# Development Notes

**August 2025**

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

## Overview

The below notes are changes applied to the git main branch after Release 24.8. They are sorted by date, with the most recent changes at the top.

### Mods Recap by Date

- 250811-MILD-GLEE uSimMarineV22 (Added ability to reset the nav solution to ground truth)
- 250811-MALT-RAKE IvPBehavior (Added support for reporting mode and submode)
- 250810-SOUR-JUDD BHV_Waypoint (Added calc dist to end of cycle or end all points)
- 250810-COLD-REED XYSegListr (Fix edge case returning centroid for segl size 1.)
- 250810-VERY-MENU BHV_Waypoint (Added policy for slowing upon arrival last point)
- 250615-HALS-LINK pMarineViewer (Added options for handling vehicle extrapolation)
- 250615-COOL-GULL pContactMgrV22 (Added ability to expunge contacts)
- 250615-DARN-DAVE pHelmIvP (Modified ABLE_FILTER handling reduced memory)
- 250615-FINE-FRED BHV_AvdColregsV22 (Reduced per-contact debugging verbosity)
- 250615-VERY-VOSS pHelmIvP (Fixed bug where stale contact behaviors not deleted)
- 250615-NEAT-PARK pHelmIvP (Added bhv status output for checking cmgr handling)
- 250417-FIVE-BANG pContactMgrV20 (Added ability to disable/re-enable alert IDs)
- 250415-JIMS-CARL pHelmIvP (BHV_AvoidObstacleV24 augmented to support vsource disabling)
- 250414-FOXY-WEBB pHelmIvP (Helm to disable/re-enable behaviors based on contact vsource)
- 250414-LIKE-PLUM pContactMgrV20 (Added ability to apply contact filter based on vsource)
- 250410-IDLE-IKER lib_contacts (Added vsource field optional part of node records)
- 250409-EASY-LOOK pMarinePIDV22 (Quieted DEPTH related subs in non-uuv missions)
- 250222-WISE-COLT BHV_AvoidObstacleV24 (Handle helm not config w/ platform model)
- 250218-THIN-SPAT pMarineViewer (Added SMR vessel type in viewer applications)
- 250216-MILD-OMNI pNodeReporter (Added static node report riders)
- 250215-DULL-VIEW pNodeReporter (Enabled dynamic node report riders)
- 250214-DULL-NICO lib_bhvutil (Bugfix in RefineryCPA for missions with non-zero min speed)
- 250127-POSH-LYLE uFldObstacleSim (Added request to pLogger to log obstacle file)
- 250126-WIDE-NOTE pMissionEval (Support for new macros, report formats, appcast output)
- 250125-DRAB-HACK pHelmIvP (Added check for initial nav solution before proceeding)
- 250113-ZERO-TONE lib_contacts (Added JSON serialization for NodeRecords)
- 250112-HALS-CLEM alogview (Added ability detach a numerical variable for plotting.)
- 250112-FULL-CARY alogsplit (Added ability to isolate numerical subcomponents of strings)
- 250114-BONY-ANTS pNodeReporter (Added option to publish JSON formatted reports)

- 250114-VAIN-FONT lib_mbutil (Added JsonUtils for converting to/from JSON to CSP strings)
- 250111-DARN-ROOM pContactMgrV20 (Added support for early warning time)
- 250108-WAVY-REID uSimMarineV22 (Added option to publish USM_NAVPOS_SUMMARY)
- 250103-SICK-CODI pContactMgrV20 (Early warning system fixes and rel speed change)
- 241229-SOFT-IVER uFldBeaconRangeSensor (Integrated ContactLedger into the app)
- 241229-WIDE-RITZ uFldContactRangeSensor (Integrated ContactLedger into the app)
- 241229-GRIM-KALE uFldCollObDetect (Integrated ContactLedger into the app)
- 241228-NUDE-GROG uFldObstacleSim (Integrated ContactLedger into to the simulator)
- 241228-FULL-DUKE uFldCollisionDetect (Integrated ContactLedger into the collision detector)
- 241222-MEGA-MOLD IvPBehavior (Modified interface for accessing contact info)
- 241222-SLIM-TOUR pHelmIvP (Integrated ContactLedger into the helm and behaviors)
- 241209-AVID-SONG pContactMgrV20 (Integrated the ContactLedger into the contact manager.)
- 241206-DANK-ACHE pMarineViewer (Integrated the ContactLedger into pMarineViewer)
- 241203-EASY-BAND lib_geodaid (Added lib_geodaid with ContactLedger util class)
- 241130-OPEN-ZALE pContactMgrV20 (Added support for event flags when contact ages out)
- 241130-COLD-COBY pContactMgrV20 (Added support for early warnings)
- 241129-NINE-ALAN pHelmIvP (Added support for disabling/enabling contacts)
- 241129-FLAT-LADY pContactMgrV20 (Added support for disabling/enabling contacts)
- 241127-DRAB-SLUM aloghelm (Added support for disabled behaviors)
- 241117-GOOD-SOUP IvPBehavior (Added new state, disabled, alongside idle, running etc)
- 241107-VAST-NOEL BHV_CutRange (Allow users to have patience param up to 100.)
- 241102-HAZY-KIAN mhash_gen (Added zhash capability for zbatch auto-testing)
- 241031-PINK-WINE pAutoPoke (New thin MOOS app for poking MOOSDB upon startup)
- 241031-HIGH-TOWN uMayFinish (New thin cmdline MOOS app for monitoring mission finish)
- 241020-DANK-GIBB IvPBehavior (Added getBufferIsKnown(string))
- 241020-MEEK-OVAL BHV_AvoidObstacleV24 (Additional macro support for obs id and min_util_cpa)
- 241020-MALE-KENT BHV_FixedTurn (Added alt subscription for DESIRED_RUDDERX)
- 241020-SOFT-ALEX uSimMarineV22 (mod uSimMarineV22 to halt on OBSTACLE_HIT=true)
- 241017-FLAT-MARK BehaviorSpec (Added optional spawning cap for templating behaviors)
- 241012-FAST-RIOT AOF_R16 (Option added to disallow any port turn during giveway)
- 241010-SAFE-CODE BHV_AvdColregsV22 (Additional status macros, quieter idle output)
- 241009-LONG-MELL uFldNodeComms (Integration of ContactLedger as key data structure)
- 241009-ZERO-LAND ContactLedger (Introduced ContactLedger class, NodeRecord mods)
- 241003-MEEK-RUST pHelmIvP (IvPHelm now handles node reports w/ only lat/lon)
- 241003-VERY-LADY lib_contacts (NodeRecord class serialize only global coords)
- 241005-DRAB-EMIL mhash_gen (mhash_gen –shore,-s produces short mhash)
- 241004-PINK-BRAM pMarineViewer (Fixed infocast pane browser reset upon refresh)
- 241003-LIVE-JAWS pNodeReporter (Enable node reports to be posted with just lat/lon)
- 241003-GRIM-WINS pMarineViewer (Handle node reports with only lat/lon and no x/y)
- 241001-HALF-PAWS pMarineViewer (Added Infocast Layout options esp for swarms)
- 240930-JUST-ZIGS pMarineViewer (Added Infocast panes darker colors)

