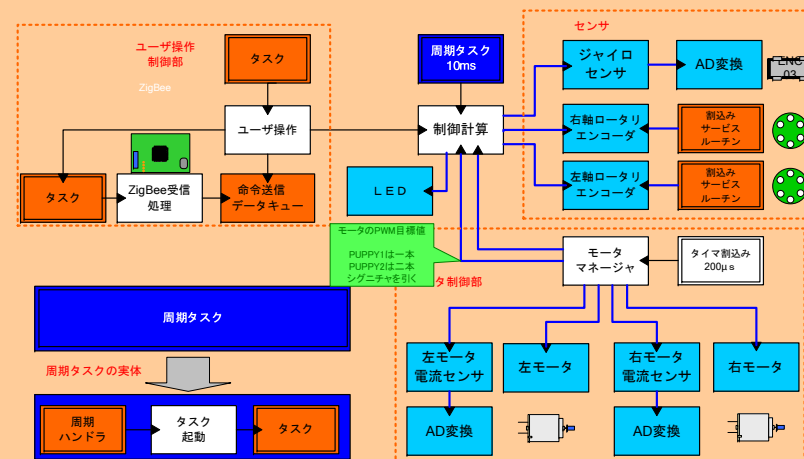


2009, 2010 NEP公開講座実施

PUPPY, PUPPY2におけるTECS適用概要



PUPPY:1軸



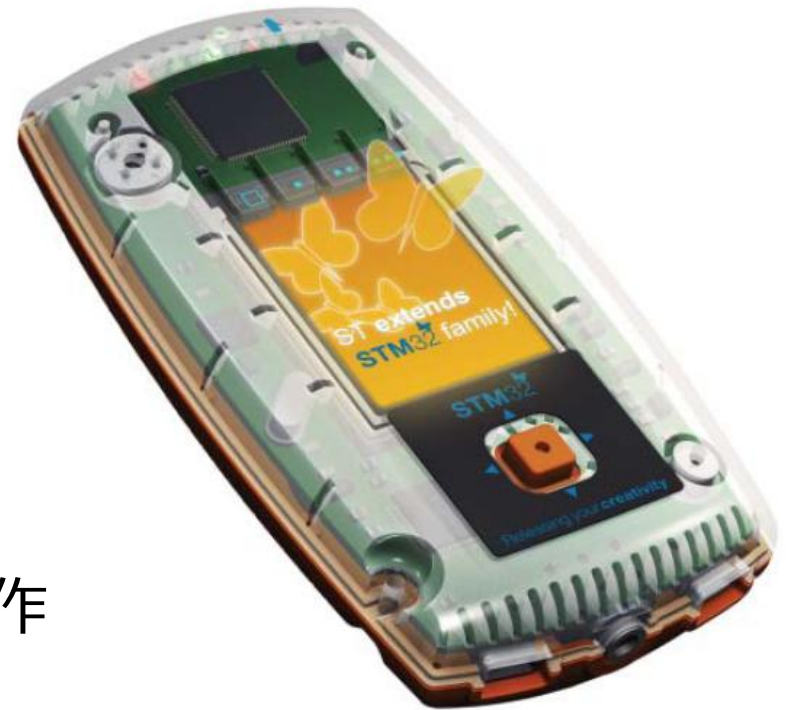
PUPPY2:2軸

- 1軸倒立制御と2軸倒立制御の共通化
 - コンポーネント流通の実践
 - 青：共通コンポーネント
 - 橙：TECS化カーネルコンポーネント
 - TECSのメリットである**プロダクトライン**的構成
 - 既存プログラムからのリバーズ開発例

