

Robotics Service Bus - Tasks #223

Tasks # 240 (Resolved): Refactor C++ Implementation to 2011-04-13 Domain Model

Model implementation of Methods according to the domain model

03/02/2011 12:06 AM - S. Wrede

Status:	Resolved	Start date:	03/02/2011
Priority:	Normal	Due date:	
Assignee:	J. Wienke	% Done:	100%
Category:	C++	Estimated time:	0.00 hour
Target version:			
Description			
Do we want to support action attributes (INSERT, UPDATE, QUERY, REQUEST, REPLY, ...) as a general feature of RSB events or do we decide that this is domain specific data?			

Associated revisions

Revision 0a180a84 - 05/06/2011 03:33 PM - J. Wienke

prepare to have a method for events.

refs #223

Revision cca59169 - 05/06/2011 03:34 PM - J. Wienke

prepare to have a method for events.

refs #223

Revision 3fd0a88f - 05/08/2011 01:15 PM - J. Wienke

matching of methods

fixes #223

History

#1 - 03/02/2011 12:13 AM - J. Wienke

This implies that there is some kind of state, which isn't- So I don't think that it is a good idea. The only thing that could also have a meaning without state would be UPDATE as a refinement of an older message. But I got the suspicion that this leads to a general misuse of this pattern and is used then more often than generally useful. So better not ;)

#2 - 03/02/2011 12:24 AM - J. Otto

But with action attributes a component is able to hold his own "state" (;

#3 - 03/02/2011 12:25 AM - J. Wienke

Jens Otto wrote:

| *But with action attributes a component is able to hold his own "state" (;*

What do you mean?

#4 - 03/02/2011 12:39 AM - S. Wrede

Well, I am not entirely convinced by Johannes initial argument. Sure, some (most) of the RSB interactions don't feature state. However, the applications using RSB do so. Why not help developers by adding this kind of information and accompanying filters to the feature set of the framework. In my opinion the alternative for developers is to (a) explicitly represent this either in user-defined event types or (b) in the content of events. (a) may lead to a bloated type hierarchy and (b) to performance penalties as you are parsing complex payload for metadata access. Furthermore, I suspect that we could use it also internally for the patterns that we want to offer as part of the basic functionality.

BTW: Ich geh' jetzt ins Bett und wir können das auch morgen direkt besprechen... ;-)

#5 - 04/29/2011 06:04 PM - J. Wienke

- *Subject changed from Discuss Action Types for Events to Model implementation of Methods according to the domain model*
- *Category changed from Protocol to C++*
- *Assignee deleted (S. Wrede)*

We have decided on a version how to perform this. This is described in the domain model. Now a model for the implementation is required.

#6 - 05/03/2011 02:15 PM - J. Wienke

- *Parent task set to #240*

#7 - 05/06/2011 03:33 PM - J. Wienke

- *Status changed from New to In Progress*
- *Assignee set to J. Wienke*

#8 - 05/11/2011 02:28 PM - J. Wienke

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

Applied in changeset r830.