### Modifications By Date

250811 uSimMarineV22 (augmentation) (MILD-GLEE)

Short: Added ability to reset the nav solution to ground truth

The simulator has the ability to maintain a "ground truth" nav solution along side the primary solution. When the simulator receives mail to USM_RESET_NAV the the primary solution is set to the ground truth solution.

---

250811 IvPBehavior (augmentation) (MALT-RAKE)

Short: Added support for reporting mode and submode

Some behaviors have an internal mode and/or submode. A virtual function was created at the IvPBehavior level to allow behaviors to be generally queried for this information by the helm when constructing an IVPHELM_SUMMARY helm report. If behaviors have no notion of this, then empty strings are fine. The helm report is ingested in alogview in the HelmPlot pop-up window, and this new information provides great debugging feedback when replaying in alogview. Initially the primary benefit is for debugging COLREGS behaviors and their complex mode/submode logic.

---

250810 BHV_Waypoint (augmentation) (SOUR-JUDD)

Short: Added calc dist to end of cycle or end all points

The Waypoint behavior, via augmentation of the WaypointEngine, now maintains a running calculation of the distance to the end of the current cycle, and the distance to the final waypoint. The final waypoint and the waypoint at the end if the current cycle are the same if the waypoints are not being repeated. This calculation enables to additional Waypoint behavior macros, CYC_DIST_TOGO and ALL_DIST_TOGO, which can be used in any existing flag. It also enables the feature of reducing the desired speed used in this behavior as it approaches the final waypoint.

---

250810 XYSegListr (bugfix) (COLD-REED)

Short: Fix edge case returning centroid for segl size 1.

The XYSegList function for returning the centroid now properly handles the case where the seglist only has one vertex. The centroid now is that vertex, whereas before it returned a "null" vertex at location 0,0.

---

250810 BHV_Waypoint (augmentation) (VERY-MENU)

Short: Added policy for slowing upon arrival last point

The Waypoint behavior normally has a transit speed that remains steady until the last waypoint has been achieved. With this change the user may specify a distance to the final waypoint at which it will begin backing off the desired transit speed, ultimately using a speed of zero with it reaches another threshold distance away from the final point. This parameter is end_spd. For example end_spd=slow_dist=15,stop_dist=5.

---

250615 pMarineViewer (augmentation) (HALS-LINK)

Short: Added options for handling vehicle extrapolation

A new config parameter dictates how/whether vehicle extrapolation is implemented for semi-stale node reports. The extrap_policy parameter lets the user choose the mode, decay policy and threshold for when extrapolation is calculated. Supported modes are hdg, cog, and off.

---

250615 pContactMgrV22 (augmentation) (COOL-GULL)

Short: Added ability to expunge contacts

Added a new action option to filter messages. The action=expunge message will result in the contact mananger immediately removing any record of the contact (or source of contacts if vsource=ais for example). And the contact manager will pass along the filter message to the helm which will result in the immediate "completion" of any behaviors matching the named contact or vsource. To enable this, the contact manager must name the incoming variable to receive these requests. Presumably something like SMR_TARGET_EXPUNGE. Note, if new contact reports are subsequently received for an expunged contact, the contact manager will resume tracking it, and the helm will spawn a new behavior. So an expunged contact is not on any kind of black list.

