



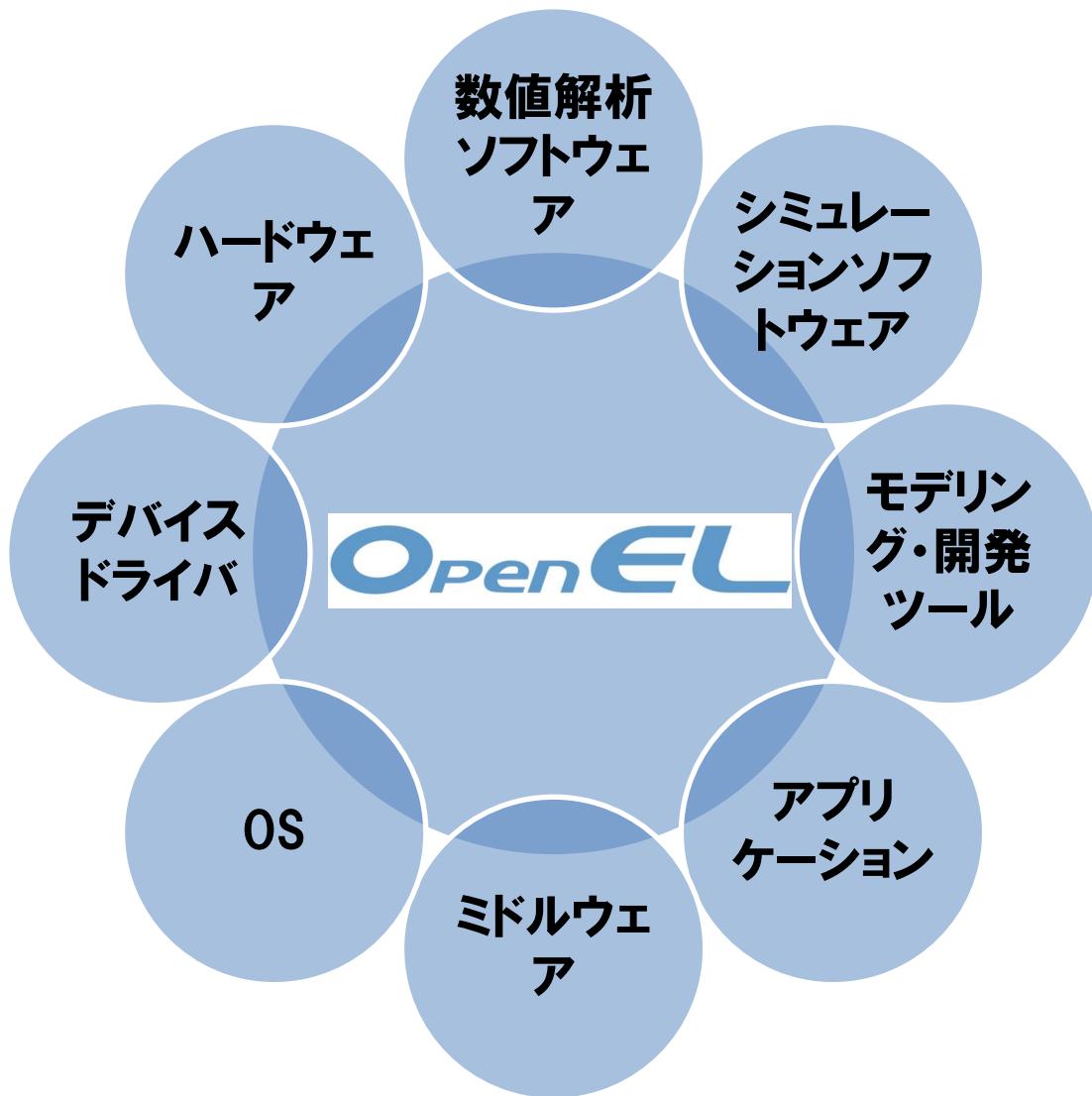
OpenELが変える組込みシステム開発

OpenEL

2019年11月22日
JASA技術本部セミナー
応用技術調査委員会 OpenEL WG 主査
中村憲一



OpenEL エコシステム



上流から下流までOpenELで一気通貫



分析

数値解析、
シミュレーション

設計

モデリング

アプリ実装

アプリケーション、
ミドルウェア、OS

デバドラ実装

デバイスドライバ

OpenEL



一般社団法人
組込みシステム技術協会
Japan Embedded Systems Technology Association



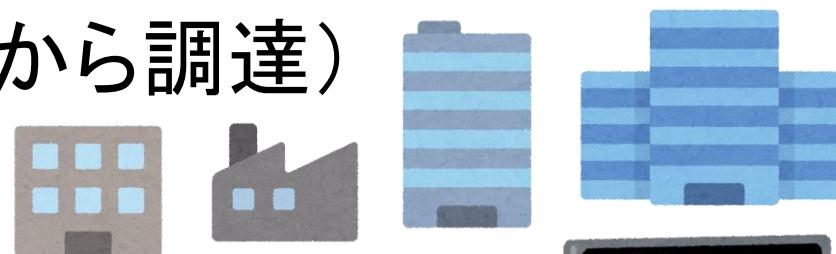
- 組込みシステム開発における課題
- 解決策
- OpenEL
- ロードマップ
- OpenEL WGのご紹介



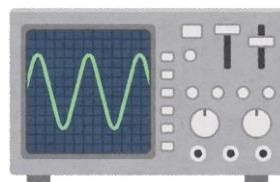
組込みシステム開発における課題

現状の組込みシステム開発のスタイル

- モデルベース開発ツールでアプリの設計後、ソースコードを自動生成
- 入出力デバイスのドライバは生成されない（ドライバのインターフェースが標準化されていない）ため、ドライバを新規開発（または様々なベンダーから調達）



- 手作業でドライバと結合後、実装
- 実機でテスト



組込みシステム開発における課題



- 組込み用の自動コード生成の品質に問題がある。
- 品質を維持するためのノウハウの蓄積が難しい。
- ドライバのコードは生成されないため、ドライバを用意する必要がある。
- なるべくプログラムを書きたくない。
 - ・ 瑕疵担保責任を負いたくない





■ メーカーの課題

- ・ ソフトウェアの開発効率
- ・ ソフトウェアの品質
- ・ 自社製以外のソフトウェアにも保証義務

■ ベンダーの課題

- ・ 他社製のソフトウェアとの相性は保証できない
- ・ ベンダーごとに異なるインターフェース仕様



解決策

解決策



- アプリからのインターフェース(API)とドライバ用のテンプレートを仕様化(効率向上)→OpenEL仕様
- センサーフュージョン(ドライバの合成)も実現可能
- バグが入りにくい(品質向上)、可読性が向上、レビューが容易
- テストパターンを自動生成し、シミュレーターによるテストの実行



