TOPPERS/ASP+TECS+mruby

大山博司
オークマ株式会社 主管技師
名古屋大学大学院　情報科学研究科　客員教授
TOPPERS プロジェクト TECS WG 主査

TOPPERS

TECS と mruby. ２つのパッケージ

- mruby-on-asp+tecsパッケージ
  - mruby の VM を TECS コンポーネント化して、TOPPERS/ASP 上で動作
  - mruby スクリプトをコンパイルして、バイトコードを生成、セルに埋め込むところまでサポート
- MrubyBridgePlugin パッケージ
  - mruby スクリプトから TECS コンポーネントの受け口関数を操作するコード(C言語プログラム)を自動生成
  - Cygwin(POSIX環境) で動作するように構成
    - 現状パッケージの状態。生成コードはASP上でも動作
  - C 言語の関数に直接インタフェースするコードを生成する Mruby2CBridgePlugin も同梱

mruby-on-asp+tecsパッケージ

mruby-on-asp+tecsパッケージの概要

- mruby VM を TECS コンポーネント化し、TOPPERS/ASP 上で動作させる
  - このパッケージでは、skyeye をプラットフォームとして実装されている
- mruby VM コンポーネントの属性に mruby スクリプトファイル (.mrb) を指定すると、TECS ジェネレータ(プラグイン)により、自動的に mruby のコンパイルが行われる
  - コンパイルそのものは mrbc が行なう
  - コンパイルされたバイトコードがmruby VM に渡され、ソースコードは実行環境には渡されない
  - mruby VM コンポーネントは、コンパイラを含まないように構成 (VM 本体のみ)

TOPPERSとmrubyの連携

- TOPPERSのOS上でmrubyを動作
  - TOPPERS/ASP +TECS
    - SkyEye(ARMのシミュレータ)
    - Mindstorms NXT
  - TOPPERS/ASP
    - SH4
  - TOPPERS/ATK1

http://www.csk.com/fukuoka/services/mruby/nxt01.html

- mruby⇒TECSブリッジ
  - mrubyからTECSコンポーネントの呼び出し

TOPPERS

SkyEye(ARMのシミュレータ)上での動作

- cd asp+tecs/sample_mruby
- make tecs
- make depend
- make
- ../../bin/skyeye.exe –e asp.exe

```
mrbtest - Embeddable Ruby and TECS Test

This is a very early version, please test and report errors.
Thanks :)

Sample program starts.
Hello World
Sample program ends.
```

サンプルプログラムの内容

asp+tecs/sample_mruby/mrb_sample.rb

```
puts("Hello World")
```

../../mruby/bin/mrbc.exe BMruby_irep -oMruby_mrb.c mrb_sample.rb;