Raisonance製 STM32 Primer2を用いた事例

ET2010 初日に一般公開

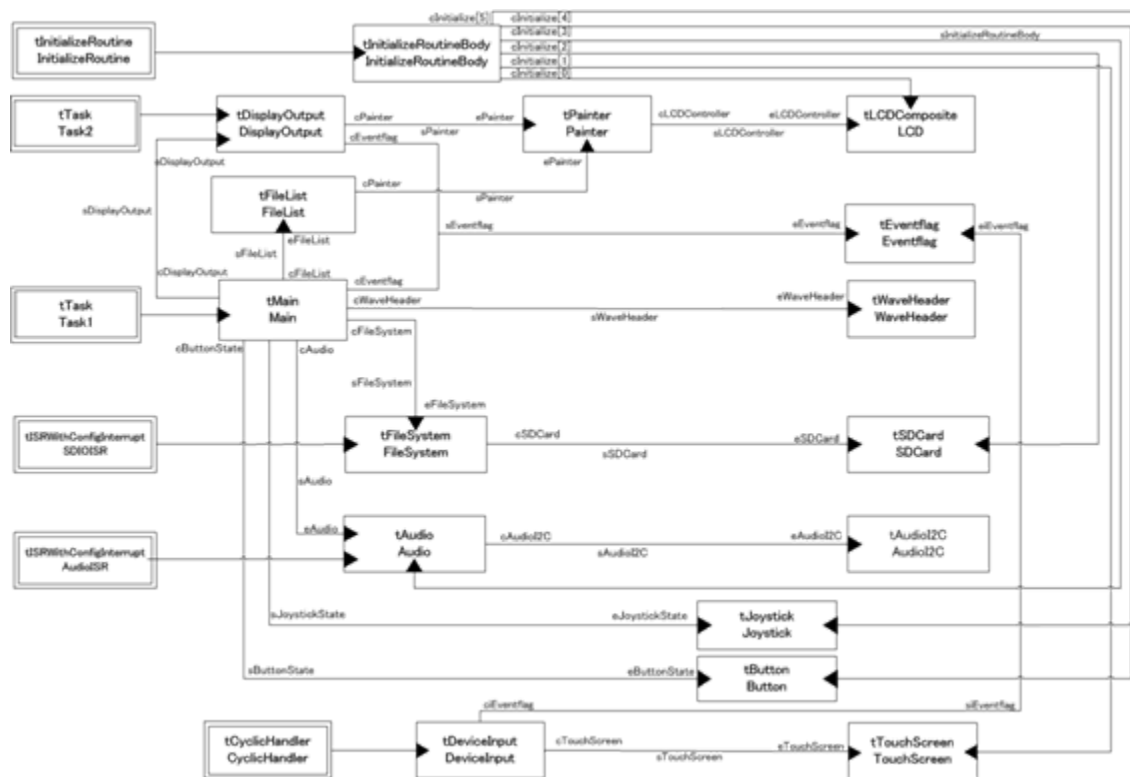
- 2009年度TOPPERS公募型事業 採択案件
- (株)デジタルクラフト担当
- TECS理解の教育教材
- 基礎から応用までカバー
- 最終的にMP3プレイヤー製作



STM32 Primer2におけるTECS適用概要

- MP3プレイヤーのコンポーネント図

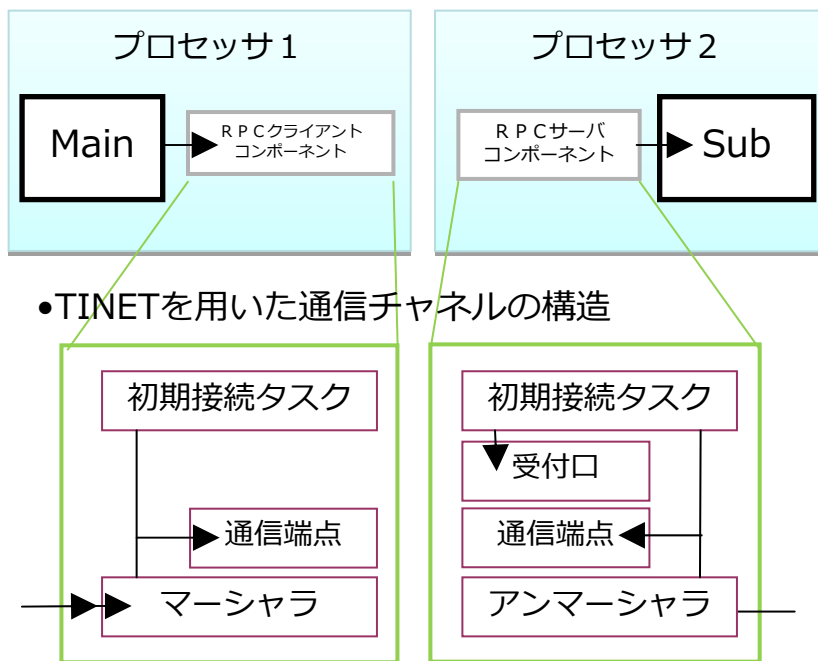
- コンポーネント設計からのトップダウン開発事例
- 各例題と共通の部品で構成：**コンポーネント流通**