---

250615 pHelmIvP (augmentation) (DARN-DAVE)

Short: Modified ABLE_FILTER handling reduced memory

The initial implementation of handling ABLE_FILTER messages by the helm used a history stack approach where the N most recent messages are stored, FIFO, and reapplied to the behaviors on each iteration. This approach was used for handling the example case where (a) all AIS behaviors are disabled, and (b) a new AIS contact arrives. In this case the new contact would spawn a new behavior and it would immediately be in the disabled mode. However, there is non-trivial CPU load by applying this entire history stack to all behaviors on every iterations. In this mod, a history stack is no longer maintained, and filter messages are simply applied to behaviors as they come in. So in the example provided, a new AIS contact after all AIS contacts have been disabled, would indeed result in a new behavior spawned that is NOT in the disabled mode. This caveat is worth the reduced complexity and reduced CPU load of maintaining a filter history stack. Furthermore, operators indicate that when/if all AIS contacts are disabled, then there is typically also no new incoming node reports for AIS contacts, thus the edge case that may have warranted the extra complexity is not a case that happens in practice.

---

250615 BHV_AvdColregsV22 (bugfix) (FINE-FRED)

Short: Reduced per-contact debugging verbosity

Contact behaviors have a configuration parameter, post_per_contact_info, which by default is false. When it is set to be true, contact related behaviors are free to publish debugging variables that include the contact name in the MOOS variable name. For example RANGE_BEN=45, or BEARING_CONTACT_003=11.2. Publishing these kinds of variables are problematic since there may be many hundreds of contacts over the course of a long mission and the number of MOOS variables and will grow unbounde, and certain MOOS status messages will also grow unbounded. Nevertheless they are very useful to have at times. For the COLREGS behavior, a few of these types of variables were being published without the conditional check for post_per_contact_info.

These postings have been correctly moved to be conditionally posted now.

---

250615 pHelmIvP (bugfix) (VERY-VOSS)

Short: Fixed bug where stale contact behaviors not deleted

Contact related behaviors typically are deleted when they go out of range from ownship. When (a) multiple contacts appear close to ownship, and for whatever reason, (b) neither the contact nor ownship are moving away from each other, and (c) the contact becomes stale (node reports cease to arrive) then the helm prior to this mod would have an unbounded growth of spawned and stale behaviors. In practice, the above three conditions are rare. With this mod, the helm has new parameter, contact_max_age, set by default to be 45 seconds. If no node report is received by the helm for a contact, after this period, the helm will delete the behavior. The accepted range for contact_max_age is between 10 and 1800 seconds (30 mins).

---

250615 pHelmIvP (augmentation) (NEAT-PARK)

Short: Added bhv status output for checking cmgr handling

The helm now produces IVPHELM_BHV_ACTIVE_CNT, IVPHELM_BHV_RUNNING_CNT, IVPHELM_BHV_IDI IVPHELM_BHV_DISABLE_CNT, and IVPHELM_BHV_DISABLED variables, for validating helm and contact manager performance as contacts arrive and depart, and as operators change the status of contacts by either disabling, enabling or expunging contacts.

---

250417 pContactMgrV20 (augmentation) (FIVE-BANG)

Short: Added ability to disable/re-enable alert IDs

The contact manager works in coordinatio with the helm, where contact related behaviors register an alert with the contact manager to indicate when a behavior should be spawned. The contact manager is also capable of generating a warning when the behavior will be soon spawned. In this modification the contact manager can disable an alert via the BCM_ALERT_REQUEST message, specifiying the alert ID and the action of enable or disable. This allows a user or run-time app to quiet possible warnings about behavior spawnings if there is otherwise reason to believe those behaviors will not be spawned.

---

250415 pHelmIvP (augmentation) (JIMS-CARL)

Short: BHV_AvoidObstacleV24 augmented to support vsource disabling

The obstacle avoidance behavior was augmented to handled cases where the source of the obstacle is set, and there has been an in-mission request to disable obstacle avoidance behaviors related to a specified vsource.

---

250414 pHelmIvP (augmentation) (FOXY-WEBB)

Short: Helm to disable/re-enable behaviors based on contact vsource

The helm handles requests to enable/disable a (typically contact) behavior via the BHV_ABLE_FILTER message. Up to this point disable criteria was based solely on the name of the contact. Now it accepts additional criteria such as the vsource (source of the contact info, e.g., AIS or radar, or

the contact name. The ContactLedger class was updated to handle vsource information, and the LedgerSnap class was also update to include vsource information. Currently behavior disabling is handled in the IvPContactBehavior superclass. But behavior disabling is defined at the IvPBehavior level to allow for future cases where behavior disabling is desired for non contact behaviors. The IvPBehavior superclass added a new virtual function, applyAbleFilter(), to be overridden by any behavior that wishes to respect filter disabling. The IvPBehavior superclass also now supports a new parameter, `can_disable`, which by default is false. To support filter disabling the behavior type needs to handle requests (override the applyAbleFilter() method), and the behavior in the mission file must declare `can_disable=true`.)

