OpenRTM-aist (C++) - 機能 #2349

RTObject_impl::initialize() 内の ECの作成方法の修正

02/04/2012 03:54 AM - n-ando

Status:	終了	**Start date:**	02/04/2012
Priority:	通常	**Due date:**	
Assignee:	n-ando	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

RTObject_impl::initialize() 内のECの作成方法の修正を行う。

主な変更点は

- RTC個別configファイル(プライベートオプション)に従ってECを作成する。
- 個別configが指定されない場合、rtc.confのオプション(グローバルオプションと呼ぶ) に従ってECを作る
- プライベートオプションではECは複数個作成することができる (execution_contextオプション)
- 個別config内で、ECのタイプ毎、ECのインスタンス名ごとにオプションを指定できる
- 個別configでECをnone指定するとECを1つも作成・アタッチせずにRTCを起動できるようにする
- 個別configではECに名前をつけることができる
- 同名のECがすでに存在する場合、新たにインスタンスを作成せずに既存のものを使う
- サンプルのrtc.confとcompoennt.confに変更を反映させる。

```
#============================================================
# execution context options
#============================================================
#
# Periodic type ExecutionContext
#
# Other availabilities in OpenRTM-aist
#
# - ExtTrigExecutionContext: External triggered EC. It is embedded in
#                            OpenRTM library.
# - OpenHRPExecutionContext: External triggred paralell execution
#                            EC. It is embedded in OpenRTM
#                            library. This is usually used with
#                            OpenHRP3.
# - RTPreemptEC:             Real-time execution context for Linux
#                            RT-preemptive pathed kernel.
# - ArtExecutionContext:     Real-time execution context for ARTLinux
#                            (http://sourceforge.net/projects/art-linux/)
# exec_cxt.periodic.type: [specify periodic type EC]

#
# The execution cycle of ExecutionContext
#
# exec_cxt.periodic.rate: [Hz]

#
# Event driven execution context (not implemented yet)
#
# exec_cxt.event_driven.type: [specify event driven type EC]
#


#
# State transition mode settings YES/NO
#
# Default: YES (efault setting is recommended.)
#
# Activating, deactivating and resetting of RTC performs state
# transition. Some execution contexts might execute main logic in
# different thread. If these flags are set to YES, activation,
```

```
# deactivation and resetting will be performed synchronously. In other
# words, if these flags are YES,
# activation/deactivation/resetting-operations must be returned after
# state transition completed.
#
# "sync_transition" will set synchronous transition flags to all other
# synchronous transition flags sync_activation/deactivation/reset.
#
# exec_cxt.sync_transition: YES
# exec_cxt.sync_activation: YES
# exec_cxt.sync_deactivation: YES
# exec_cxt.sync_reset: YES

#
# Timeout of synchronous state transition [s]
#
# Default: 0.5 [s]
#
# When synchronous transition flags are set to YES, the following
# timeout settings are valid. If "transition_timeout" is set, the
# value will be set to all other timeout of activation/deactivation
# and resetting
#
# exec_cxt.transition_timeout: 0.5
# exec_cxt.activation_timeout: 0.5
# exec_cxt.deactivation_timeout: 0.5
# exec_cxt.reset_timeout: 0.5

#
# Specifying Execution Contexts
#
# Default: No default
#
# execution_contexts: None or <EC0>,<EC1>,...
# <EC?>: ECtype(ECname)
#
# RTC can be attached with zero or more Execution
# Contexts. "execution_contexts" option specifies RTC-specific
# attached ECs and its name. If the option is not specified, the
# internal global options or rtc.conf options related to EC will be
# used. If None is specified, no EC will be created.
#
# Availabilities in OpenRTM-aist
#
# - ExtTrigExecutionContext: External triggered EC. It is embedded in
#                            OpenRTM library.
# - OpenHRPExecutionContext: External triggred paralell execution
#                            EC. It is embedded in OpenRTM
#                            library. This is usually used with
#                            OpenHRP3.
# - RTPreemptEC:             Real-time execution context for Linux
#                            RT-preemptive pathed kernel.
# - ArtExecutionContext:     Real-time execution context for ARTLinux
#                            (http://sourceforge.net/projects/art-linux/)
#
# execution_contexts: PeriodicExecutionContext(pec1000Hz), \
#                     PeriodicExecutionContext(pec500Hz)

#
# EC specific configurations
#
# Default: No default
#
# Each EC can have its own configuration. Individual configuration can
# be specified by using EC type name or EC instance name. Attached ECs
# would be specified in execution_context option like <EC type
# name>(<EC instance name>), ...  EC specific option can be specified
```

```
# as follows.
#
# ec.<EC type name>.<option>
# ec.<EC instance name>.<option>
#
# Example:
# ec.PeriodicExecutionContext.sync_transition: NO
# ec.pec1000Hz.rate: 1000
# ec.pec1000Hz.synch_transition: YES
# ec.pec1000Hz.transition_timeout: 0.5
# ec.pec500Hz.rate: 500
# ec.pec500Hz.synch_activation: YES
# ec.pec500Hz.synch_deactivation: NO
# ec.pec500Hz.synch_reset: YES
# ec.pec500Hz.activation_timeout: 0.5
# ec.pec500Hz.reset_timeout: 0.5
```

Related issues:

Related to OpenRTM-aist (Java) - □□ #2350: RTObject_impl::initialize() □□EC□□...	□□	**02/04/2012**

Associated revisions

Revision 2302 - 02/04/2012 03:57 AM - n-ando

[incompat,header/imple,func] The method to create EC has been changed. Now zero or more EC can be attached to RTC. Configuration options for EC have been changed changed. refs #2349

Revision 2302 - 02/04/2012 03:57 AM - n-ando

[incompat,header/imple,func] The method to create EC has been changed. Now zero or more EC can be attached to RTC. Configuration options for EC have been changed changed. refs #2349

Revision 2302 - 02/04/2012 03:57 AM - n-ando

[incompat,header/imple,func] The method to create EC has been changed. Now zero or more EC can be attached to RTC. Configuration options for EC have been changed changed. refs #2349

History

#1 - 02/04/2012 03:58 AM - n-ando

- Status changed from □□ to □□

- % Done changed from 0 to 100