mruby用のバイトコードの生成

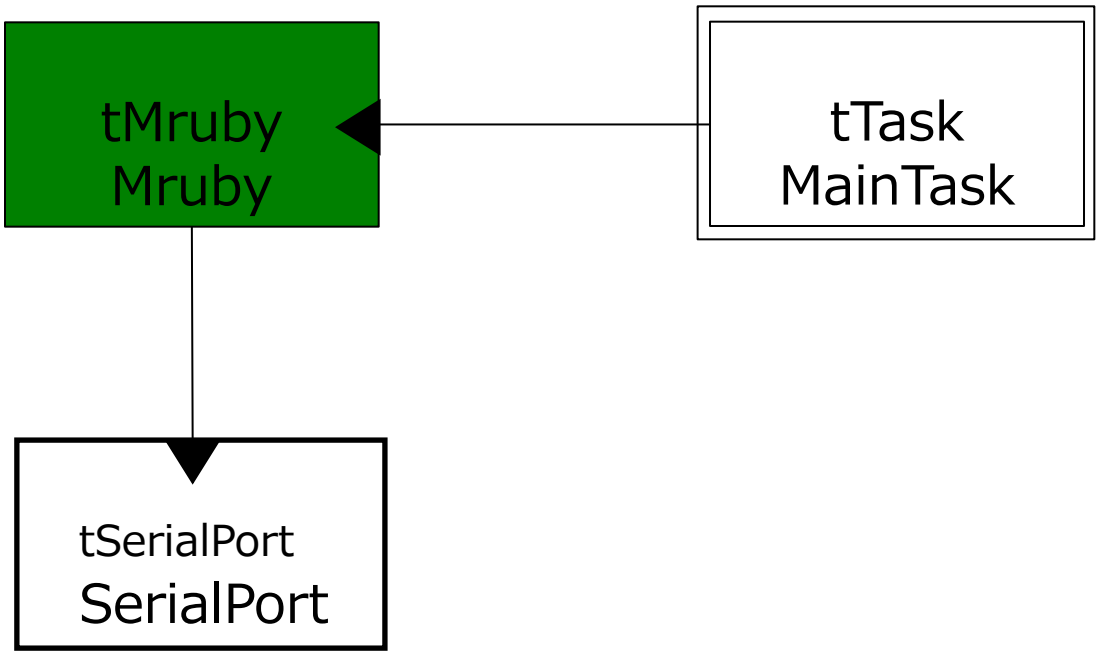
asp+tecs/sample_mruby/Mruby_irep.ctmp

```
const char Mruby_irep [] = {
0x52,0x49,0x54,0x45,0x30,0x30,0x30,0x39,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x39,
0x30,0x30,0x30,0x30,0x4d,0x41,0x54,0x5a,0x20,0x20,0x20,0x20,0x30,0x30,0x30,0x39,
0x30,0x30,0x30,0x30,0x00,0x00,0x00,0x7e,0x00,0x01,0x00,0x00,0x20,0x20,0x20,0x20,
0x20,0x20,0x20,0x20,0xaa,0x60,0x00,0x00,0x00,0x40,0x53,0x43,0x00,0x02,0x00,0x04,
0x00,0x02,0x6f,0x28,0x00,0x00,0x00,0x04,0x01,0x00,0x00,0x06,0x01,0x80,0x00,0x3d,
0x01,0x00,0x00,0xa0,0x00,0x00,0x00,0x4a,0x8a,0x21,0x00,0x00,0x00,0x01,0x10,0x00,
0x0b,0x48,0x65,0x6c,0x6c,0x6f,0x20,0x57,0x6f,0x72,0x6c,0x64,0x36,0x8d,0x00,0x00,
0x00,0x01,0x00,0x04,0x70,0x75,0x74,0x73,0x24,0x89,0x00,0x00,0x00,0x00,
};
```


