

## GROMACS - Task #2945

### Give MdModules access to simulation resources (e.g. atom selection manager or communication infrastructure)

05/08/2019 06:14 PM - Christian Blau

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	Christian Blau
<b>Category:</b>	mdrun
<b>Target version:</b>	2020
<b>Difficulty:</b>	hard
<b>Description</b>	
MdModules depend on resources that are not yet available when they are build.	
The goal of this task is to build an infrastructure that	
- provides the MDModules with the resources they need (preferably without changes to the IMDModules interface and avoiding function bloat of MDModules)	
- ensures that resource dependencies are resolved in the right order (preferably at compile time)	
- callback when resources are not longer available	
<b>Related issues:</b>	
Related to GROMACS - Task #3040: Refactor Restraint module	<b>New</b>

#### Associated revisions

##### Revision 8b0ce9f5 - 08/08/2019 09:47 AM - Test User

Provide callbacks/notifications for MDModules

Adds functionality for MdModules to subscribe to be called back during the simulation. Within the run, subscribed modules are notified of events, that are distinguished by the function argument by the call back function.

Implements the callbacks based on storing function pointers, following the discussion in <https://gerrit.gromacs.org/c/gromacs/+/10942>

refs #2945

Change-Id: I61589215fa9beb79825f0b5261ed50b4116046ff

#### History

##### #1 - 05/19/2019 01:26 PM - Eric Irrgang

In the case of initializing the modules, this would seem to be a good case for a Builder pattern. Is there an argument against adding resources to module builders as Mdrunner::mdrunner progresses? Presumably, the builders themselves would also have to have "addSubscriber" aspects for inter-module dependencies.

But since Modules will need to consume some resources during simulation, likely repeatedly, it could be that both problems can be solved together with an Observer pattern such that resources passed during set up are just a special case of one-time updates. However, we then also need a Visitor pattern to allow Modules to subscribe themselves to resources, and possibly a Mediator to serve as the subject to which the modules are subscribed and which, in turn, is the subscriber to the various resources as they become available during Mdrunner::mdrunner.

##### #2 - 07/18/2019 04:29 PM - Eric Irrgang

- Related to Task #3040: Refactor Restraint module added

##### #3 - 12/27/2019 04:26 PM - Paul Bauer

- Status changed from New to Resolved

I guess this is resolved?

##### #4 - 12/28/2019 10:37 AM - Paul Bauer

- Status changed from Resolved to Closed

No comment means yes