OpenRTM-aist (Java) - 機能 #2350

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

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

ステータス:	終了	**開始日:**	2012/02/04
優先度:	通常	**期日:**	
担当者:	fsi-takahashi	**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

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
```

関連するチケット:

関連している OpenRTM-aist (C++) - 機能 #2349: RTObject_impl::init...	**終了**	**2012/02/04**

関係しているリビジョン

リビジョン 674 - 2012/02/16 10:10 - fsi-katami

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 #2350

リビジョン 675 - 2012/02/24 18:06 - kurihara

refs #2350

履歴

#1 - 2012/02/16 10:13 - 匿名ユーザー

- ステータス を 新規 から 担当 に変更

- 進捗率 を 0 から 30 に変更

getInheritedECOptions()、getPrivateContextOptions()、getGlobalContextOptions()、getContextOptions()、findExistingEC()、createContexts()
の実装

#2 - 2012/02/27 10:30 - fsi-takahashi

- 担当者 を 匿名ユーザー から fsi-takahashi に変更

#3 - 2012/02/27 10:30 - fsi-takahashi

- ステータス を 担当 から 解決 に変更

#4 - 2012/02/27 10:30 - fsi-takahashi

- 進捗率 を 30 から 100 に変更

#5 - 2012/02/27 10:34 - fsi-takahashi

initialize()の修正完了

#6 - 2012/03/30 18:32 - n-ando

- ステータス を 解決 から 終了 に変更