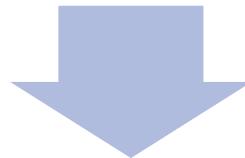


品質と効率を上げる組込み用の モデルベース開発を実現！

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。



OpenEL

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

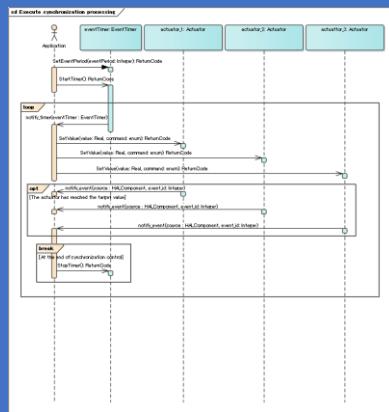
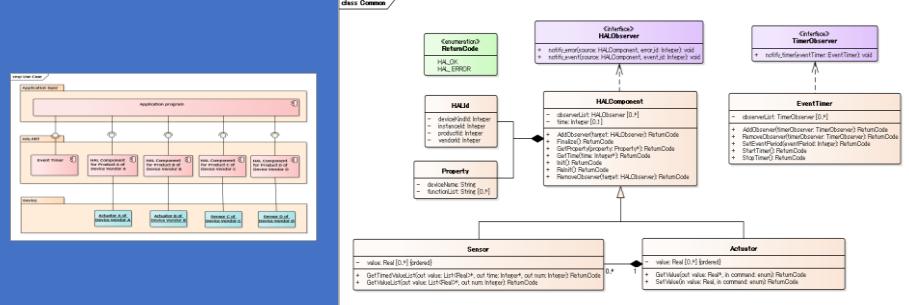


将来

- ・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一貫したコードを自動生成
- ・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_C_O_M_P_O_N_I_T_H
#define H_A_L_C_O_M_P_O_N_I_T_H
#include <string>
#include <vector>
#include <list>
#include <map>
#include <assert.h>
#include "HALServer.h"
#include "Property.h"
class HALComponent
{
private:
    HALObserver observerList[];
    int time;
    HALObserver HALObserver;
public:
    ReturnCode AddObserver( HALObserver target )
    {
        ReturnCode finalCode;
        /* */
        return finalCode;
    }
    ReturnCode Finalize();
    /* */
};

#endif // H_A_L_C_O_M_P_O_N_I_T_H
```

```
#include <string>
#include <vector>
#include <list>
#include <map>
#include <assert.h>
#include "HALComponent.h"
#include "HALServer.h"
#include "Property.h"
class HALComponent : public HALObserver
{
public:
    ReturnCode HALComponent::AddObserver( HALObserver target )
    {
        return 0;
    }
    ReturnCode HALComponent::Finalize()
    {
        return 0;
    }
    ReturnCode HALComponent::GetProperty( Property property )
    {
        return 0;
    }
    ReturnCode HALComponent::GetTime( int time )
    {
        return 0;
    }
    ReturnCode HALComponent::Init()
```

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

```
core_freq at 2749044 Hz
HalInit():0.12345678_1
Int( 0000000012345678 )
HalInit():0.00000001_1
Int( 0000000000000001 )
HalSetPosition()
HalGetPosition()
SetPosition_0000000012345678
GetPosition_0000000012345678
position=123.000000
HalSetPosition(1)
SetPosition_0000000012345678
position=123.000000
HalGetPosition()
This2>
HalGetPosition()
HAL_ID
0.000000
HalTargetSensorGetT.GetValue()
GetT.GetValue_0000000000000001
C:\>
```

The terminal output shows the simulation results for a sensor and actuator. The sensor value is 123.000000 and the actuator value is 0.000000.

実機への実装、実機でのテスト

<https://github.com/openel>
<https://www.jasa.or.jp/TOP/openel/openel/>
jasainfo@jasa.or.jp