- **品質と効率を上げる組込み用のモデルベース開発を実現！**

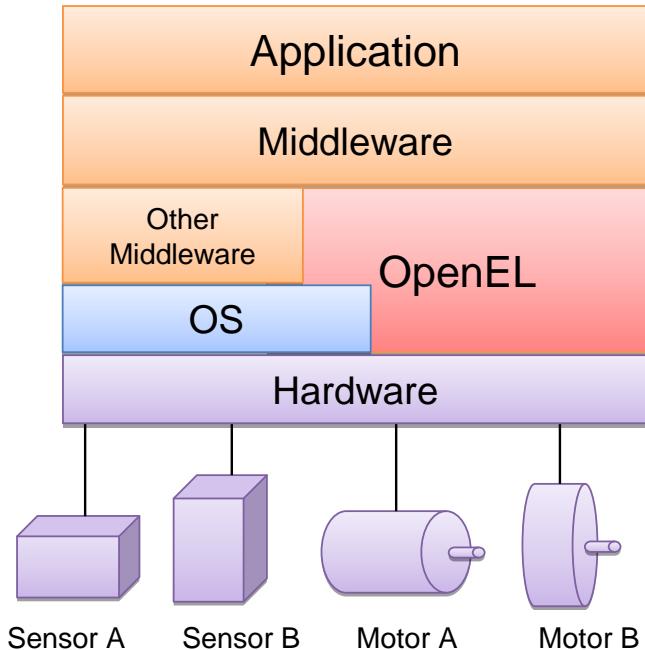


OPENEL

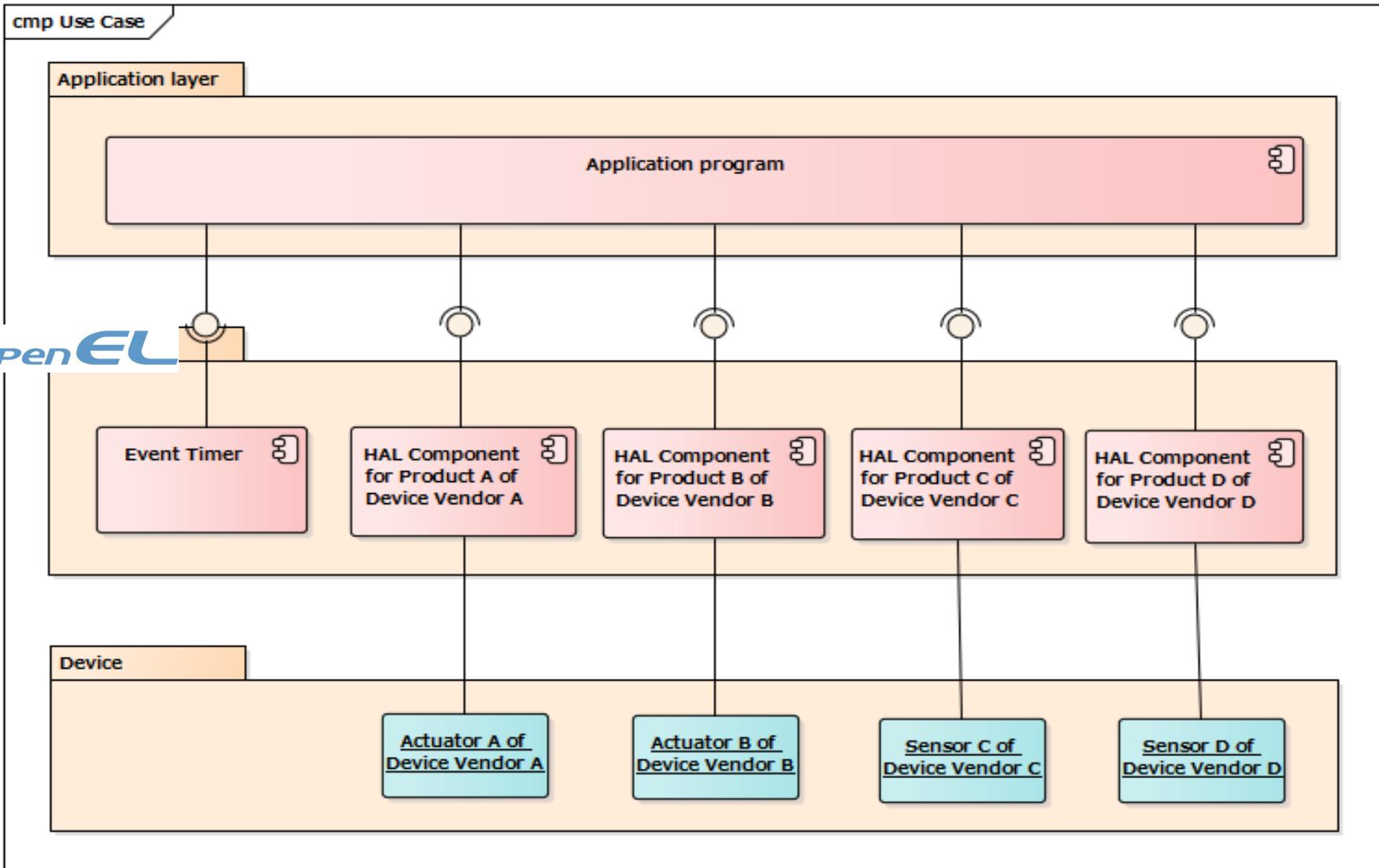
OpenEL(Open Embedded Library)とは?



