

Robotics Service Bus - Enhancement #468

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:	Common Lisp	Estimated time:	0.00 hour
Target version:	0.4		
Description			

Associated revisions

Revision f1429493 - 08/08/2011 03:24 AM - J. Moringen

Added future protocol, class in src/patterns/{protocol,future}.lisp

refs #468

- src/patterns/protocol.lisp (future-done?): new generic function; check status of future
(future-result): new generic function; obtain result of a future
(self future-result): new generic function; supply the result of a future
(self future-error): new generic function; mark a future as failed
- src/patterns/future.lisp: new file; contains `future' class and protocol implementation
- src/patterns/package.lisp (package rsb.patterns): added exported symbols future-done?, future-result, future-error and future
- test/patterns/future.lisp: new file; contains unit tests for the future protocol and `future' class
- test/patterns/package.lisp: new file; contains package definition for unit tests of the patterns module
- cl-rsb.asd (system cl-rsb): added file src/patterns/future.lisp
(system cl-rsb-test): added module patterns with files test/patterns/package.lisp and test/patterns/future.lisp

Revision 80df46d0 - 08/08/2011 06:29 AM - J. Moringen

Added cons-based fast-path future impl in src/patterns/future.lisp

refs #468

- src/patterns/future.lisp (future-result future): use `%dispatch-result'
(self future-error): use `%make-failure-value'
(future-done? cons): new method; for cons-based fast-path future
(future-result cons): likewise
(self future-result cons): likewise
(self future-error cons): likewise
(%dispatch-result): new function; shared between future implementations
(%make-failure-value): likewise

Revision cffc2d5f - 08/08/2011 06:29 AM - J. Moringen

Use future in remote-method in src/patterns/remote-server.lisp

refs #468

- src/patterns/protocol.lisp (call): added keyword parameter block?
- src/patterns/remote-server.lisp
 - (setf %method-listener :after t remote-method): use future interface for storing results
 - (call t remote-method event): accept keyword arguments; create a `future' instance to manage the result of the method call
 - (call :around t remote-method event): added block? keyword parameter; wait for result depending on it
 - (remote-server::timeout): removed slot; handled via `future' instances now
 - (call remote-server string t): accept and forward keyword arguments
 - (call t remote-method t): likewise
 - (%call-result->future): new function; helper for setting future result/error
- src/patterns/local-server.lisp (call t local-method event): accept keyword parameters

Revision b1e5ae14 - 08/09/2011 10:14 AM - J. Moringen

Added tests in test/patterns/integration.lisp

refs #468

- test/patterns/integration.lisp: new file; contains integration tests for the client and server components of the client/server communication pattern
- test/patterns/package.lisp (package rsb.patterns.test): added used package rsb
- cl-rsb.asd (system cl-rsb-test): added file test/patterns/integration.lisp

Revision f4a09a53 - 08/09/2011 10:14 AM - J. Moringen

Fixed handling of results in src/patterns/remote-server.lisp

fixes #468

- src/patterns/remote-server.lisp (call t remote-method event): pass request to event handler for more precise error; pass return to `%call-result->future'
- (call :around t remote-method event): added return keyword parameter
- (%call-result->future): added return parameter which controls whether the event or its payload is used result

History

#1 - 08/08/2011 02:00 AM - J. Moringen

- Status changed from *New* to *In Progress*

#2 - 08/08/2011 06:30 AM - J. Moringen

- % Done changed from 0 to 80

#3 - 08/09/2011 10:14 AM - J. Moringen

- Status changed from *In Progress* to *Resolved*

- % Done changed from 80 to 100

Applied in changeset r2323.