---

250414 pContactMgrV20 (augmentation) (LIKE-PLUM)

Short: Added ability to apply contact filter based on vsource

Augmented the contact manager to accept disbable/enable incoming messages where the criteria is based on "vessel source", e.g., from AIS, radar and so on. As with enable/disable based on vessel name/ID, the message comes via a configure MOOS variable for receiving enable/disable messages, e.g., XYZ_DISABLE_TARGET. Disabling based on vessel source may come in a message like "action=disable, vsource=AIS".

---

250410 lib_contacts (augmentation) (IDLE-IKER)

Short: Added vsource field optional part of node records

A node record has a name, type and group, and now it has a vsource field. The latter is meant to describe the source of an incomnig node record, e.g., from an AIS system, camera, radar, or an outgoing report, ownship. This can enable a subscriber of node reports, e.g., the helm or a contact manager to be able to filter out reports from certain sources.

---

250409 pMarinePIDV22 (augmentation) (EASY-LOOK)

Short: Quieted DEPTH related subs in non-uuv missions

The PID controller app, pMarinePIDV22, has the ability to support PID control for UUV (underwater) platforms. In USV (surface vehicle) applications, the PID controller had been registering for a few MOOS variables, e.g., NAV_DEPTH, NAV_PITCH, DESIRED_DEPTH etc. This fix quieted these subscription for mission not inolving depth. Depth control is enabled the required depth control config parameters are provided. If they are ommitted, then it is assumed that no depth control is needed and the corresponding registrations are also omitted.

---

250222 BHV_AvoidObstacleV24 (bugfix) (WISE-COLT)

Short: Handle helm not config w/ platform model

The AvoidObstacleV24 behavior is the first behavior to use the new platform model feature of the helm. However the behavior was not properly handling the case where the helm was not configured with a Dubins model. This has been corrected, to default the helm to a holonomic model, and post a warning that a Dubins model is prefered for this behavior. The warning can be silenced by settting holonomic_ok = true when configuring the behavior.

---

250218 pMarineViewer (augmentation) (THIN-SPAT)

Short: Added SMR vessel type in viewer applications

Added a new vessel type for the SMR USV, selectable with the vessel type "smr" in pNodeReporter. The type will render in pMarineViewer and alogview.

---

250216 pNodeReporter (augmentation) (MILD-OMNI)

Short: Added static node report riders

A node report generatxed by pNodeReporter has certain fields present by default, such as lat/lon, heading, speed. The app can now be configured to include custom additions by param=value pairs in the with the platform_aspect config parameter. These pairs are tagged onto every node report and are not tied to any incoming MOOS variable (as with dynamic riders).

---

250215 pNodeReporter (augmentation) (DULL-VIEW)

Short: Enabled dynamic node report riders

A node report generated by pNodeReporter has certain fields present by default, such as lat/lon, heading, speed. The app can now be configured to include custom additions by specifying (a) a MOOS variable to subscribe for and watch, (b) a field name in the custom node report, (c) the frequency of the addition to the node report, and (d) the precision of the numerical value of the field if it is of type double.

---

250214 lib_bhvutil (bugfix) (DULL-NICO)

Short: Bugfix in RefineryCPA for missions with non-zero min speed

The RefineryCPA utility is used with collision avoidance behaviors to create IvP functions with far fewer pieces. An inner loop could be entered indefinitely in certain conditions, essentially hanging the helm. This was triggered in helm configurations where the IvPDomain for speed had a minimum value different from zero. This may be the case for missions with air vehicles where choosing a speed of zero is not an option. For this reason it went un-noticed until this bug fix. The bug fix now handles cases with non-zero min speed.

---

250127 uFldObstacleSim (augmentation) (POSH-LYLE)

Short: Added request to pLogger to log obstacle file

The obstacle file is ingested upon startup of uFldObstacleSim. At that point a message is posted to pLogger, PLOGGER_CMD to copy the obstacle file into the log folder.

---

250126 pMissionEval (augmentation) (WIDE-NOTE)

Short: Support for new macros, report formats, appcast output

Added several new macros the user can use for formatting custom report lines, including time/date options, the mission form, mission mod, and grade. User has more separator options in formating report lines. The appcast output provide supported macro confirmation for user defined macros derived from naming MOOS variables in a report column. Appcast output also better conveys

progress in overall mission test. Normally, once a mission and evaluation has been developed and tested, appcast output is not seen, since the mission is presumably being run headlessly. Improved appcast output is helpful when developing/debugging the mission.

---

250125 pHelmIvP (augmentation) (DRAB-HACK)

Short: Added check for initial nav solution before proceeding

The helm now supports a grace period, nav_grace=N, upon startup. Within this grace period it will not post an appcast warning if the nav values have not yet been received, x, y, speed, heading and depth if an underwater vehicle.

---

250113 lib_contacts (augmentation) (ZERO-TONE)

Short: Added JSON serialization for NodeRecords

The string2NodeRecord deserialization method now checks if the string is in CSP or JSON format and deserializes accordingly.

---

250112 alogview (augmentation) (HALS-CLEM)

Short: Added ability detach a numerical variable for plotting.

Normally alogview launches and offers the option to plot any numerical variable from the log files. However, string variables with multiple numerical parts, e.g., POINT=x=4,y=5,z=0, are not plottable. With the command-line option --detatched=POINT:y, alogview, in the pre-launch splitting phase, will create a separate option to plot a variable presented as POINT:y.