- ロボットや制御システムなどのソフトウェアの実装仕様を標準化する組込みシステム向けのオープンなプラットフォーム



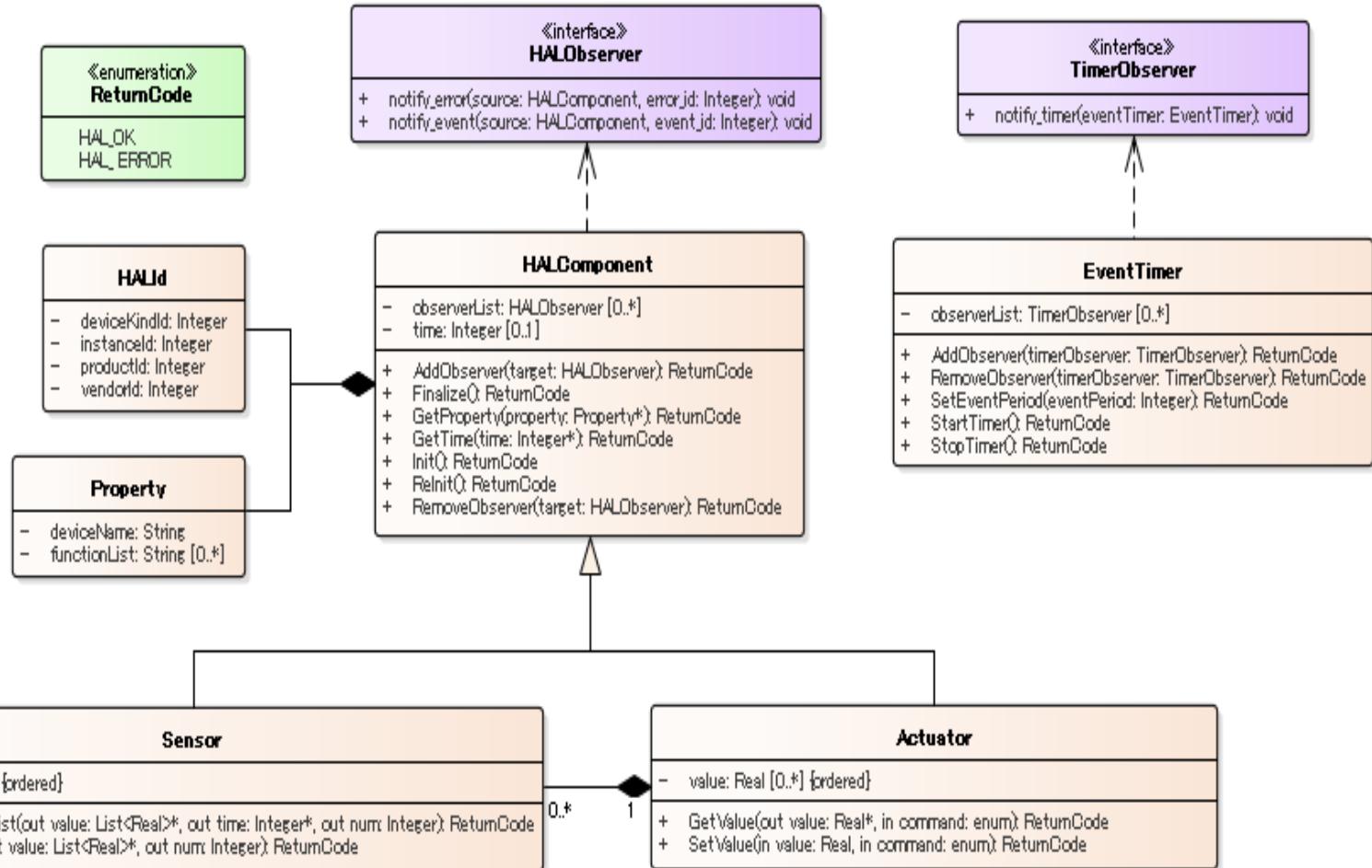
OpenEL 3.1



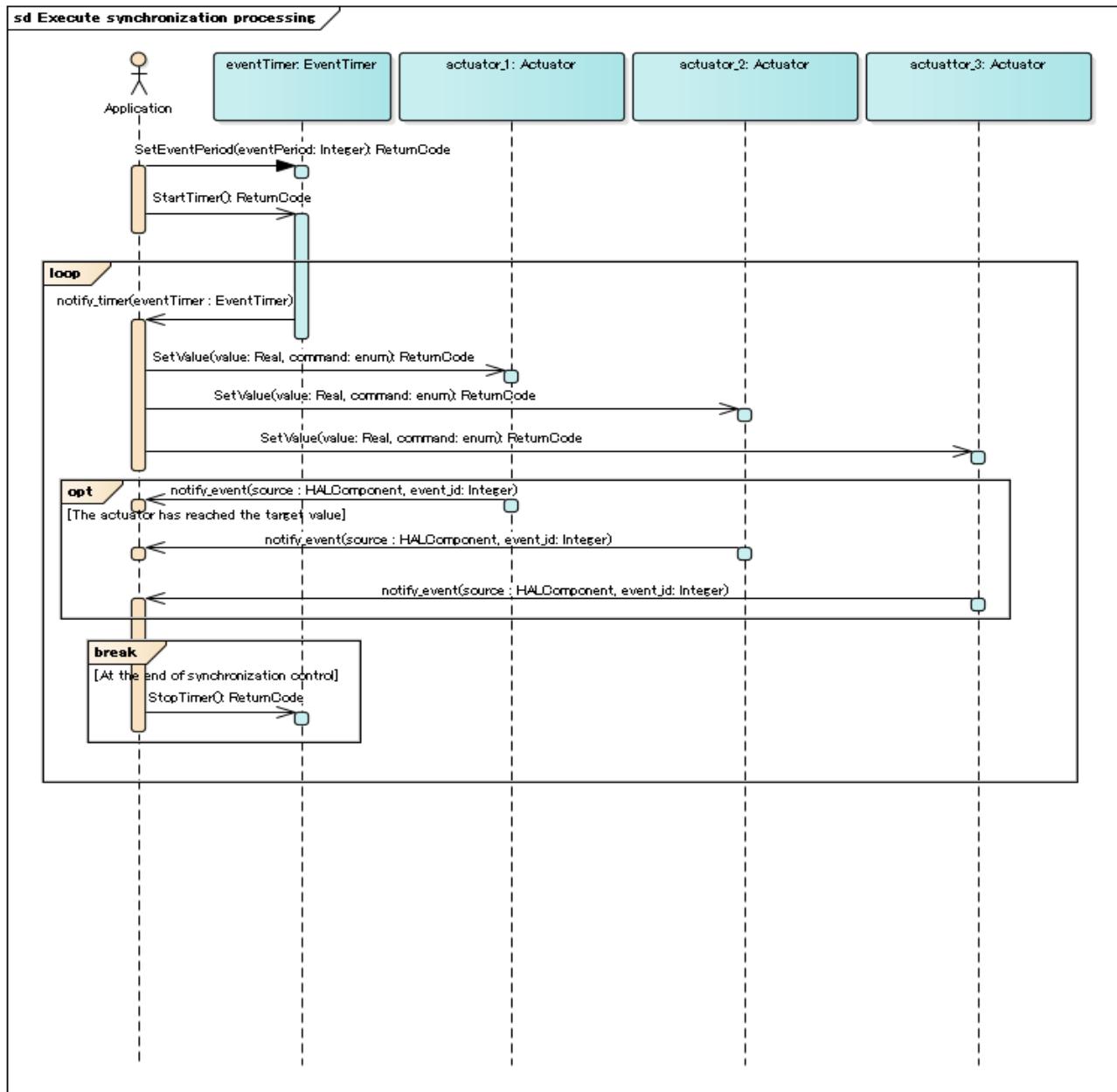
OpenEL 3.1



class Common



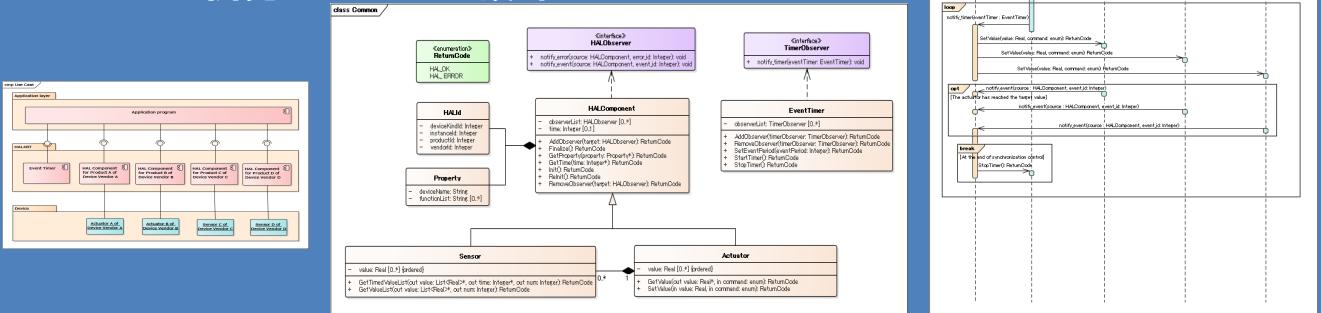
OpenEL 3.1



OpenELが変える組込みシステム開発



モデルベース開発ツールによる設計



OpenEL

モデルベース開発ツールによるコード生成
(OpenEL APIを使用するコードを生成)

```
#ifndef H_A_L_COMPONENT_H
#define H_A_L_COMPONENT_H

#include <string>
#include <vector>
#include <list>
#include <iostream>
#include <assert.h>
#include "HALComponent.h"
#include "ReturnCode.h"
#include "Property.h"

class HALComponent
{
private:
    HALObserver observerList;
    int time;

    HALObserver halObserver;

public:
    ReturnCode AddObserver(HALObserver target)
    {
        ReturnCode Finalize();
        /*
        */
    }

    ReturnCode HALComponent::AddObserver(HALObserver target)
    {
        return #;
    }

    ReturnCode HALComponent::Finalize()
    {
        return #;
    }

    ReturnCode HALComponent::GetProperty(Property property)
    {
        return #;
    }

    ReturnCode HALComponent::GetTime(int time)
    {
        return #;
    }
};

HALComponent@localhost:~/opt/freedom-e-sdk
```

The screenshot shows the generated C++ code for the HALComponent class. It includes private members for an observer list and time, and public methods for adding observers and finalizing. The generated code uses placeholder '#'. Below the code is a terminal window showing the compilation and execution of the generated code, which exits with code 0.

