The TaskMuster Behavior

Spring 2025

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139
`project-pavlab/bhvdocs/bhv_task_wpt`

Contents

1 The TaskMuster Behavior

The *TaskMuster* behavior is a behavior for participating in a decentralized auction among a group of vehicles, to determine which vehicle is most appropriate to transit to a given muster region. In this simple behavior the auction bit is solely based on linear distance to the center of the muster region. Most of the functionality of this behavior is comprised of the `IvPTaskBehavior` superclass.

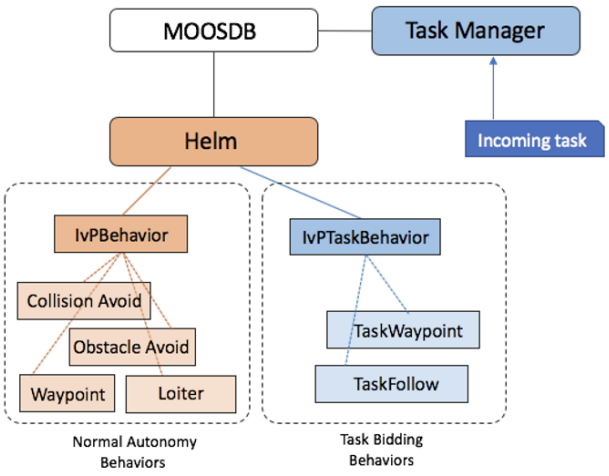

Figure 1: **Task Behaviors:** A task behavior are behaviors of the helm just like any other behavior. They all have the `IvPTaskBehavior` superclass to support the common components needed to participate in inter-vehicle auctions.

The primary function of a task behavior, like the `TaskMuster` behavior, is to participate in an inter-vehicle auction. The high-level view of an auction is shown in Figure **??** below.

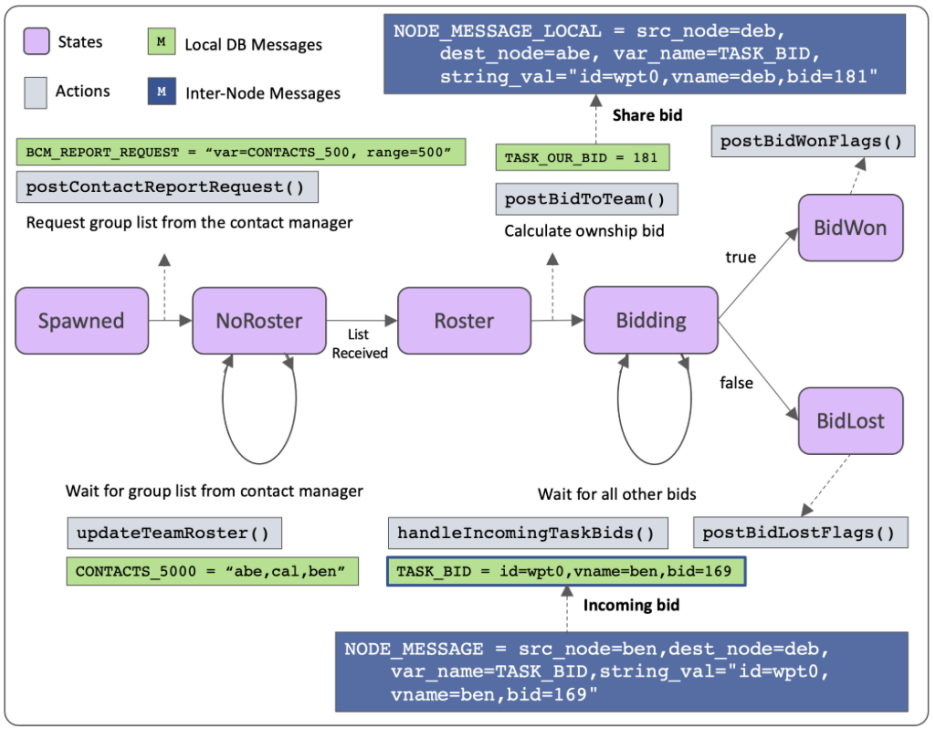

Figure 2: **Task Behavior Progression:** A task behavior proceeds through the states shown, beginning from initial spawning by the task manager, and generation of bids to collaborators. Depending on ownship bid compared to collaborators, the task behavior will generate further publications, i.e., flags, that will spawn the appropriate behavior to perform the task. The primary difference between different task behaviors is in the implementation of bid generation.

1.1 Configuration Parameters

Certain configuration parameters available to LegRun behavior are are unique to the LegRun behavior and others are inherited from the `IvPBehavior` superclass. Here parameters unique to the LegRun behavior are presented.

The set of configuration parameters is long, and split into two groups. The first group deals mainly with the shape of the leg, including position, angle, length and nature of the Williamson turn. The LegRun behavior maintains an instance of a LegRun C++ class, and these parameters are essentially passed directly to that class. The remainder of the configuration parameters relate to the operation of the behavior.

Listing 1.1: Configuration Parameters for the TaskBehavior Superclass.

Parameter	Description
id:	TBD
src:	TBD
hash:	TBD
time:	TBD

type:	TBD
exempt:	TBD
max_bid_wins:	If true, then highest bid wins.
time_range:	TBD
team_by_group:	TBD
bidwonflag:	Posted when/if the behavior wins its auction.
bidlostflag:	Posted when/if the behavior loses its auction.
xbidwonflag:	TBD
xbidlostflag:	TBD
bid_value_var:	TBD
task_state_var:	TBD

Listing 1.2: Configuration Parameters for TaskMuster Behavior.

Parameter	Description
waypt:	The x-y location of the waypoint.

Listing 1.3: Example Configuration Block.

```
      name = tuster_
    updates = TASK_MUSTER
 templating = spawn

         type = muster
 max_bid_wins = false

   team_range = 5000

  team_by_group = true

#include plugs.bhv <muster_regions>

  bidwonflag = UP_MUSTER  = muster_region=$[REGION]
  bidwonflag = MUSTER  = true
  bidwonflag = STATION = false
  bidwonflag = RETURN  = false
  bidwonflag = CONVOY  = false

  bidwonflag = LAST_TASK=$[LAST_TASK]
  //  xbidwonflag = MISSION_TASK = type=convoy,id=convoy$[ID+1], \
                   contact=$[OWNSHIP],exempt=$[OWNSHIP]
  xbidwonflag = MISSION_TASK = type=convoy,id=follow_$[OWNSHIP], \
                contact=$[OWNSHIP],exempt=$[OWNSHIP]
}
```

1.2 Publications

Listing 1.4: Configuration Parameters for the LegRun Behavior.

Variable	Description
TM_ALERT_REQUEST:	

1.3 Flags and Macros

The `TaskMuster` behavior supports the below set of event flags in addition to the standard behavior flags, e.g., endflags, runflags. These are:

- `bidwonflag`: Posted when/if the behavior wins its auction.
- `bidlostflag`: Posted when/if the behavior loses its auction.

The following macros are supported in the `TaskMuster` behavior. These macros will be expanded in any event flag, including event flags defined for all IvP behaviors as well as event flags defined only for the `TaskMuster` behavior.

- `$[PTX]`: The x location of the waypoint.
- `$[PTY]`: The y location of the waypoint.