---

250112 alogsplit (augmentation) (FULL-CARY)

Short: Added ability to isolate numerical subcomponents of strings

Normally alogsplit will split into klog files based on logged MOOS variable name, e.g, NAV_X will be split into a file NAV_X.klog. And a variable say POINT which may have values like x=4,y=5,z=0 is similarly split into a file POINT.klog. The alogsplit app now allows the user to specify a "detached" variable and sub-component. For example, on the command-line, the argument --detached=POINT:y, would result in the creation of the file POINT:y.klog. This is primarily useful because the internal splitting functionality is also used by alogview, and this allows alogview to plot numerical variables that may be buried in a multi-part string. This functionality works on both CSP and JSON string formats.

---

250114 pNodeReporter (augmentation) (BONY-ANTS)

Short: Added option to publish JSON formatted reports

The pNodeReporter app normally publishes node reports in CSP formatted strings, e.g, NAME=abe,LON=42.35843
71.087448,SPD=2.2. The user now has the option to publish this in JSON format, e.g., "NAME":"abe","LON":42.35
71.087448,"SPD":2.2. By setting json_report=true, the node reports are in JSON format. By setting json_report=VAR, an additional node report is published to VAR, while the CSP formatted report is published as normal.

---

250114 lib_mbutil (augmentation) (VAIN-FONT)

Short: Added JsonUtils for converting to/from JSON to CSP strings

The most common format for MOOS-IvP strings for complex object is the common-separated pairs format, e.g., x=23,y=5,z=7. Some users prefer to deal with JSON strings, e.g., "x":23,"y":5,"z":5. These two new utilities allow for conversion between formats.

---

250111 pContactMgrV20 (augmentation) (DARN-ROOM)

Short: Added support for early warning time

The contact manager, pContactMgrV20, supports early warning configuration with earl_warning_time=N. When configured, configurable warnings as event flags are generated when the contact comes within range to ownship, roughly 30 secs prior to an alert event. If there are multiple alert events configured for contacts, the early warning time is relative to the event with the largest range value. The user may configure to have early warning radii rendered as XYCircle postings. In this augmentation, the feature of early warning based on reference speed and fixed warning distance was disabled. Benefits: The newer approached is automatically tied to the configured alert of the contact behaviors, and the configuration parameter of "warning time" is essentially in the units that the operator cares about.

---

250108 uSimMarineV22 (augmentation) (WAVY-REID)

Short: Added option to publish USM_NAVPOS_SUMMARY

uSimMarineV22 will publish both NAV_X/Y and NAV_LAT/LONG as doubles, and pLogger records double out to 5 decimal places. The USM_NAVPOS_SUMMARY will publish a string with both, and at sufficient precision, e.g., x=68.932,y=-60.32,lat=43.824767,lon=-70.3295309. To opt for this publication the new configuration parameter: `post_navpos_summary=true` should be used in the uSimMarineV22 configuration.

---

250103 pContactMgrV20 (bugfix) (SICK-CODI)

Short: Early warning system fixes and rel speed change

The early warning feature added previously was fixed and augmented to use the relative speed between ownship and the contact in its policy to expand the warning range. A new mission, berta_cmgr was created and added to missions-auto for testing the new features.

---

241229 uFldBeaconRangeSensor (augmentation) (SOFT-IVER)

Short: Integrated ContactLedger into the app

The ContactLedger utility was integrated to handle all incoming NodeReport messages. The ContactLedger has built-in support for dynamic datums and handling of global and local coordinates.

---

241229 uFldContactRangeSensor (augmentation) (WIDE-RITZ)

Short: Integrated ContactLedger into the app

The ContactLedger utility was integrated to handle all incoming NodeReport messages. The ContactLedger has built-in support for dynamic datums and handling of global and local coordinates.

---

241229 uFldCollObDetect (augmentation) (GRIM-KALE)

Short: Integrated ContactLedger into the app

The ContactLedger utility was integrated to handle all incoming NodeReport messages. The ContactLedger has built-in support for dynamic datums and handling of global and local coordinates. While uFldCollObDetect is not part of the autonomous decision-making process, it is an essential tool for monitoring and validating the obstacle avoidance algorithms.

---

241228 uFldObstacleSim (augmentation) (NUDE-GROG)

Short: Integrated ContactLedger into to the simulator

The ContactLedger utility was integrated to handle all incoming NodeReport messages. The ContactLedger has built-in support for dynamic datums and handling of global and local coordinates. While uFldObstacleSim is not part of the autonomous decision-making process, it is an essential tool for monitoring and validating the obstacle avoidance algorithms.

---

241228 uFldCollisionDetect (augmentation) (FULL-DUKE)

Short: Integrated ContactLedger into the collision detector

The ContactLedger utility was integrated to handle all incoming NodeReport messages. The ContactLedger has built-in support for dynamic datums and handling of global and local coordinates. While uFldCollisionDetect is not part of the autonomous decision-making process, it is an essential tool for monitoring and validating the collision avoidance algorithms.