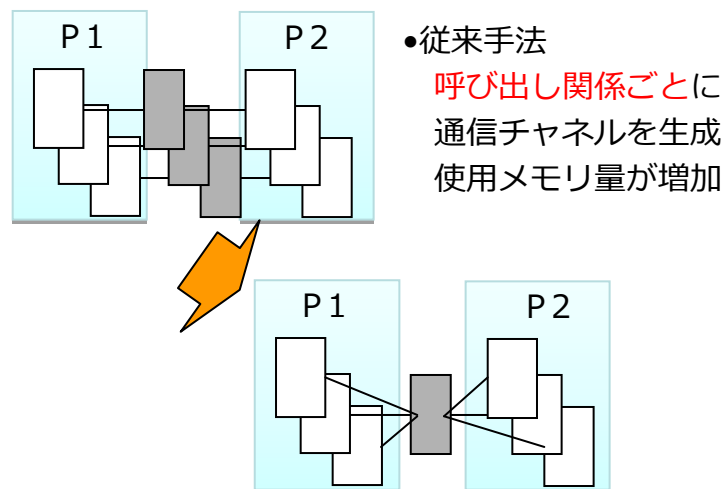
TECSとTINETを用いたRPC実装事例

● 省メモリ型RPC機構の実現

- TINET（組込み向けTCP/IPプロトコルスタック）を用いたRPC機構の提案
- RPC機構の通信チャンネルをプロセッサごとに集約し，メモリ使用量を削減
- **性能改善**のための適用事例



● 通信チャンネルの集約



- 通信チャンネルの集約による省メモリ化
サーバ側でROM2.6%減，RAM26.3%減
クライアント側でROM1.7%減，RAM18.1%減

※シミュレータによる動作確認結果

TECSの醍醐味

TECS利用シーンとして掲げてきたもの

TECSでできること

- ソフトウェア構造の見える化
 - 大規模化、複雑化するソフトウェアの見通しをよくする
 - 開発の分担を容易化
- インタフェースの見える化
 - 人と人のインタフェースミスを低減
 - 再利用の促進
 - インタフェース標準化をしやすくし、部品流通を促進
- 分散フレームワーク(RPC)
 - プロセッサ間の機能呼び出しを容易化
- その他
 - セキュリティ機能
 - 検証の支援などなど

TECS設計方針

- 静的なコンポーネント生成と結合
 - 柔軟性と最適化によるリソース削減
- 全てのコンポーネント化・非隠蔽
 - メモリ管理・RPCも隠蔽しない
- インタフェース定義の曖昧さ排除
 - 入出力、型、配列要素数など
- C言語で扱える
 - オブジェクト指向的な使い方（純ではない）
- ROM/RAM支援
 - 細かな出力先領域の制御
- 動作コンテキストの指定
- フレームワーク・フレームワーク実現

