OpenRTM-aist (Python) - 機能 #3406

⑦　トピックベースのポート接続機能

2015/12/22 09:44 - n-ando

ステータス:	終了	**開始日:**	2015/12/22
優先度:	通常	**期日:**	2016/03/25
担当者:	miyamoto	**進捗率:**	100%
カテゴリ:		**予定工数:**	30.00時間
対象バージョン:	RELEASE_1_2_0		

説明

DDSのようなトピックベースのポート接続機能を実装すること。ただし、この機能はデータポート、サービスポートにも利用できるように実装すること。

関係しているリビジョン

リビジョン 636 - 2016/02/01 13:37 - miyamoto

[incompat,new_func,->RELENG_1_2] Topic-based connection has been implemente. refs #3406

リビジョン 638 - 2016/02/01 14:05 - miyamoto

[incompat,new_func,->RELENG_1_2] Now NamingManager can register a reference of Port to naming service. refs #3406

リビジョン 648 - 2016/02/01 19:47 - miyamoto

[incompat,new_func,->RELENG_1_2] Topic-based connection has been implemented. refs #3406

履歴

#1 - 2016/01/14 16:18 - miyamoto

- 期日 を 2016/03/25 にセット

- 担当者 を miyamoto 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 90 に変更

- 予定工数 を 30.00時間 にセット

#2 - 2016/01/14 20:45 - miyamoto

- ファイル test_Manager_new.py を追加

- 進捗率 を 90 から 50 に変更

実装

Managerクラスに以下の関数を追加した。

publishPorts関数
指定のRTCが保持するポートをネーミングサービスへバインドする
subscribePorts関数
指定のRTCが保持するポートを同じトピック名のコンテキストの下にバインドしてあるポートオブジェクトと接続する
getPortsOnNameServers関数
与えられたパス以下の指定のkindのポートオブジェクトのリストを取得する
connectDataPorts関数
対象のデータポートと指定したリスト内のデータポートを接続する
connectServicePorts関数
対象のサービスポートと指定したリスト内のサービスポートを接続する

NamingManagerクラスに以下の関数、クラスを追加した。

bindPortObject関数
ポートを指定したパスでネーミングサービスへバインドする
registerPortName関数
対象のポートと名前をポート管理用オブジェクト(Port)に格納し、管理用オブジェクトをリストに追加する
unregisterPortName関数
指定の名前のポートをリストから削除する

Portクラス
ポート管理用オブジェクト

C++版でNamesがNameServerに変更されていたため、こちらでも変更を行った。
またコンストラクタ、unbindAll関数にもポートの管理用リストに関するコードを追加した。

NamingOnCorbaクラスに以下の関数を追加した。

bindPortObject
ポートを指定したパスでネーミングサービスへバインドする

CorbaNamingクラスに以下の関数を追加した。

listByKind関数
与えられたパス以下で指定のkindのオブジェクトを検索する
listBinding関数
与えられたパス以下のオブジェクトを取得する

Managerクラスのregister Component関数内で登録したRTCのポートをpublishPorts関数でネームサービスにバインドして、subscribePorts関数で既にバインド済みのポートと接続する。

ポートがバインドされるパスはトピック名をtopic_test、ポート名をport0とした場合、以下のようになる。

インポート
dataports.port_cxt/topic_test.topic_cxt/port0.inport
アウトポート
dataports.port_cxt/topic_test.topic_cxt/port0.outport
サービスポート
svcports.port_cxt/topic_test.topic_cxt/port0.svc

トピック名はポートのpublish_topicというプロパティに設定する。

テスト

添付のテスト用コードでテストを行った。
テスト用コードは以下の動作を行う。

setUp関数

マネージャ初期化
self.manager = OpenRTM_aist.Manager.init(sys.argv)
self.manager.setModuleInitProc(MyModuleInit)
self.manager.activateManager()

ここでTestComp10、TestComp20の2つのRTCを起動する。
TestComp10はアウトポートtopic_outとサービスポートtopic_serviceを保持しており、TestComp20はインポートtopic_inとサービスポートtopic_serviceを保持している。
つまり互いのデータポートとサービスポートが接続されていれば正常に動作しているという事になる。

test_Topic関数

データポートが接続されたかの確認
ans = OpenRTM_aist.already_connected(inport, outport)
self.assertTrue(ans)

サービスポートが接続されたかの確認
ans = OpenRTM_aist.already_connected(provided, required)
self.assertTrue(ans)

#3 - 2016/03/17 11:00 - miyamoto

- 進捗率 を 50 から 100 に変更

#4 - 2017/08/30 14:19 - n-ando

- ステータス を 新規 から 終了 に変更

ファイル

test_Manager_new.py	7.38 KB	2016/01/14	miyamoto