Robotics Service Bus - Feature #517
Necessity of Factory?

08/20/2011 02:17 PM - J. Moringen

Status:	Resolved	**Start date:**	08/20/2011
Priority:	Normal	**Due date:**	
Assignee:	J. Moringen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Originally reported by Robert Haschke:

> *looking at the RSB examples, I'm wondering, why it is necessary to explicitly instantiate and use the rsb::Factory? I guess, that's a singleton. As a user I find it quite annoying to write two code lines to instantiate a listener or informer somewhere. Hence a appreciate the introduction of convenience methods which do this, e.g.:*
>
> *> template <class T>*
> *> Informer<T>::Ptr createInformer (...) {*
> *> return rsb::Factory::getInstance().createInformer<T> (...);*
> *> }*
> *>*
>
> *What do you think?*

Associated revisions

Revision 46f828c8 - 09/02/2011 09:14 PM - J. Moringen

Added convenience functions in src/rsb/CreateFunctions.{h,cpp}

fixes #517

- src/CMakeLists.txt: added files src/rsb/CreateFunctions.{h,cpp}
- src/rsb/CreateFunctions.{h,cpp}: new files; contain convenience functions for creating participants like listeners and informers without using the RSB factory explicitly

History

#1 - 08/24/2011 07:46 PM - J. Wienke

This increases the coupling and reduces the possibility of producing testable code. If you have a factory instance, that can be passed around (which means preventing to use getInstance all the time), you can configure your user objects with a different factory instance for unit testing. With direct function calls this is not possible.

#2 - 09/02/2011 08:41 PM - J. Moringen

- Status changed from New to In Progress

- Assignee set to J. Moringen

- % Done changed from 0 to 30

We decided to address the testability problem as follows:

1. Initially, the convenience functions are implemented as proposed by Robert
2. A Factory interface is extracted to allow for alternative factory implementations
3. The global Factory instance is made exchangable via Singleton<Factory>::setInstance() or something similar

4. Using thread-local storage or similar mechanisms, Factory instances are made **locally** exchangable, i.e.

```
    {
  WithScopedFactory bla(new MyMockFactory());

  // do something with Factory::getInstance()
}
```

#3 - 09/02/2011 09:16 PM - J. Moringen

- Status changed from In Progress to Resolved

- % Done changed from 30 to 100

Applied in changeset r2515.