TOPPERS

サンプルプログラムのコンポーネント図

※一部省略

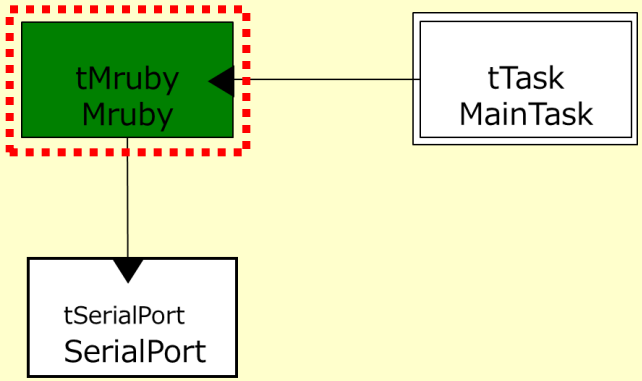


TOPPERS

サンプルのコンポーネント記述

asp+tecs/sample_mruby/tMruby.cdl

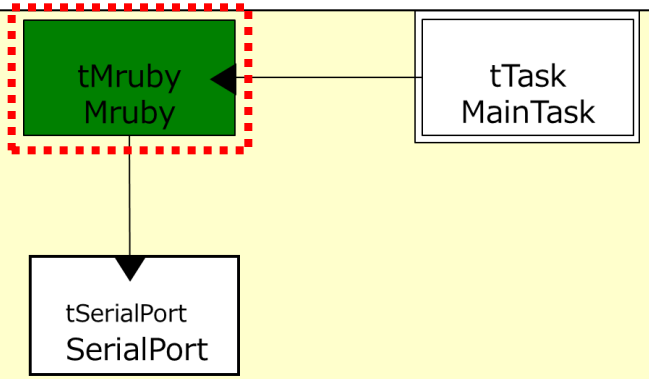
```
signature sTaskBody{
        void main(void);
};


celltype tMruby{

        entry sTaskBody eMrubyBody;
        attr{
                [omit]char_t *mrubyFile;
                char_t *irep=C_EXP(&$cell$_irep);
        }

        /* write関数の接続先 */
        [optional]      call sSerialPort cSerialPort;
};
```



TOPPERS

サンプルのコンポーネント記述

asp+tecs/sample_mruby/tMruby.cdl




```
celltype tMruby{

…

 factory{
   write("Makefile.tecsgen",
           "POST_TECSGEN_TARGET := $(POST_TECSGEN_TARGET) $cell$_mrb.c");
   write("Makefile.tecsgen", “$cell$_mrb.c : %s", mrubyFile);
   write("Makefile.tecsgen",
           " $(MRBC) –B$cell$_irep –o$cell$_mrb.c %s", mrubyFile);
   write("Makefile.tecsgen", "TECS_COBJS := $(TECS_COBJS) $cell$_mrb.o");
   write("$ct$_factory.h","extern const char_t * $cell$_irep;");
  };

};
```


TOPPERS

生成されるMakefile.tecsgen

asp+tecs/sample_mruby/gen/Makefile.tecgen

セル名が**Mruby**, **Mruby2**の場合
Mrubyのファイルが**mrb_sample.rb**
mrb_sample2.rbの場合

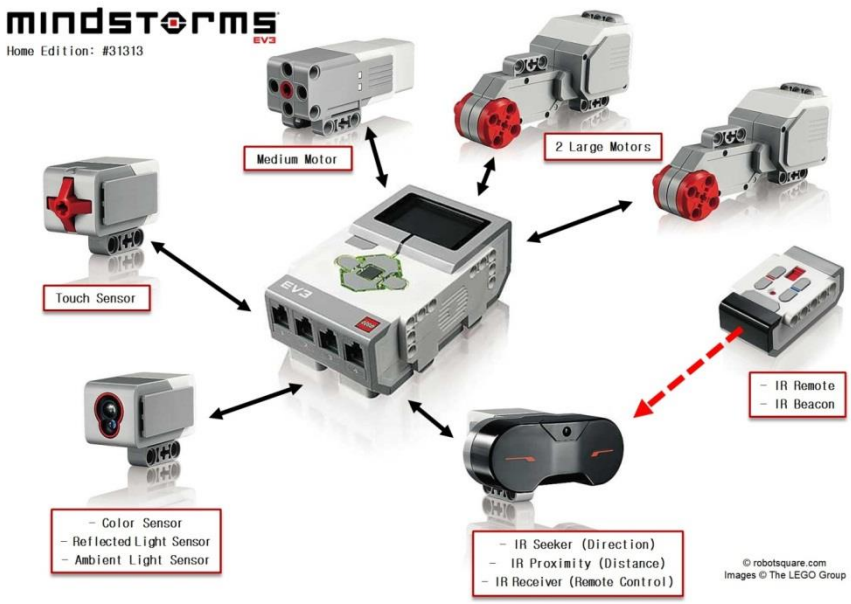
```
POST_TECSGEN_TARGET := $(POST_TECSGEN_TARGET) Mruby_mrb.c
Mruby_mrb.c : mrb_sample.rb
        $(MRBC) -BMruby_irep -oMruby_mrb.c mrb_sample.rb
TECS_COBJS := $(TECS_COBJS) Mruby_mrb.o
POST_TECSGEN_TARGET := $(POST_TECSGEN_TARGET) Mruby2_mrb.c
Mruby2_mrb.c : mrb_sample2.rb
        $(MRBC) -BMruby2_irep -oMruby2_mrb.c mrb_sample2.rb
TECS_COBJS := $(TECS_COBJS) Mruby2_mrb.o
```

