

OpenRTM-aist (Python) - 整備 #2323

ExecutionContextBaseの見直し

2012/01/11 02:00 - n-ando

ステータス:	終了	開始日:	2012/01/11
優先度:	通常	期日:	
担当者:	kurihara	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 ExecutionContextBaseとこれを継承しているPeriodicExecutionContextなどの見直しを行う。 1. ECをGlobalFactoryで生成するように変更する。(C++, r2268) 2. ExecutionContextBaseはExecutionContextProfileをメンバとして持ち、すべての関数をdelegateする。 3. ECのプロファイルを管理するRTC_impl::ExecutionContextProfileクラス(RTC::ExecutionContextProfile)の導入 (C++, r2268) 4. 実装クラス内でRTC_impl::ExecutionContextProfileを利用するように変更する。(C++, r2268) 5. ExecutionContextBaseの親クラス ExtTrigEC, RefCountServantBaseを実装クラス(PeriodicExecutionContext)へ移動。(C++, r2268) 6. RTCのStateMachine管理クラスを導入し、ExecutionContextBaseにメンバとして持たせる。 7. 各実装クラスで上記のSM管理クラスを利用するように変更。			
関連するチケット:			
関連している OpenRTM-aist (Python) - 機能 #2384: ExecutionContext...		終了	2012/03/12
関連している OpenRTM-aist (Python) - 機能 #2343: RTObjectStateMac...		終了	2012/02/04
関連している OpenRTM-aist (Python) - 機能 #2348: ExecutionContext...		終了	2012/02/04

関係しているリビジョン

リビジョン 522 - 2012/03/12 21:37 - kurihara

[incompatible] ExecutionContextBase's super classes have been moved to PeriodicExecutionContext class. Delegation to ExecutionContextProfile has been added. refs #2323

[incompatible behavior] Now onExecute and onStateUpdate execution behavior has been changed when two or more RTCs attached to the EC.

Old: A::onExecute, A::onStateUpdate, B::onExecute, B::onStateUpdate

New: A::onExecute, B::onExecute, A::onStateUpdate, B::onStateUpdate

リビジョン 527 - 2012/03/13 16:43 - kurihara

[incompatible] ExecutionContextBase's super classes have been moved to PeriodicExecutionContext class. refs #2323

[incomaptible API] Now ExecutionContextProfile management has been moved to ExecutionContextBase. getObjRef() is also moved to ExecutionContextBase.

[incompatible behavior] Now onExecute and onStateUpdate execution behavior has been changed when two or more RTCs attached to the EC. refs #2320

Old: A::onExecute, A::onStateUpdate, B::onExecute, B::onStateUpdate

New: A::onExecute, B::onExecute, A::onStateUpdate, B::onStateUpdate

リビジョン 530 - 2012/03/13 17:14 - kurihara

Moved initialization processing in onStart(). refs #2323

リビジョン 531 - 2012/03/13 17:18 - kurihara

ExecutionContextProfile,RTObjectStateMachine and ExecutionContextWorker have been added. refs #2384 #2343 #2348 #2323

リビジョン 538 - 2012/03/14 16:55 - kurihara

- DEEFAULT_PERIOD variable has been added.
- Log messages have been corrected. refs #2323

リビジョン 540 - 2012/03/14 17:39 - kurihara

ExtTrigExecutionContext and OpenHRPExecutionContext have been updated. refs #2323

履歴

#1 - 2012/02/03 11:41 - kurihara

- ステータスを新規から担当に変更

#2 - 2012/02/06 10:36 - kurihara

- 担当者を n-ando から kurihara に変更
- 進捗率を 0 から 50 に変更

#3 - 2012/02/07 11:31 - n-ando

ExecutionContextBaseの実装について

ExecutionContextBase はテンプレートメソッドパターンで実装する。

- あるpublic関数 XXX に対して、実際に操作を行う関数 XXX を内部で呼び出す
- xxx() を呼び出す前に onXxing()関数を呼び出す
- xxx() を呼び出したあとに onXxed() 関数を呼び出す。
- activate/deactivate/resetComponent() 関数については、状態遷移待ちを行いその際に onWaitingXxed()関数を呼び出す。

ExecutionContextBaseでのECオプションの扱いについて

ExecutionContextBase::init() 関数でオプションを取得するようにする。

取得するオプションは以下のとおり。

- rate: 実行レート
- sync_transition: 同期遷移
- sync_activation: 同期アクティブ化
- sync_deactivation: 同期非アクティブ化
- sync_reset: 同期リセット
- transition_timeout: 同期遷移時のタイムアウト
- activation_timeout: 同期アクティブ化時のタイムアウト
- deactivation_timeout: 同期非アクティブ化時のタイムアウト
- reset_timeout: 同期リセット時のタイムアウト

EC実装について

従来通り PeriodicExecutionContext, ExtTrigExecutionContext, OpenHRPExecutionContext を ExecutionContextBase を使用して実装すること。

#4 - 2012/03/12 21:39 - kurihara

- 進捗率 を 50 から 80 に変更

[r522](#) にてExecutionContextBaseの修正とtest_ExecutionContextBase.py(ユニットテスト)を追加。

#5 - 2012/03/14 17:42 - kurihara

- ステータス を 担当 から 解決 に変更

- 進捗率 を 80 から 100 に変更

ExtTrigExecutionContext, OpenHRPExecutionContext を ExecutionContextBase を使用して実装。 [r540](#)

#6 - 2012/03/30 17:41 - n-ando

- ステータス を 解決 から 終了 に変更