

ステータス:	担当	開始日:	2017/09/12
優先度:	通常	期日:	
担当者:	katami	進捗率:	100%
カテゴリ:		予定工数:	0.00時間
対象バージョン:			

## 説明

#3704 の動作確認において、rthsell (rtls localhost/ -l) で確認したところ、コンポーネントがActiveであることを確認できたが、openrtp(RTSE)で確認したところ、Sensor0.rtcだけがActiveで、Motor0.rtc、Controller0.rtcはInactiveの状態だった。  
openrtpのバージョンは eclipse442-openrtp112v20160526-ja-linux-gtk-x86\_64。

動作確認は、examples/Composite のSensorCompを利用して、初期化関数を以下のように変更

- ControllerとMotorはrtc.confで.soをロードしておく。
- 以下のように、SensorのECを取り出し、Controller, Motor にアタッチ
- Motor, Controller, Sensor の順で activate した。

```
public void myModuleInit(Manager mgr) {
    Properties prop = new Properties(Sensor.component_conf);
    mgr.registerFactory(prop, new Sensor(), new Sensor());

    System.out.println("1");
    // Create a component
    RTObject_impl scomp = mgr.createComponent("Sensor");
    if( scomp==null ) {
        System.err.println("Component create faile);."
        System.exit(0);
    }
    System.out.println("2");
    RTC.RTObject srtc = scomp.getObjRef();

    System.out.println("3");
    // Create a component
    RTObject_impl ccomp = mgr.createComponent("Controller");
    if( ccomp==null ) {
        System.err.println("Component create faile);."
        System.exit(0);
    }
    System.out.println("4");
    RTC.RTObject crtc = ccomp.getObjRef();

    System.out.println("5");
    // Create a component
    RTObject_impl mcomp = mgr.createComponent("Motor");
    if( mcomp==null ) {
        System.err.println("Component create faile);."
        System.exit(0);
    }
    System.out.println("6");
    RTC.RTObject mrtc = mcomp.getObjRef();

    System.out.println("7");
    ExecutionContextListHolder eclist = new ExecutionContextListHolder();
    eclist.value = srtc.get_owned_contexts();

    System.out.println("8");
    eclist.value[0].add_component(crtc);
    System.out.println("9");
```

```

eclist.value[0].add_component(mrtc);
System.out.println("10");

eclist.value[0].activate_component(mrtc);
System.out.println("11");
eclist.value[0].activate_component(crtc);
System.out.println("12");
eclist.value[0].activate_component(srvc);
System.out.println("13");
}

```

## 履歴

#1 - 2017/09/12 17:08 - katami

この動作確認では、participating EC から RTCをactivateしました。通常、openrtp(RTSE)のRTCの状態はオーナーECから取得した状態を示します。openrtp(RTSE)を使用する場合は、実行コンテキストビューで実行コンテキストをparticipatingに切り替えて確認することができます。  
[ExecutionContext.png](#)

[#3704](#) の不具合の報告では、participating EC から RTCをactivateして、そのparticipating ECがRTCの状態を取得してactivate状態を確認していますので、[#3704](#) の対応で問題ないと思われれます。  
<https://github.com/fkanehiro/hrpsys-base/blob/master/python/rtm.py#L454>

以下は、[#3704](#) の動作確認の説明です。

RTCは複数のEC(owned、participating)を持つことができ、ECはRTCをキーとしてステートマシン(RTObjectStateMachine)を保持します。

Motor, Controller, Sensorコンポーネントはそれぞれ owned ECを持ちます。

テストで使用したプログラム(Sensor Comp)では、

```
eclist.value = srvc.get_owned_contexts();
```

で Sensorのowned ECを取得して、

```

System.out.println("8");
eclist.value[0].add_component(crtc);
System.out.println("9");
eclist.value[0].add_component(mrtc);
System.out.println("10");

```

でコンポーネントを追加しています。

ここで、Motor, Controller は Sensorの owned EC を participating EC に登録して、参加します。

```

eclist.value[0].activate_component(mrtc);
System.out.println("11");
eclist.value[0].activate_component(crtc);
System.out.println("12");
eclist.value[0].activate_component(srvc);
System.out.println("13");

```

でコンポーネントをactivateしています。

Motor, Controller は participating EC からactivateされます。

この状態で、Motor, Controller の owned EC から RTCの状態を取得する(get\_component\_state())とINACTIVEのままですが、Motor, Controller の participating EC から RTCの状態を取得する(get\_component\_state())とACTIVEとなります

これは、ECがRTCをキーとしてステートマシン(RTObjectStateMachine)を保持しているためです。

以下を追加して再度テストしてACTIVEすることを確認しました。

```

try{
    Thread.sleep(3000);
}catch(InterruptedException e){}

```

```

System.out.println("14");
LifecycleState state;
state = eclist.value[0].get_component_state(crtc);
if (state == LifecycleState.ACTIVE_STATE) {
    System.out.println("Controller is ACTIVE_STA");}
}

```

```
System.out.println("15");
state = eclist.value[0].get_component_state(mrtc);
if (state == LifecycleState.ACTIVE_STATE) {
    System.out.println("Motor is ACTIVE_STA");
}
System.out.println("16");
```

participating EC から activate された場合でも、RTCでは、onActivatedのイベントが発生します。引数のECのIDで、participating ECから activateされたイベントだということがわかります。

#2 - 2017/09/12 17:09 - katami

- ステータスを新規から担当に変更
- 担当者を katami にセット

#3 - 2017/09/13 10:59 - katami

- ファイル ExecutionContext.png を追加

#4 - 2017/09/13 11:18 - katami

- 進捗率を 0 から 100 に変更

## ファイル

---

ExecutionContext.png	42 KB	2017/09/13	katami
----------------------	-------	------------	--------