TOPPERS/ASP+TECS+mrubyの使い方

- mrubyの使いどころ
  - 組込み機器のテストケース入力として利用
    - デバイスドライバのテスト
  - UIのカスタマイズ
    - 自動販売機
  - 分散対応
    - プロセッサ間、ネット-ワーク間の通信は、TECSでサポートできる

現在対応中

- LEGO®MINDSTORMS EV3（9月発売予定）
  - リアルタイムOS移植
  - TECS化（ドライバ等含む）
  - mrubyで動作



TOPPERS

MrubyBridgePlugin パッケージ

MrubyBridgePlugin パッケージの概要

- Cygwin(POSIX) 上で動作するように構成
  - TECS ジェネレータの生成する Makefile (テンプレート)がそのまま使える
  - TOPPERS/ASP 等の環境へは、 mruby-on-asp+tecs パッケージを参考に移植してください
- シグニチャプラグイン (MrubyBridgePlugin, Mruby2CBridgePlugin) により、mruby ⇒ TECS(C言語)への変換コードを生成

2つのブリッジプラグインタイプ

- MrubyBridgePlugin ← **本日はこちらがメイン**
  - TECS コンポーネントの受け口関数を mruby スクリプトから呼び出し可能にする
  - 複数VMがある場合、登録しわけることができる
- Mruby2CBridgePlugin
  - インタフェース定義(signature記述)を、そのまま C の関数名として呼び出す
    - TECS の知識不要
      - 実際には signature 記述の書き方を理解する必要があるが、見よう見まねで習得可能なレベル
    - MrubyBridgePlugin と TECS2CBridgePlugin を連結して実現
      - TECS2C 変換はインライン関数なのでオーバーヘッドなし

mruby⇒TECSブリッジで、できること

- mruby ⇒ 受け口関数のI/Fの自動生成

- C言語で記述したデバイスドライバを簡単にmrubyからアクセス可能
- TOPPERS/ASP など リアルタイムOS のオブジェクトをmruby から操作可能
- (FatFS（ファイルシステム）, TINET（ネットワークプロトコルスタック） も容易にmruby から操作可能)