「2つ（構造・I F）の見える化」のもたらすもの

「ソフトウェア構造の見える化」

「インタフェースの見える化」

大規模化、複雑化に対する解決の布石

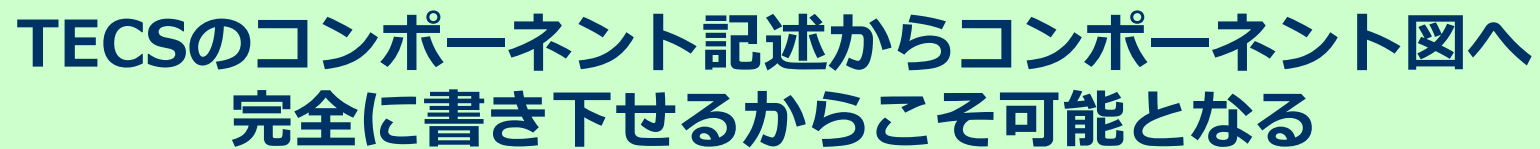
→問題の現場では何が起きているのか？

- 機能追加によるコードとリソースの肥大化
 - － 初期アーキテクチャのまま時間を理由に歪な継ぎ足しを繰り返す
 - － 機能の肥大化がリソースの拡充に直結する
- 製品バリエーションに対応するコードが混在
 - － 条件分岐・マクロスイッチのネストや数が膨大で理解性に乏しい
 - － 過剰な「変数・関数の複写」が増加の一途を辿る



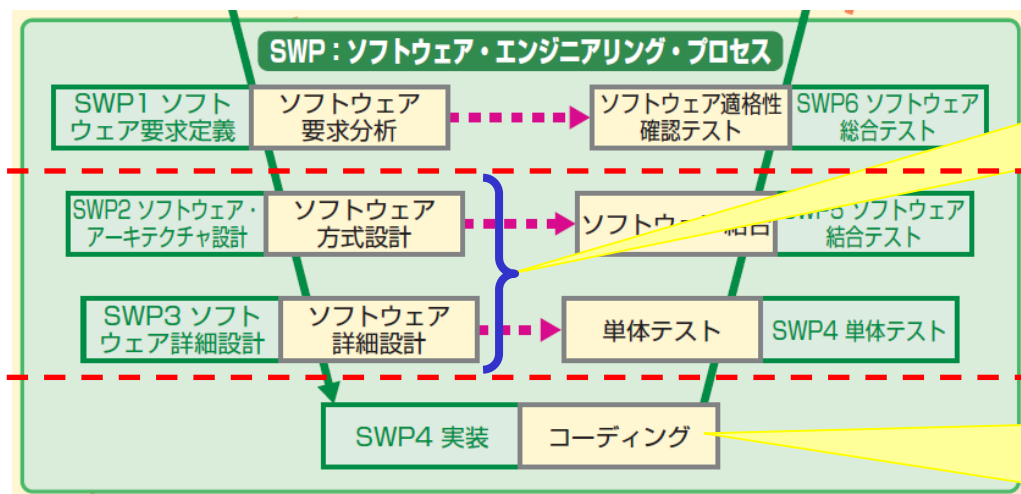
- リソースの拡充が許されない場合はどう対処するか？
- 複雑化した構造を全て把握するベテランの負荷に頼るか？
- ベテランはいつまでいるのか？

TECSは構造・I Fをコンポーネントで取り扱う
 →バリエーションをコンポーネントの組替えで達成
 →ソースコード上に残しておくのではない



「2つ（構造・I F）の見える化」のもたらすもの

開発工程で俯瞰してみると・・・



コンポーネント結合
コンポーネント記述
統一された結合規則

マクロスイッチ
条件分岐
コメントによる設計情報
(曖昧さを含む)

出典：SEC 組込みソフトウェア向け開発プロセスガイド

- コンポーネント結合はソフトウェア設計に影響
- 開発工程をコーディングより数段階抽象化している

ソフトウェアアーキテクトの意識との乖離を抑止する

フレームワーク・フレームワークとは

そもそもフレームワークって一体何だろう？



世の中のソフトウェアには大なり小なり実装されている

OMGに見るフレームワーク

- OMG (Object Management Group)
 - 数々のモデル化言語が仕様化され策定されている
 - OMGにおいてもFrameworkの単語は散見できる



Catalog Of UML Profile Specifications

This page provides a catalog of UML Profile specifications. A **UML profile** is a specification that does one or more of the following:

- Identifies a subset of the UML metamodel.
- Specifies "well-formedness rules" beyond those specified by the identified subset of the UML metamodel.
"Well-formedness rule" is a term used in the normative UML metamodel specification to describe a set of constraints written in UML's Object Constraint Language (OCL) that contributes to the definition of a metamodel element.
- Specifies "standard elements" beyond those specified by the identified subset of the UML metamodel.
"Standard element" is a term used in the UML metamodel specification to describe a standard instance of a UML stereotype, tagged value or constraint.
- Specifies semantics, expressed in natural language, beyond those specified by the identified subset of the UML metamodel.
- Specifies common model elements, expressed in terms of the profile.

