

Robotics Service Bus - Bug #515

Data handlers cannot deal with unexpected types

08/19/2011 11:29 PM - J. Moringen

Status:	New	Start date:	08/19/2011
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Specification	Estimated time:	0.00 hour
Target version:	rsb-0.18		

Description

See also: the first topic of [\[\[Meetings2011-08-02\]\]](#).

Problem Statement

Associated handlers of listeners basically have to expect events with payloads of any type (unless a type filter is installed on the listener). This can be a problem for handler classes like `DataHandler` or `DataFunctionHandler` in C++ that can only handle events with certain payload type, because the behavior for events with incompatible payload types has not been specified yet.

Problems of the current implementation (restricted to C++ for clarity here):

- Clients may get the impression that handlers like `DataHandler` cause the associated listener to only receive events that have compatible payload types
- This is currently not true; Instead, the program will most likely crashes due to an invalid `static_pointer_cast`

Potential Solutions

Signal Errors for Incompatible Events

Classes like `DataHandler` could signal an error when encountering incompatible payload types.

Variations include

- Aborting the program
- Displaying an error message but continuing
- Configurable behavior

Drop Incompatible Events

`DataHandler` and friends could check the payload types of events being dispatched to them and simply ignore events with incompatible payload types.

This would lead to inefficient processing if a single listener with multiple handlers for different payload types was used, since the dispatching mechanism would still dispatch all events to all handlers which would then drop unsuitable events only after most of the processing already happened.

General Considerations

Both previously mentioned solutions can be implemented in several different ways:

- Introduce a common base class for `DataHandler` and friends that implements the chosen behavior (error or drop) in a processable method:
 - processable would be called to determine whether an event should be dispatched to a handler
 - If processable returned true, the event would be dispatched in the usual way by calling `handle`
- Integrate feedback from handlers into the dispatch logic
 - Dispatching could display warnings or signal errors if events are dropped because they are not handled by any handler
 - This could be achieved by adding a return value to `handle` or the above mentioned processable method
 - The downside of a return value for `handle` would be potential confusion of clients implementing the Handler interface
- It may be necessary for efficiency reasons to communicate information regarding the desired payload types of registered handlers all the way back to the transport layer

Related issues:

Related to Robotics Service Bus - Bug # 1153: unknown data types cause crash ...	Rejected	08/29/2012
Related to Robotics Service Bus - Enhancement # 1515: Add data type to all pa...	New	05/27/2013
Blocks Compliant Control Architecture - Bug # 973: Type filter for input ports	New	03/28/2012

Associated revisions

Revision 568192cc - 07/09/2012 06:02 PM - J. Moringen

Added FilteringHandler in src/rsb/Handler.h

refs #515

- src/rsb/Handler.h: added class FilteringHandler

History

#1 - 10/21/2011 05:04 PM - S. Wrede

- Target version set to rsb-0.7

#2 - 10/22/2011 12:27 AM - R. Haschke

As third method to handle the problem, we also discussed with Jan to install a filter, which ignores non-matching events as early as possible. This would require a type-based routing of incoming events to appropriate listeners.

#3 - 06/18/2012 03:55 PM - J. Moringen

- Target version changed from rsb-0.7 to rsb-0.9

#4 - 04/17/2013 04:11 PM - J. Moringen

- Target version changed from rsb-0.9 to rsb-0.10

#5 - 12/10/2013 11:47 PM - J. Moringen

- Target version changed from rsb-0.10 to rsb-0.11

#6 - 09/26/2014 01:57 PM - R. Haschke

With commit:5ee2b2c66 Jan implemented the required TypeFilter. In combination with FilteringHandler(TypeFilter, DataHandler/DataFunctionHandler) one can now ensure type safety.

If you don't vote for instantiating the TypeFilter by default as soon as one uses the DataHandler/DataFunctionHandler (which of course wouldn't allow to handle differently typed events on a single Listener), you should add some docs in DataHandler+DataFunctionHandler pointing the programmer to the potential issue and its TypeFilter solution.

#7 - 12/17/2014 10:45 AM - J. Wienke

- Target version changed from rsb-0.11 to rsb-0.12

#8 - 04/27/2015 11:14 AM - J. Wienke

- Target version changed from rsb-0.12 to rsb-0.13

#9 - 03/02/2016 08:55 AM - J. Moringen

- Target version changed from rsb-0.13 to rsb-0.14

#10 - 05/13/2016 01:58 PM - J. Moringen

- Target version changed from rsb-0.14 to rsb-0.15

#11 - 07/26/2016 11:36 AM - J. Moringen

- Target version changed from rsb-0.15 to rsb-0.16

#12 - 04/10/2017 08:58 PM - J. Moringen

- Target version changed from rsb-0.16 to rsb-0.17

#13 - 10/10/2017 06:21 PM - J. Moringen

- *Target version changed from rsb-0.17 to rsb-0.18*