---

241222 IvPBehavior (augmentation) (MEGA-MOLD)

Short: Modified interface for accessing contact info

Contact info prior to this mod has resided in the information buffer, a pointer to which is shared between the helm and all behaviors. For most contact behaviors, the contact position was handled at the IvPContactBehavior superclass level in the updatePlatformInfo function. So this mod should not require any behavior authors to modify code unless their behavior was accessing contact info from the info buffer directly. The IvPContact behavior now access information from the ContactLedger snapshot through four new functions, e.g., getLedgerInfoDbl(), which takes the name of the contact and the field information, e.g., the x position or longitude position of the contact. This change is a preparation for the helm and behaviors supporting dynamic datums in an upcoming release.

---

241222 pHelmIvP (augmentation) (SLIM-TOUR)

Short: Integrated ContactLedger into the helm and behaviors

The ContactLedger utility was integrated to handle all incoming NodeReport messages. The ContactLedger has built-in support for dynamic datums and handling of global and local coordinates. Support was added to allow access for all helm behaviors, typically only IvPContactBehaviors, to the contact ledger. The helm behaviors no longer obtain contact information from the information buffer. This addresses a potential unbounded memory growth issue for long autonomy missions with thousands+ contact IDs posted and expiring, since the information buffer didn't clear old

contact information. The contact ledger handles expiring contacts gracefully. The helm behaviors only get a snapshot of the contacts currently held in the contact ledger and thus behaviors are not concerned with contact memory management.

---

241209 pContactMgrV20 (augmentation) (AVID-SONG)

Short: Integrated the ContactLedger into the contact manager.

The ContactLedger utility was integrated to handle all incoming NodeReport messages. The ContactLedger has built-in support for dynamic datums and handling of global and local coordinates.

---

241206 pMarineViewer (augmentation) (DANK-ACHE)

Short: Integrated the ContactLedger into pMarineViewer

The ContactLedger utility was integrated to handle all incoming NodeReport messages. The ContactLedger has built-in support for dynamic datums and handling of global and local coordinates.

---

241203 lib_geodaid (augmentation) (EASY-BAND)

Short: Added lib_geodaid with ContactLedger util class

The lib_geodaid library was created, with a the ContactLedger utility class. This class is for use in all MOOS-IvP apps that receive, store and reason about NodeReports. The ContactLedger has a Geodesy and declared datum. As node reports are received with lat/lon coordinates the the local planar coordinates are filled in, based on the current dataum setting. If a datum shifts mid-mission, any app using the contact ledger simply needs to set to the new datum, and all previously received and held node reports will be updated with new local coords. The ContactLedger was moved to its own library, rather than existing in lib_contacts, to allow apps (and behaviors) that previously had no dependency on Geodesy, to remain so.

---

241130 pContactMgrV20 (augmentation) (OPEN-ZALE)

Short: Added support for event flags when contact ages out

The contact manager will drop a contact after some period of time with no updates from the contact. There is now the user option of posting a flag upon this event, for each contact, with support for macros revealing the contact name and UTC time of timeout. This can be useful for proactively informing a UI module to cease rendering a contact.

---

241130 pContactMgrV20 (augmentation) (COLD-COBY)

Short: Added support for early warnings

The contact manager now supports an early warning capability. The user specifies a range to ownship within which a contact will trigger an early warning, typically substantially before a contact related behavior will be spawned. This allows an operator or other app to act on the early warning by choosing to disable a contact if other information indicates that the contact does not pose a safety concern. The early warning range can be speed regulated, potentially enlarging for contacts above a certain reference speed threshold.

---

241129 pHelmIvP (augmentation) (NINE-ALAN)

Short: Added support for disabling/enabling contacts

The helm can now accept requests to disable or re-enable a named contact. This change is in coordination with the changes to the IvPBehavior and IvPContactBehavior to support the new disabled state for behaviors. HelmEngine maintains an ordered cache of disable/enable requests, to ensure that such requests are applied even if a behavior has not yet been spawned.

---

241129 pContactMgrV20 (augmentation) (FLAT-LADY)

Short: Added support for disabling/enabling contacts

The contact manager can now optionally accept input through a user-provided variable to broker the helm the disabling of behavior(s) associated with a named contact. Same is true for re-enabling a contact. The contact manager will keep a local slate of disabled/enabled contacts. The contact manager has new flags associated with the disabling/enabling events with associate macros. The latter allows the contact manager to post confirmations to whatever app requested the disabling or enabling.

---

241127 aloghelm (augmentation) (DRAB-SLUM)

Short: Added support for disabled behaviors

The aloghelm output for the –bhvs mode, now reflects disabled behaviors as part of its standard output. This is a good post-mission method for verifying behavior disabling and re-enabling proceeded as intended.

---

241117 IvPBehavior (augmentation) (GOOD-SOUP)

Short: Added new state, disabled, alongside idle, running etc

In certain, especially USV, operations an operator may wish to disable a safety behavior for a specific contact or set of contacts by direct command. Although the contact manager can be configured to filter out contacts based on a number of criteria, in this scenario, the behavior may have already been spawned, and the operator may simply wish to apply their discretion to filter out an active behavior. When a behavior is in this new ”disabled” state, it will essentially do nothing. The behavior author can implement an onDisabledState() function, and users can utilize the new disable_flag and enable_flag options. Run-time disabling is communicated to the helm with the BHV_ABLE_FILTER message. This can contain a message identifying the contact explicitly, and/or also identifying the behavior type. Subsequent messages to this variable can server to (re)enable the behavior. Behaviors can be configured with can_disable=false which will make them immune to run-time disable messages. For example a close-in last line of defense collision avoidance behavior can be set with can_disable=false.

