

Robotics Service Bus - Feature #1736

Feature # 44 (Resolved): Implement basic introspection support

Implement participant creation and destruction hooks

02/06/2014 06:28 PM - J. Moringen

Status:	Resolved	Start date:	03/24/2014
Priority:	Normal	Due date:	03/24/2014
Assignee:	J. Moringen	% Done:	100%
Category:	C++	Estimated time:	0.00 hour
Target version:	rsb-0.11		
Description			

Associated revisions

Revision aec45970 - 09/03/2014 02:14 PM - J. Moringen

Enable access to factory while loading plugins in `src/rsb/Factory.cpp`

refs #1736

Previously, plugin init functions could access the factory because of locking and initialization issues. This is solved by bypassing the locking and initialization logic during plugin loading.

- `src/rsb/Factory.cpp` (`factoryWhileLoadingPlugins`): new variable; stores a pointer to the factory while it loads plugins
(`getFactory`): use value of `factoryWhileLoadingPlugins` if it is not NULL
(`Factory::Factory`): set `factoryWhileLoadingPlugins` while loading plugins

Revision 15863313 - 09/03/2014 02:14 PM - J. Moringen

Added `signalParticipant{Created,Destroyed}` in `src/rsb/{Participant,Factory}.cpp`

fixes #1736

- `src/rsb/Participant.{h,cpp}` (header): updated copyright
(`SignalParticipantDestroyed`): new type; signal for participant creation
(`Participant::Impl::signalParticipantDestroyed`): new member variable; reference to factory's signal object
(`Participant::~~Participant`): emit participant destruction signal
(`Participant::setSignalParticipantDestroyed`): new method; used to install factory's signal object in the participant
- `src/rsb/Factory.{h,cpp}` (`SignalParticipantCreated`): new type; signal for participant creation
(`Factory::getSignalParticipantCreated`): new method; return signal for participant creation
(`Factory::getSignalParticipantDestroyed`): new method; return signal for participant destruction
(`Factory::createInformer`): install participant destroyed signal object

- in created participant; emit participant created signal
- (Factory::createInformerBase): likewise
- (Factory::createListener): likewise
- (Factory::createReader): likewise
- (Factory::createLocalServer): likewise
- (Factory::createRemoteServer): likewise
- src/rsb/patterns/LocalServer.cpp (LocalServer::registerMethod):
likewise
- src/rsb/patterns/RemoteServer.cpp (RemoteServer::getMethod): likewise

Revision 5faa3e16 - 09/05/2014 01:01 AM - J. Moringen

Enable access to factory while loading plugins in src/rsb/Factory.cpp

refs #1736

Previously, plugin init functions could not access the factory because of locking and initialization issues. This is solved by bypassing the locking and initialization logic during plugin loading.

- src/rsb/Factory.cpp (factoryWhileLoadingPlugins): new variable; stores a pointer to the factory while it loads plugins
- (getFactory): use value of factoryWhileLoadingPlugins if it is not NULL
- (Factory::Factory): set factoryWhileLoadingPlugins while loading plugins

Reviewed-by: Johannes Wienke <jwienke@techfak.uni-bielefeld.de>

Revision 4ac487bc - 09/05/2014 01:28 AM - J. Moringen

Added signalParticipant{Created,Destroyed} in src/rsb/{Participant,Factory}.cpp

fixes #1736

Also add Factory::create{Local,Remote}Method methods to enable a coherent implementation of the new signals in which basically only the Factory deals with the signals.

- src/rsb/Participant.{h,cpp} (header): updated copyright
- (SignalParticipantDestroyed): new type; signal for participant creation
- (Participant::Impl::signalParticipantDestroyed): new member variable; reference to factory's signal object
- (Participant::~~Participant): emit participant destruction signal
- (Participant::setSignalParticipantDestroyed): new method; used to install factory's signal object in the participant
- src/rsb/Factory.{h,cpp} (header): updated copyright
- (SignalParticipantCreated): new type; signal for participant creation
- (Factory::getSignalParticipantCreated): new method; return signal for participant creation

- (Factory::getSignalParticipantDestroyed): new method; return signal for participant destruction
- (Factory::createInformer): install participant destroyed signal object in created participant; emit participant created signal
- (Factory::createInformerBase): likewise
- (Factory::createListener): likewise
- (Factory::createReader): likewise
- (Factory::createLocalMethod): new method; creates LocalMethod instances; takes care of creation and destruction signals
- (Factory::createLocalServer): take care of creation and destruction signals as described above
- (Factory::createRemoteMethod): new method; creates RemoteMethod instances; takes care of creation and destruction signals
- (Factory::createRemoteServer): take care of creation and destruction signals as described above
- src/rsb/patterns/LocalServer.cpp (LocalServer::registerMethod): use new Factory::createLocalMethod method
- src/rsb/patterns/RemoteServer.cpp (RemoteServer::getMethod): use new Factory::createRemoteMethod method

Reviewed-by: Johannes Wienke <jwienke@techfak.uni-bielefeld.de>

History

#1 - 03/18/2014 08:10 PM - J. Moringen

- % Done changed from 30 to 80

#2 - 03/23/2014 08:49 PM - J. Moringen

- Assignee set to J. Moringen

#3 - 03/27/2014 06:53 PM - J. Moringen

- % Done changed from 80 to 90

#4 - 09/03/2014 06:39 PM - J. Moringen

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

Applied in changeset commit:rsb-cpp|15863313bc326264b50e55bc31e195774e98ba90.