Specifications are listed alphabetically.

Description Of Catalog Entries

Click [here](#) for a description of each field for specifications catalogued below.

Understanding Terms Used

In order to understand the various terms used to identify an OMG specification as it moves through its editing cycles, consult the [OMG Specifications Tutorial](#).

出典 : OMG Specification catalogues page.

代表的なフレームワーク仕様

- SysML
- SorML
- CORBA
- CORBA CCM
- EDI/EDOC
- MARTE
- SoC
- AUTOSAR

※SysML以外の仕様は明確に
UML Profile を仕様の一部として
位置づけている

OMGに見るフレームワーク

OMGではメタモデル階層を厳密に定義している
先に登場したフレームワークの数々は、UMLの
プロファイル(Profile)として定義されている

モデルレベル	説明
M3	メタメタモデル：MOFコア
M2	メタモデル：MOF要素の実態 (UMLで定義されるクラス自体など)
M1	モデル：メタモデル要素の実態 (作成したクラスなど：車クラス…)
M0	実態：自分で定義したクラスの実態 (車1, 車2, …)

Profileとは

メタモデルに「制約」を与え
UML自体を拡張、あるいは
機能追加する手段である

SysML, MARTEなども
Profileの一つ

- [参考]：AUTOSAR の メタモデル階層は、OMGで定義しているモデルを参照するが、0から4の5階層からなる点異なる（抽象度はM4が最も高い）

フレームワーク・フレームワークということ

フレームワークのフレームワークとはつまり…

メタモデルの制約を実装する機能を持つこと

フレームワークを独自に設計することも！

モデルレベル	説明
M3	メタメタモデル：MOFコア
M2	メタモデル：MOF要素の実態 (UMLで定義されるクラス自体など)
M1	モデル：メタモデル要素の実態 (作成したクラスなど：車クラス…)
M0	実態：自分で定義したクラスの実態 (車1, 車2, …)

TECSによる実装

SysML, MARTE,
CORBA, Schedule, etc..
※策定中 EAST-ADL, Timmo,
AUTOSAR

フレームワークの試作という新しい側面が！

アーキテクトの魔法の杖として

秘めたアーキテクチャの具現化ツールに

- 日本は技術の囲い込みによる成長が得意
 - クローズド・インテグラルな開発寄り
 - 擦り合わせによる高品質・高性能が売り
 - 規格創造よりも既存規格のブラッシュアップに一日の長
 - 規格浸透後のマイニングが早い
- 昨今のグローバル戦略においては競争領域がシフト
 - ラピッド開発を得意とする諸国が急進
 - 規格化で先を行く欧州圏は協業領域と競争領域を明確化
 - 技術の囲い込み範囲がより集約されてしまう



フレームワーク・メタ領域での技術向上を！

出典：東大ものづくり経営研究センター長：藤本隆宏「ものづくりと産業競争力」

進化を続けるTECS

RTミドルウェア連携 TECS-Unit バイナリ流通支援

XML接続ハードウェア統合 形式手法



.NETブリッジ

動的結合

最適結合探査

Profile策定

検証

軽量化Rubyブリッジ

TECSイノベーションの矢は放たれた

さいごに

FlexRay
Timmo CAN LIN
Jaspar OSEK/VDX
AUTOSAR uITRON
MARTE HIS 「発信する」 SPE-ED
uC-Kernel SS-MIX IEEE11073
XMPP
Modbus Android DICOM
コンティニューア規格
SysML M2M Linux BPMN EAST-ADL