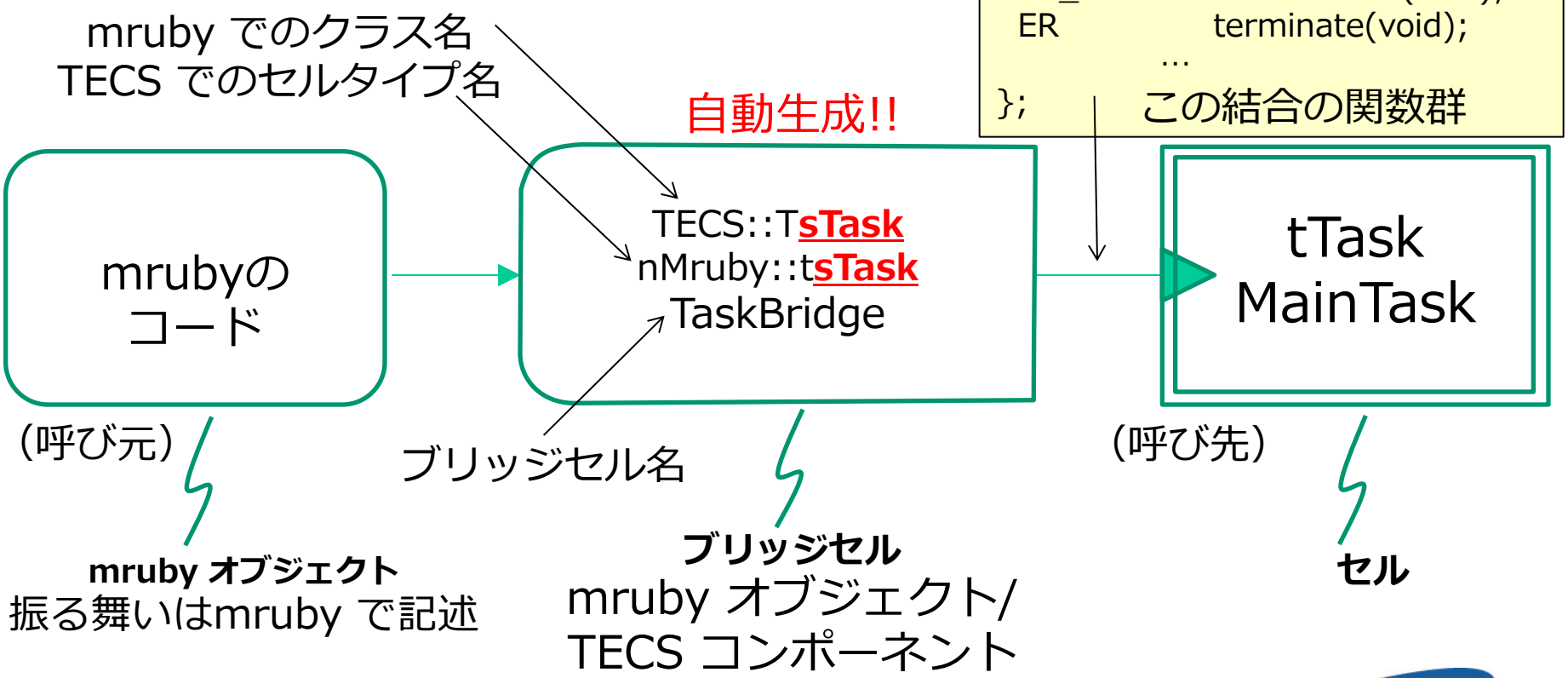
mruby⇒TECSブリッジ

ユーザーのCDL上に現れないセルを省略 (複数VMの頁を参照)

- mruby ⇒ TECS(C 言語)のI/Fの自動生成
  - TECSのインタフェース(signature)定義を利用
  - 以下はタスクを利用する例

```
signature sTask {
  ER            activate(void);
  ER_UINT  cancelActivate(void);
  ER            terminate(void);
                …
};
```

この結合の関数群



TOPPERS

mruby⇒TECSブリッジサンプルコード

- TOPPERS/ASPのタスクを呼び出す例

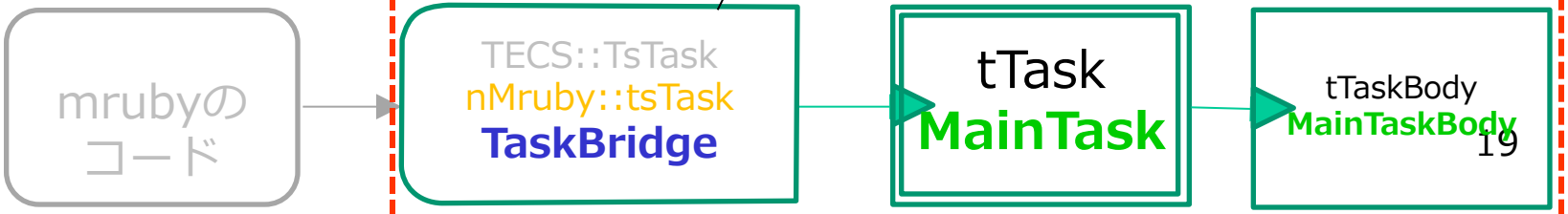
TECSのコンポーネント記述

```
generate( MrubyBridgePlugin, sTask, "" );
cell tTask MainTask {
        /* 呼び口の結合 */
        cBody = MainTaskBody.eTaskBody;
        /* 属性の設定 */
        priority = 1;
        stackSize = C_EXP("STACK_SIZE");
};

cell nMruby::tsTask TaskBridge{
        cTECS = MainTask.eTask;
};
```

