uFldScope: Gathering a Multi-Vehicle Status Summary

June 2018

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

1 Overview

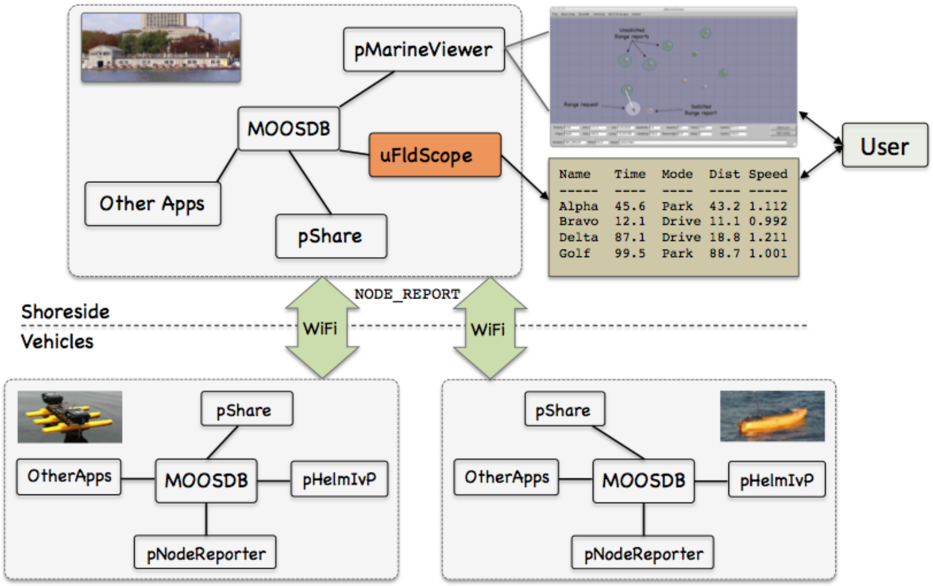
The uFldScope application is a tool for collecting diverse sets of information regarding a field of vehicles remotely deployed. Suppose, for example, one is interested in monitoring, for each deployed vehicle, the (a) helm mode, (b) total distance travelled, (c) battery level, and (d) the number of times it has visited a certain beacon. Each piece of information may be embedded in one of a number of MOOS variables, perhaps along with a lot of other information of no concern. For example, a typical NODE_REPORT posting contains the helm mode, but the full string may look like:

```
NODE_REPORT= "NAME=alpha,TYPE=UUV,TIME=1252348077.59,X=51.71,Y=-35.50,LAT=43.824981,
              LON=-70.329755,SPD=2.00,HDG=118.85,DEPTH=4.63,LENGTH=3.8,MODE=LOITERING"
```

While there are several methods to scope on the above variable and pick out the helm mode, the goal of the uFldScope tool is to have this information readily visible for each vehicle perhaps alongside other key fields for all vehicles, in a continuously updated simple table like the following:

```
VName        MODE    TripDist  Speed  STREAMING(2)
=====     =======    ========  =====  (15)
alpha     LOITERING      66.8   1.96
bravo          PARK       0.0   0.00
charlie   RETURNING    1466.3   1.05
```

The assumption is that uFldScope is running on a topside computer, interacting with a user, and receiving information on deployed vehicles primarily through node reports or other summary report variables. The typical layout is shown in Figure 1



```
Name   Time  Mode  Dist Speed
-----  ----  ----  ---- -----
Alpha  45.6  Park  43.2 1.112
Bravo  12.1  Drive 11.1 0.992
Delta  87.1  Drive 18.8 1.211
Golf   99.5  Park  88.7 1.001
```

Figure 1: **Typical uFldScope Topology:** A shoreside or topside community is receiving information from several deployed vehicles. Key information is embedded in one of several possible MOOS report variables. The uFldScope tool runs in the topside community to parse key information from the variables and display them in a table format configured by the user.

2 Configuration Parameters of uFldScope

The following parameters are defined for uFldScope. A more detailed description is provided in other parts of this section. Parameters having default values are indicated so.

Listing 2.1: Configuration parameters for uFldScope*.*

scope: Description info include in main report along with source info.
layout: An alternative table layout showing only selected fields in each report.

2.1 An Example MOOS Configuration Block

To see an example MOOS configuration block, enter the following from the command-line:

```
$ uFldScope --example or -e
```

This will show the output shown in Listing 2 below.

Listing 2.2: Example configuration of the uFldScope *application.*

```
===============================================================
uFldScope Example MOOS Configuration
===============================================================

ProcessConfig = uFldScope
{
  AppTick   = 4
  CommsTick = 4

  scope = var=NODE_REPORT,key=vname,fld=TIME,alias=Time
  scope = var=NODE_REPORT,key=vname,fld=MODE
  scope = var=SPEED_REPORT,key=vname,fld=avg_speed,alias=speed
  scope = var=ODOMETRY_REPORT,key=vname,fld=trip_dist
  scope = var=ODOMETRY_REPORT,key=vname,fld=total_dist

  layout = trip_dist, total_dist
  layout = MODE, speed, Time
}
```

3 Publications and Subscriptions for uFldScope

The interface for uFldScope, in terms of publications and subscriptions, is described below. This same information may also be obtained from the terminal with:

```
$ uFldScope --interface or -i
```

3.1 Variables Published by uFldScope

The primary output of uFldScope to the MOOSDB is the report output to the terminal or appast. The only publication to the MOOSDB is the APPCAST publication.

3.2 MOOS Variables Subscribed for by uFldScope

The uFldScope application subscribes to the following MOOS variables:

- APPCAST_REQ: A request to generate and post a new apppcast report, with reporting criteria, and expiration.