---

241107 BHV_CutRange (augmentation) (VAST-NOEL)

Short: Allow users to have patience param up to 100.

The CutRange behavior allowed for a patience parameter up to 100, but the value was essentially clipped in AOF_CutRangeCPA, to be 85. Some users want the full range. The default setting and

upper bound remain the same, but now users can set `max_patience` in the CutRange behavior to be say 90 or 100. Then, a higher patience parameter will be respected.

---

241102 mhash_gen (augmentation) (HAZY-KIAN)

Short: Added zhash capability for zbatch auto-testing

A zhash is new hash type comprised of US cities, for tagging batches of missions in auto-testing. The 'z' is due to the convention of using a zlaunch script to run a batch of missions. With distributed auto-testing over a cluster of machines, the missions may not be orchestrated by a zlaunch script, but nevertheless we settle on the term of a "zbatch" to apply to a set of auto-tested missions and the notion of "zhash" to be tagged to a zbatch. This mod is mostly implemented in the HashUtils.cpp utility which is used by mhash_gen when passed the -z flag.

---

241031 pAutoPoke (augmentation) (PINK-WINE)

Short: New thin MOOS app for poking MOOSDB upon startup

pAutoPoke is a thin MOOS app for poking the MOOSDB with a user-configured set of pokes, typically at the start of the mission. In the initial verison of this app, the intended use is in a shoreside community, after some specified number of vehicles have been declared to be connected, based on the UFSB_NODE_COUNT variable. This ensures that the full N-vehicle plus shoreside is running before the poke(s) are made. In follow-on versions, the condition will be generalized to any logic condition. This app was added to remove complexity of the xlaunch.sh script used in auto-testing. With this app, the auto-launch simply launches the mission and the kickoff commands are injected into the shoreside when ready. In follow-on versions, when general logic conditions are supported, this app can also be run in a vehicle community.

---

241031 uMayFinish (augmentation) (HIGH-TOWN)

Short: New thin cmdline MOOS app for monitoring mission finish

uMayFinish is typically a terminal-launched MOOS app launched within a shell script, e.g., xlaunch.sh. It will connect to a MOOS community and monitor for a completion event or timeout based on DB_UPTIME. When completed, it simply exits. Presumably to allow the executing script to proceed to a next phase. For example a script could proceed to bring down the MOOS community for the mission it was monitoring. This app was created as a thinnner, simpler alternative to uQueryDB when running in a shell script running an automated test mission. The script will launch uMayFinish and when uMayFinish completes, and disconnects from the MOOSDB, the script may proceed to bring down the mission.

---

241020 IvPBehavior (augmentation) (DANK-GIBB)

Short: Added getBufferIsKnown(string)

Added the getBufferIsKnown(string var) function to allow behavior authors to test whether the info buffer has ever had the var.

---

241020 BHV_AvoidObstacleV24 (augmentation) (MEEK-OVAL)

Short: Additional macro support for obs id and min_util_cpa

The BHV_AvoidObstacleV24 behavior was augmented to support addional macros, (a) MINU_CPA, for the config parameter for min_util_cpa, (b) MAXU_CPA, for the config parameter for max_util_cpa and (c) the OIDX, for the obstacle ID based on the last component of the obstacle label, separated by underscored, e.g., "1234" from a label "alpha_1234".

---

241020 BHV_FixedTurn (augmentation) (MALE-KENT)

Short: Added alt subscription for DESIRED_RUDDERX

The FixedTurn behavior will normally subscribe for DESIRED_RUDDER simply to log the rudder values during a turn. It is not part of the behavior functionality. When using uSimMarineV22 with an embedded PID controller, DESIRED_RUDDER is not published. uSimMarineV22 can be configured with post_des_rudder=DESIRED_RUDDERX to publish the internal rudder values and feed the FixedTurn behavior in this operation.

---

241020 uSimMarineV22 (augmentation) (SOFT-ALEX)

Short: mod uSimMarineV22 to halt on OBSTACLE_HIT=true

uSimMarineV22 had a partially implemented feature to react to a posting to OBSTACLE_HIT=true, resulting in the vehicle coming to stop. This mod ensures that the event (a) more immediately results in speed=0, and also ensures that the NAV_*_GT values also instantly halt.

---

241017 BehaviorSpec (augmentation) (FLAT-MARK)

Short: Added optional spawning cap for templating behaviors

For a behavior configured for templating, prior to this change, an arbitrary number of instances could be spawned. For a behavior like collision avoidance, this is fine. For a goal-oriented behavior like CutRange or Trail, it doesn't make sense to spawn more than one instance at a time. This mod allows for a behavior to be configured to cap the number of instances to an arbitrary number like 1. NOTE: currently if a behavior is spawned and completes, the completion event does not decrement the counter.

---

241012 AOF_R16 (augmentation) (FAST-RIOT)

Short: Option added to disallow any port turn during giveway

A user opted for a slightly different conservative policy in the COLREGS behavior, specifically in the "Give-way" mode. It pertains to which set of maneuvers are outright prohibited in this mode. The AOF_R16 class was modified to support conservative policy as a configuration option.