mrubyに提供する機能の選択
この場合はsTask

タスクの定義（簡略化）

```
celltype tTask {
   entry   sTask     eTask;
   entry   siTask    eiTask;
   call    sTaskBody
           cBody;
};
```



TOPPERS

mruby⇒TECSブリッジサンプルコード

- TOPPERS/ASPのタスクを呼び出す例

```
signature sTask {
        ER          activate(void);
        ER_UINT     cancelActivate(void);
        ER          terminate(void);
        …
};
```

sTaskで定義されている
機能を選択

mrubyのコード

```
taskBridge = TECS::TsTask.new “TaskBridge“   ← タスククラス

taskBridge.activate   ← タスクの起動
```

TOPPERS

プラグインのメッセージ

- TECS ジェネレータ tecsgen 実行時、プラグインが出力するメッセージを見れば、どのような名前が付けられたか分かる

– [signature] の行に ブリッジのセルタイプ名、クラス名を表示
  セル名は任意で付けてよい

MrubyBridgePlugin: [signature] ::sTask
  => [celltype] nMruby::tsTask => [class] TECS::TsTask

– 初期化セルへの結合
  VM_TECSInitializer が生成されるので、VM の初期化呼び口から結合する

MrubyBridgePlugin: join your VM's cInitialize to
  VM_TECSInitializer.eInitialize

TOPPERS

生成されるコード

tTask の場合

- gen/nMruby_tsTask.c
  - インタフェースコードの本体
    - signature sTask の関数群をインタフェースする関数
  - initialize メソッドのコード
- gen/nMruby_tTask_Initializer.c
  - mruby VM へ登録するコード

扱える引数

- in, out, inoutのみ
  - これらはいずれも呼び元が記憶域を準備する
  - send/receive は不可
- 逸脱にあたるものは、取扱い不可
  - void へのポインタ、ポインタ多重度が合わない
- ポインタ型は、記憶域を確保して渡す
  - 記憶域を確保する mruby のクラスを用意
  - アドレス値を渡したい場合は intptr_t 型を使う
- 構造体型のメンバーは基本型のみ
  - 整数型（含文字型、ブール型)、浮動小数は可
  - 構造体型、ポインタ型、配列型は不可

扱える戻り値

- 基本型 or void 型のみ
  - 整数型（含文字型、ブール型)、浮動小数は可
  - void 型の場合 nil が返される
  - 構造体型、ポインタ型は不可

ポインタの扱い

- 用意されたポインタクラスを利用する
- PRI * の場合 (プライオリティへのポインタ)
  - TECS::IntPointer … mruby クラス名
    - 元の型名からクラス名が決められる
    - プラグインメッセージを見れば、クラス名が分かる

MrubyBridgePlugin: [pointer] PRI* => class TECS::IntPointer

-- mruby のコード --

p_pri = TECSPointer.new(1) # 配列要素数を指定(非配列なら1)

taskBridge.**getPriority** *p_pri* # プライオリティを取得

print *p_pri.value* # 配列形式でもアクセス可 *p_pri[0]*

参考

- nMruby::tIntPointer…TECS のセルタイプ名
  - これを操作することはない．mruby VM にクラスを登録するために用いられる．プラグインにより必要な結合が行われる．

