

ステータス:	担当	開始日:	2016/11/07
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:	RELEASE_1_2_0		
説明			
#3697 と同じ			

関係しているリビジョン

リビジョン 1038 - 2017/09/07 16:29 - t-katami

[compat,bugfix,->releg_1_2] EC activation bug for multiple RTCs when they activated from participants RTCs. refs #3704

履歴

#1 - 2017/09/07 16:05 - katami

- ステータス を 新規 から 担当 に変更

動作確認

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).\n");
    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).\n");
    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).\n");
    System.exit(0);
}
System.out.println("6");
RTC.RTObject mrtc = mcomp.getObjRef();
```

```
System.out.println("7");
ExecutionContextListHolder eclist = new ExecutionContextListHolder();
```

```
eclist.value = src.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(src);  
System.out.println("13");  
}
```

```
./Sensor.sh -o exec_cxt.periodic.rate:1
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13
```

```
Sensor Activated  
Controller Activated  
Motor Activated
```

```
$ rtls localhost/ -l  
Active 2/0 1/0 1/0 0/0 Motor0.rtc  
Active 2/0 1/0 1/0 0/0 Sensor0.rtc  
Active 2/0 1/0 1/0 0/0 Controller0.rtc  
- - - - - ubuntu.host_cxt
```

#2 - 2017/09/08 14:20 - katami

- 担当者を katami にセット
- 対象バージョンを RELEASE_1_2_0 にセット
- 進捗率を 0 から 100 に変更

#3 - 2017/09/13 11:22 - katami

テスト用のプログラムを変更して再度確認

```
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 src = 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(srtc);
System.out.println("13");
```

```
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");
}
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
Sensor Activated ec_id=0
Controller Activated ec_id=1000
Motor Activated ec_id=1000
14
Controller is ACTIVE_STATE
15
Motor is ACTIVE_STATE
16
```

openrtp(RTSE)では、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、ACTIVEであることを確認した。