Robotics Service Bus - Enhancement #1876
Use service-provider system for participant classes

05/11/2014 04:30 PM - J. Moringen

Status:	Resolved	**Start date:**	05/11/2014
Priority:	Normal	**Due date:**	
Assignee:	J. Moringen	**% Done:**	100%
Category:	Common Lisp	**Estimated time:**	0.00 hour
Target version:	rsb-0.11		

Description

Related issues:

Related to Robotics Service Bus - Feature # 44: Implement basic introspection...	**Resolved**	**03/24/2014**	**03/24/2014**

Associated revisions

Revision f1ed8658 - 08/03/2014 10:48 AM - J. Moringen

Mention make-participant and underlying service in news.rst

refs #1876

- news.rst (RSB 0.11): mention make-participant and the underlying
  service-provider protocol

Revision 3405fbf8 - 08/13/2014 08:34 AM - J. Moringen

Added make-participant[-using-class] generic functions in src/protocol.lisp

refs #1876

Participants are now created by the two generic functions `make-participant' and `make-participant-using-class'. The former is intended to be called by clients of the protocol and calls latter with a prototype object of the requested participant class.

- src/protocol.lisp (toplevel): added documentation comment for
  `make-participant' protocol
  (make-participant): new generic function; create and return a
  participant given its class
  (make-participant-using-class): new generic function; create and
  return a participant given its class and a prototype instance of that
  class
  (make-participant-using-class class t scope): new method; default
  behavior consists in just calling `make-instance'
- src/mixins.lisp
  (make-participant-using-class :around class error-hook-mixin scope):
  new method; connect supplied error-policy to error-hook of the created
  participant instance
- src/participant.lisp (make-participant t scope): changed function into
  method; locate the requested class and call
  `make-participant-using-class'
  (make-participant-using-class :around class client scope): new method;

take care of default transports and converters keyword arguments
(make-participant-using-class class client scope): new method; do what
the `make-participant' function did previously
(make-participant t uri): new method; split the URI into a scope and
options and call `make-participant' again
(make-participant t string): new method; parse the supplied string as
URI or scope and call `make-participant' again
(make-participant :around t scope): new method; establish restarts for
retrying and supplying a different scope
(make-participant :around t uri): new method; establish restarts for
retrying and supplying a different URI
(make-participant :around t t): new method; translate error conditions
into `participant-creation-error' conditions

- src/reader.lisp (make-reader scope): call `make-participant' instead
  of creating the instance directly
- src/listener.lisp (make-listener scope): likewise
- src/informer.lisp (make-participant-using-class class informer scope):
  new method; connect processor of configurator to informer
  (make-informer scope): call `make-participant' instead of creating the
  instance directly; no longer connect processor to informer as the new
  `make-participant-using-class' method does that
- src/package.lisp (package rsb): added exported symbol
  make-participant-using-class
- src/patterns/request-reply/server.lisp
  (make-participant-using-class class server scope): new method; call
  next method with adjusted transport options
- src/patterns/request-reply/local-server.lisp
  (setf server-method function local-server string): use
  `make-participant' instead of `make-instance'
  (make-local-server scope): likewise; error-hook plumbing is no longer
  necessary
- src/patterns/request-reply/remote-server.lisp
  (server-method remote-server string): use `make-participant' instead
  of `make-instance'
  (make-remote-server scope): likewise; error-hook plumbing is no longer
  necessary
- test/reader.lisp (define-basic-participant-test-cases reader): adapted
  to changed interface of `define-basic-participant-test-cases'
- test/listener.lisp (define-basic-participant-test-cases listener):
  likewise
- test/informer.lisp (define-basic-participant-test-cases informer):
  likewise
- test/package.lisp (define-basic-participant-test-cases): generate a
  second construction test which uses `make-participant' to construct
  the participant; accept distinct and common arguments for both
  construction modes

Revision f86bc9c5 - 08/13/2014 08:37 AM - J. Moringen

Added participant service in src/protocol.lisp

fixes #1876

Participant classes are registered with the new participant service and instantiated via the service by `make-participant'.

- src/protocol.lisp (toplevel): extended documentation comment for `make-participant' protocol
  (make-participant): changed parameter name class -> kind
  (participant-provider): new class; specializer provider class for the participant service
  (shared-initialize :after participant-provider t): new method; finalize the associated participant class
  (make-provider t participant-provider): new method; retrieve the prototype of the finalized participant class and call `make-participant-using-class'
  (define-service participant): new service; provided by participant classes
  (register-participant-class): new function; helper function for registering participant classes
- src/participant.lisp (make-participant t scope): call `make-provider' instead of preparing and calling `make-participant-using-class'
  (make-participant t uri): changed parameter name designator -> kind
- src/reader.lisp (register-participant-class reader): new toplevel form; register `reader' participant class
  (make-reader scope): call `make-participant' with designator instead of class name
- src/listener.lisp (register-participant-class listener): new toplevel form; register `listener' participant class
  (make-listener scope): call `make-participant' with designator instead of class name
- src/informer.lisp
  (register-participant-class informer): new toplevel form; register `informer' participant class
  (make-informer scope): call `make-participant' with designator instead of class name
- src/patterns/request-reply/local-server.lisp
  (register-participant-class local-method): new toplevel form; register `local-method' participant class
  (register-participant-class local-server): similar
  (setf server-method function local-server string): call `make-participant' with designator instead of class name
  (make-local-server scope): call `make-participant' with designator instead of class name
- src/patterns/request-reply/remote-server.lisp
  (register-participant-class remote-method): new toplevel form; register `remote-method' participant class
  (register-participant-class remote-server): similar
  (server-method remote-server string): call `make-participant' with designator instead of class name
  (make-remote-server scope): call `make-participant' with designator instead of class name
- test/package.lisp (define-basic-participant-test-cases): call `make-participant' with kind instead of class name
- cl-rsb.asd (system cl-rsb): added system dependency on architecture.service-provider

Revision ebcf53a5 - 08/13/2014 09:12 AM - J. Moringen

Added option to list participants in info/main.lisp

refs #1876

- info/main.lisp (header): updated copyright
  (update-synopsis): added --participants commandline option
  (first-line-or-less): new function; helper function for displaying
  first line or less of documentation strings
  (main): retrieve value of participants commandline option and print
  participant class list accordingly

Revision 4967b8fd - 08/13/2014 09:36 AM - J. Moringen

Mention --participants commandline option in info.rst

refs #1876

- info.rst (Description): mention --participants commandline option

Revision a1ae8d9c - 08/13/2014 09:55 AM - J. Moringen

Added option to list participants in info/main.lisp

refs #1876

- info/main.lisp (header): updated copyright
  (update-synopsis): added --participants commandline option
  (first-line-or-less): new function; helper function for displaying
  first line or less of documentation strings
  (main): retrieve value of participants commandline option and print
  participant class list accordingly
- CMakeLists.txt: test --participants option of info binary

History

#1 - 05/11/2014 04:30 PM - J. Moringen

- Tracker changed from Bug to Enhancement

#2 - 08/13/2014 09:20 AM - J. Moringen

- Status changed from In Progress to Resolved
- % Done changed from 0 to 100

Applied in changeset commit:rsb-cl|f86bc9c5d5a51a46656546e8762b55045b2623d3.

#3 - 08/13/2014 09:28 AM - J. Moringen

- Related to Feature #44: Implement basic introspection support added