TOPPERS

構造体の扱い

- プラグインが自動生成するクラスを使用する
- T_RTSK の場合 (タスクの属性参照の結果を返す構造体)
  - TECS::Structt_rtsk … クラス名
    - タグ名を元にクラス名が決定される
    - プラグインメッセージを見ればクラス名が分かる

```
MrubyBridgePlugin: [struct] T_RTSK => class TECS::Structt_rtsk
        -- mruby のコード --
        pk_taskStatus = TECS::Structt_rtsk.new   # 記憶域準備
        taskBridge.refer pk_taskStatus          # refere を呼び出す
        print "TaskBridge.refer¥n"
        print "    tskstat=", pk_taskStatus.tskstat, "¥n"
        print "    tskpri=", pk_taskStatus.tskpri, "¥n"
        print "    tskbpri=", pk_taskStatus.tskbpri, "¥n"
        …
```

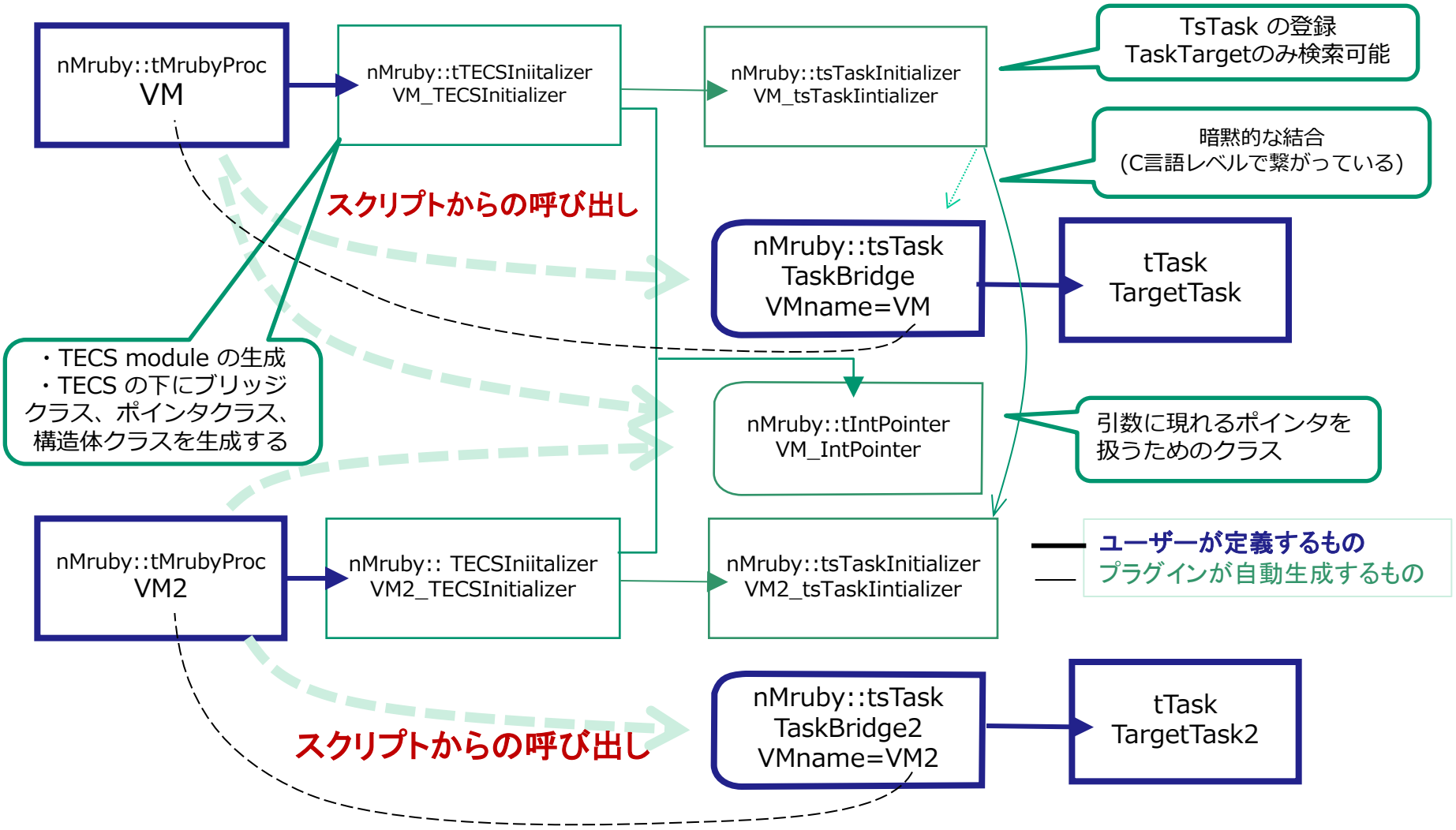
オプション

- プラグインオプション
  - exclude … 不要関数登録抑止
  - include … 必要関数のみ登録
    - exclude と include は、どちらか一方のみ指定可
    - メモリ消費の低減、アクセス制限に利用する
  - ignoreUnsigned … ポインタ型の signedness を無視する

- コンパイルオプション
  - TECS_NO_VAL_CHECK … 整数範囲のチェックを抑止

複数VMがある場合

プラグインが内部で生成し
結合するものも示している



TOPPERS

CDL コード

```
import( “cygwin_kernel.cdl” );                  // POSIX 用(cygwin 以外でも可)
import( “mruby.cdl” );
generate( MrubyBridgePlugin, sTask, “”);        // sTask を指定してMrubyBridePlugin を適用
cell TargetTask {                               //  TargetTask の定義
    …                                           // 属性指定を省略
};
cell nMruby::tsTask TaskBridge {                // TaskBridge　へのブリッジセル
            cTECS = TargetTask.eTask;
};
cell nMruby::MrubyProc VM {                     // MrubyVM (main が定義されているタイプのVM)