アクチュエーター制御やセンサー入力を
シミュレーションしてテストを実行

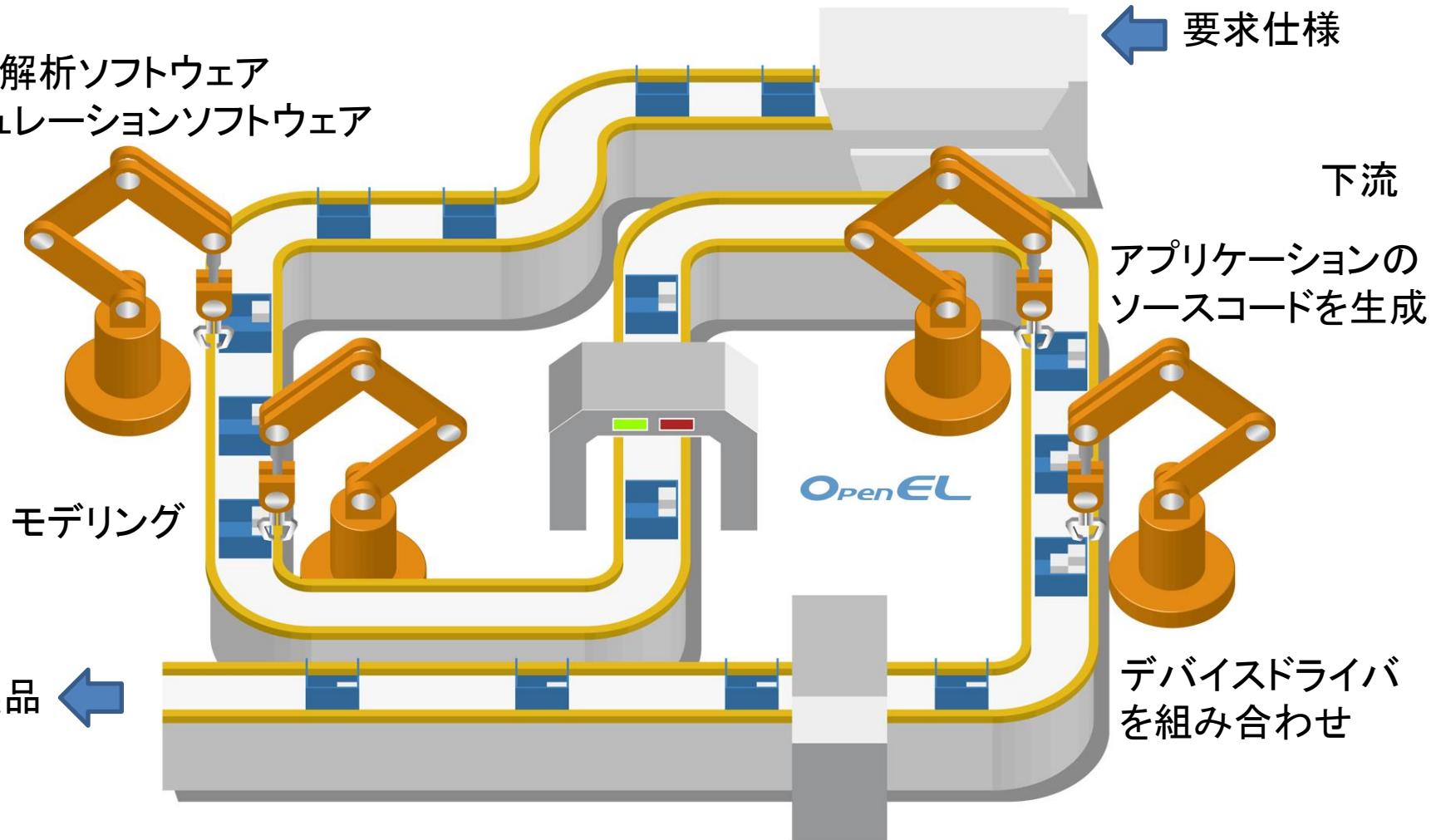
実機への実装、実機でのテスト

OpenELで組込みソフトウェア工場を実現



上流

数値解析ソフトウェア
シミュレーションソフトウェア



下流

アプリケーションの
ソースコードを生成

モデリング

製品



一般社団法人
組込みシステム技術協会
Japan Embedded Systems Technology Association



ロードマップ

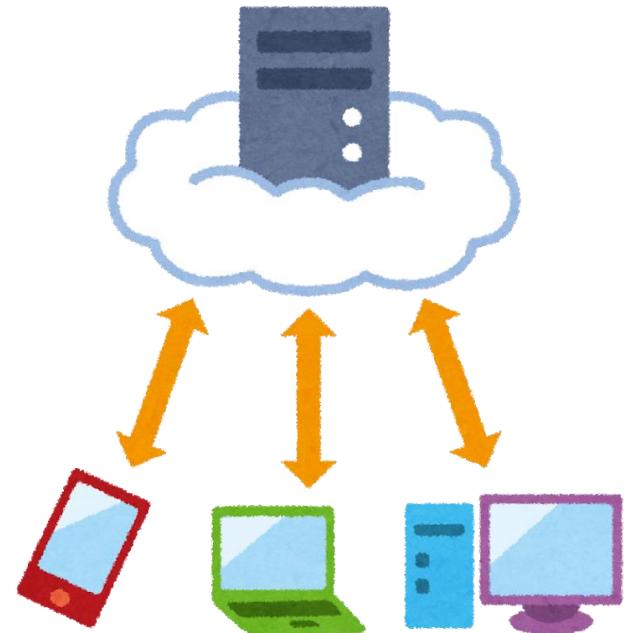


- サポートハードウェアの増加
 - ・ マイコンボード、モーター、センサー
- OpenEL各種ツールの整備
 - ・ ドライバのテンプレート生成
 - ・ ドライバのシミュレーター
- 組込み用の誰でも使えるモデルベース開発環境-OpenELエコシステムの提供
- 上流工程から下流工程まで一気通貫した開発の実現



将来

- AIや数値解析ソフトウェアを利用して、アプリケーションからドライバまで、上流工程から下流工程まで一気通貫したコードを自動生成
- クラウドとの連携
- クラウドで機能を定義し、デプロイ(実装)
- 物理シミュレーターとの接続





OPENEL WGのご紹介

OpenEL WGの活動概要



- OpenEL WGでは、OpenELの仕様の強化、OpenELの国内外における普及、OpenELの国際標準化の可能性の調査を行います。

1. OpenELの仕様の強化

- OpenEL バージョン3.2または4.0仕様書の発行(成果物:文書)
- OpenEL バージョン3.2または4.0の実装例のGitHubでの公開(成果物:ソフトウェア)
- OpenEL バージョン3.2または4.0のデモンストレーション(成果物:システム)

2. OpenELの国内外における普及

- OpenELの推進のための強固な体制を構築します。
- OpenELに関する情報をホームページなどで広く公開します。

3. OpenELの国際標準への提案

- OpenELを国際標準とするための提案先や提案方法の調査を行います。

OpenEL WGの開催日時



- 毎月1回(第4火曜日15:00～17:30、JASA会議室)
- 4/23(火)、5/28(火)、6/25(火)、7/23(火)、8/27(火)、
9/25(水)、10/29(火)、
- 11/26(火)、12/24(火)、
2020/1/21(火)、2020/2/25(火)、2020/3/24(火)



- <http://www.jasa.or.jp/TOP/openel/openel/>
- <https://github.com/openel>





OpenELが変える組込みシステム開発

2019/11/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋大伝馬町6-7
TEL: 03(5643)0211 FAX: 03(5643)0212
URL: <http://www.jasa.or.jp/>

本書の著作権は一般社団法人組込みシステム技術協会（以下、JASA）が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。