Robotics Service Bus - Enhancement #469
Add Future-based Interface to Request/Reply Subsystem

08/03/2011 04:35 AM - J. Moringen

Status:	Resolved	**Start date:**	08/03/2011
Priority:	Normal	**Due date:**	
Assignee:	J. Moringen	**% Done:**	100%
Category:	Python	**Estimated time:**	0.00 hour
Target version:	0.4		

Description

Associated revisions

Revision e170dd13 - 08/07/2011 12:31 AM - J. Moringen

Added a simple future implementation in rsb/patterns/future.py

refs #469

- rsb/patterns/future.py: new file; contains a simple implementation of the future pattern; intended to be used in the request/reply implementation and interface

Revision d826488c - 08/07/2011 01:41 AM - J. Moringen

Improved nonsensical interface in rsb/patterns/future.py

refs #469

- rsb/patterns/future.py: removed getTimeout() method and added a timeout keyword parameter to get() method; improved documentation strings

Revision fcc3288f - 08/07/2011 01:57 AM - J. Moringen

Changed calls to remote methods to use futures in rsb/patterns/__init__.py

refs #469

- rsb/__init__.py: changed createServer to not accept a timeout parameter; this is handled using futures; documentation fixes
- rsb/patterns/__init__.py: removed Call class which basically was a Future; removed TimeoutError and RemoteExecutionError; changed RemoteServer constructor to not accept a timeout keyword parameter

Revision 14d5a895 - 08/07/2011 02:06 AM - J. Moringen

Adapted Python client to new future interface in integrationtest/python/client.py

refs #469

- python/client.py: added calls to Future.get() to all method calls to restore previous semantic

Revision fa935013 - 08/09/2011 04:11 AM - J. Moringen

Implemented sync/async, event/payload calls in patterns/__init__.py

fixes #469

- patterns/__init__.py: changed RemoteMethod.__call__ to call RemoteMethod.async and wait for the result; added method RemoteMethod.async which sends a request and returns a Future and DataFuture depending on whether the request is an Event or another

datum

- patterns/future.py: added class DataFuture which can be used to retrieve and return the payload of an Event in a future-based interface

History

#1 - 08/06/2011 10:54 PM - J. Moringen

- Status changed from New to In Progress

- Assignee changed from J. Wienke to J. Moringen

#2 - 08/07/2011 02:20 AM - J. Moringen

- Status changed from In Progress to Feedback

Should we add a mechanism for making blocking calls or should we just go with server.mymethod('foo').get()?

#3 - 08/07/2011 02:21 AM - J. Moringen

- % Done changed from 0 to 100

#4 - 08/09/2011 04:13 AM - J. Moringen

- Status changed from Feedback to Resolved

Applied in changeset r2311.