            cTECSInitializer = VM_TECSInitialize.eInitializer;
                                    // VM_TECSInitializer はプラグインにより自動生成される
};
cell TargetTask2 {                              //  TargetTask2 の定義
    …                                           // 属性指定を省略
};
cell nMruby::tsTask TaskBridge2 {               // TaskBridge2　へのブリッジセル
            cTECS = TargetTask2.eTask;
            VMname = “VM2;                      // ← VM セルの名前を指定
};
cell nMruby::MrubyProc VM2 {                    // MrubyVM
            cTECSInitializer = VM2_TECSInitialize.eInitializer;
                                    // VM2_TECSInitializer はプラグインにより自動生成される
};
```

1個目のVM設置

2個目のVM設置

TOPPERS

おまけ Mruby2CBridgePlugin

- TECS の知識が、ほとんど不要なバージョン
  - mruby から C へインタフェースするプログラムを生成
  - signature の各関数を、そのまま C の関数とみなして呼び出す
  - MrubyBridgePlugin に TECS2CBridgePlugin を組合わせて実現
    - TECS2CBridgePlugin は inline 実装なのでオーバーヘッドなし (TECS の関数名から、Cの関数名への変換を行う. TECSでは関数名に、セルタイプ名、受け口名が付加されている)
    - 初期化用に C2TECSBridgePlugin も利用
- TECS開発ブログ参照
  - http://tecs22022.blog85.fc2.com/

サンプルコード

- MrubyBridge
  - SimpleSample の例
- MrubyBridge-ASP
  - TOPPERS/ASPのsignature群のテストコード
  - MemroyPool のみ扱えない (void *のため)
  - ビルドには TOPPERS/ASPのパッケージが必要
- MrubyBridge-test
  - 扱える引数、戻り値を一通りテストするコード
- Mruby2C-File
  - Mruby2CBridgePlugin で低水準、および高水準のファイル I/O 関数を扱う例

参照ドキュメント

- MrubyBridge.txt
  - 一通りの説明が書かれている
- TECSPointer.txt
  - ポインタクラスについての説明
- TECSStruct.txt
  - 構造体クラスについての説明