---

241010 BHV_AvdColregsV22 (augmentation) (SAFE-CODE)

Short: Additional status macros, quieter idle output

The BHV_AvdColregs behavior produces a number of status messages, mostly for post-mission analysis/debugging. To facilate users who wish to make their own status messages, a number of supported macros were added. These include AVD_MODE (the main avoid mode, e.g., headon,

giveway), AVD_SMODE (the sub-mode when the main mode has sub-modes), APWT (the applied priority weight given by the static priority wt and relevance, MODE_ID (a unique integer associated with all mode:sub-mode values), and FULL_MODE (simply the string comprised of mode:sub-mode). Macros may be used in any flag defined for the behavior and behavior superclasses. This behavior is now configured by default to not publish any status messages in the idle mode. This can be overridden by setting post_status_info_on_idle=true.

---

241009 uFldNodeComms (augmentation) (LONG-MELL)

Short: Integration of ContactLedger as key data structure

The uFldNodeComms app received node repprts from all vehicles, and uses this information to pass on node reports to other vehicles, and uses the node report position information to decide whether to allow messages to be routed to other vehicles. The ContactLedger class is a new super-utilty class for receiving and storing node reports. The old data structures were swapped out and replaced with the ContactLedger. This allows this app to support dynamic datum changes.

---

241009 ContactLedger (augmentation) (ZERO-LAND)

Short: Introduced ContactLedger class, NodeRecord mods

The ContractLedger class is a new super-utility class to be used in all applications that receive and reason about node reports. It ensures that every node report has both Lat/Lon and X/Y coordinates, converting as needed if a node report has only one or the other. It the dynamic update of a datum, recalculating local coordinates on each update. It also is able to answer queries about node report staleness and requests to remove stale nodes. It it first being integrated into uFldNodeComms, but will also be integrated into pMarineViewer, pHelmIvP, pContactMgrV20 and any other app reasoning about contacts.

---

241003 pHelmIvP (augmentation) (MEEK-RUST)

Short: IvPHelm now handles node reports w/ only lat/lon

Prior to this change, the helm received node reports assuming each node report contained local X/Y coordinates. The helm needs the local coordinates for many behaviors using planar math. Things are moving to where all inter-vehicle messages with position coordinates will use only lat/lon coordinates and not local coordinates at all. The helm was augmented to have an instances of MOOSGeodesy to allow incoming node reports to convert reports with only lat/lon to also have local coordinates, stored locally in the helm for access by the behaviors. This change was to HelmIvP.h/.cpp.

---

241003 lib_contacts (augmentation) (VERY-LADY)

Short: NodeRecord class serialize only global coords

Serialization of node records happends in the NodeRecord class. The class was augmented to have a "coord_policy" which can be set to global, resulting in all serializations using only lat/lon coordinates while omitting local x/y coordinates.

---

241005 mhash_gen (augmentation) (DRAB-EMIL)

Short: mhash_gen –shore,-s produces short mhash

The mhash_gen utility produces a full mission hash, e.g., 241005-1954L-SOFT-LEAF. With the –short (-s) option, the short version can be produced, e.g., SOFT-LEAF. This is useful in some shell scripts.

---

241004 pMarineViewer (bugfix) (PINK-BRAM)

Short: Fixed infocast pane browser reset upon refresh

Infocast panes with content larger than the pane will result in a scroll bar. Using the scroll bar only worked until an instant later when the pane was updated and the scroll position reset to the top. Previous work-arounds included (a) making the pane bigger or (b) pausing the infocast refresh. This bug fix ensures the position of the content of the pane remains the same even as refreshes are arriving.

---

241003 pNodeReporter (augmentation) (LIVE-JAWS)

Short: Enable node reports to be posted with just lat/lon

New support for a configuration that results in NODE_REPORT messages containing only Lat/Lon value and NOT include X/Y coordinates. This is to support a larger effort to no longer use local coordinates between vehicles. To enable this mode of operation, a new configuration parameter was added, coord_global_policy. When set to true, only LAT/LON position information will be present in outgoing NODE_REPORT_LOCAL messages. Note, for backward compatability purposes, coord_global_policy is FALSE by default. Perhaps in the furture, the inclusion of X/Y in reports will be deprecated, but for now we err on the side of no unexpected behavior for those who are not concerned by this issue.

---

241003 pMarineViewer (augmentation) (GRIM-WINS)

Short: Handle node reports with only lat/lon and no x/y

Now able to handle incoming NODE_REPORT messages that do not contain an X/Y value, only LAT/LON. Part of the migration to use only global coordinates between vehicles and command/control. This will support certain users where vehcles have different datums, or set their datum to the current vehicle position at time of mission start.

---

241001 pMarineViewer (augmentation) (HALF-PAWS)

Short: Added Infocast Layout options esp for swarms

Support was added for optional infocast pane layouts. In addition to the default layout (nodes upper left, procs upper right, cast on the botttom), two other layouts are supported. A ”swarm” mode with nodes taking the entire PMV height on the left, and a ”fullcast” mode where only the cast pane is shown large.

---

240930 pMarineViewer (augmentation) (JUST-ZIGS)

Short: Added Infocast panes darker colors

Support was added for a few more darker colors for infocast background. To help in certain lighting situations or on monitors where lighter colors appear washed out.

---