The uFldScope application will also subscribe for the MOOS variables prescribed in the scope configuration parameter(s).

Command Line Usage of uFldScope

Although the uFldScope application may be launched with pAntler, it is typically launched separately from the command line:

```
$ uFldScope --help or -h
```

This will show the output shown in Listing 3 below.

Listing 3.3: Command line usage for the uFldScope tool.

```
Usage: uFldScope file.moos [OPTIONS]

Options:
  --alias=<ProcessName>
      Launch uFldScope with the given process
      name rather than uFldScope.
  --example, -e
      Display example MOOS configuration block
  --help, -h
      Display this help message.
  --version,-v
      Display the release version of uFldScope.
```

4 Configuring the uFldScope Utility

The uFldScope utility may be configured to choose (a) which MOOS variables are scoped, (b) which fields in those messages are scoped, and (c) how that data is presented to the user. In all usage scenarios it is presumed that the messages are strings comprised of comma-separated variable=value pairs. For example:

```
NODE_REPORT= "NAME=alpha,TYPE=UUV,TIME=1252348077.59,X=51.71,Y=-35.50,LAT=43.824981,
             LON=-70.329755,SPD=2.00,HDG=118.85,DEPTH=4.63,LENGTH=3.8,MODE=LOITERING"
```

For each variable specified by the user for scoping, a *key* needs to also be specified identifying the vehicle name. In the above case, the key is the string "NAME".

4.1 Configuring Scope Elements

A *scope element* corresponds to particular MOOS message and a particular field in that MOOS message. It also corresponds to a column in the tabular report presented to the user. For example, the second column in the table below, with the header "MODE", corresponds to a scope element deriving its information from postings to NODE_REPORT, in the field "MODE".

```
VName          MODE    TripDist  Speed  STREAMING(2)
=====       =======    ========  =====  (15)
alpha     LOITERING        66.8   1.96
bravo          PARK         0.0   0.00
charlie   RETURNING      1466.3   1.05
```

A scope element is configure in the mission configuration file with entries of the form:

```
scope = var=<MOOSVar>, key=<KEYNAME>, fld=<FIELDNAME>, alias=<ALIAS>
```

The <MOOSVAR> specifies the name of the MOOS variable. Multiple scope elements may use the same MOOS variable. The <KEYNAME> specifies the field in the message used to designate the name of the vehicle. There should be only one right answer for this, and if it is wrongly specified, the column for that scope element will simply be empty. The <FIELDNAME> specifies the other field in the message holding the information of interest. For example, line 10 in Listing 2 is the scope configuration resulting in the second column of the above tabular output. The <ALIAS> is a string to use in the column output header if the user doesn't want to just use the name of scoped field.

4.2 Configuring Scope Layouts

By default the tabular output produced by uFldScope contains a column for each scope element. If the number of scope elements is large the user may be interested, at times, in rendering only a subset of the scope elements. The user may define these subsets using the LAYOUT configuration parameter, of the form:

```
LAYOUT = <FLDNAME>, <FLDNAME>, ..., <FLDNAME>
```

The <FLDNAME> specifies the name of the field in a given scope element. Note that it is *possible* that the field names may be the same for two different scope elements. For this reason the field name used in the layout definition is the field name alias. This gives the user the chance to distinguish two otherwise identical field names. If the user does not specify an alias in configuration of scope element, the alias is by default the same as the field name.

For the uFldScope configuration shown in Listing 2, the following five scope elements are rendered to the user as follows.

```
VName           Time                   MODE  speed  trip_dist  total_dist  PAUSED(A)
=====  =============  =====================  =====  =========  ==========  (1800)
gilda  2651922526.73  MODE@ACTIVE:LOITERING   1.06      517.1       517.1
henry  2651922526.54  MODE@ACTIVE:LOITERING   1.17      526.4       526.4
  ike  2651922526.45  MODE@ACTIVE:LOITERING   1.07      520.1       520.1
james  2651922526.65  MODE@ACTIVE:LOITERING   1.16      515.1       515.1
```

Note the "(A)" at the end of the first line. This indicates that all scope elements are being presented. If the user hits the 'l' or 'L' keys the presentation will be toggled through the various layouts configured by the user. In this example case, the following two layouts may be selected:

```
VName           Time                   MODE  speed  PAUSED(1)
=====  =============  =====================  =====  (1802)
gilda   2651923071.3  MODE@ACTIVE:LOITERING   1.17
henry  2651923071.11  MODE@ACTIVE:LOITERING   1.17
  ike  2651923071.03  MODE@ACTIVE:LOITERING   1.18
james  2651923071.24  MODE@ACTIVE:LOITERING   1.17
```

Note that only the *Time*, *MODE*, and *speed* scope elements are produced, corresponding to the layout configured on Line 16 in Listing 2. Also note that the "(A)" on the first line switched to "(1)" to indicate that the first user-configured layout is being used. By hitting 'L' once more, the second user-configured layout will be instead shown:

```
VName  trip_dist  total_dist  PAUSED(2)
=====  =========  ==========  (1801)
gilda     1081.9      1081.9
henry     1072.1      1072.1
  ike     1117.7      1117.7
james     1067.0      1067.0
```

Note that the "(1)" on the first line switched to "(2)" to indicate that the second user-configured layout is being used. By hitting 'L' once more, the full table will again be rendered.

4.3 Further Control of the Terminal Output

If multiple layouts have been configured, the user may either toggle through the list of layouts with the 'l' or 'L' key as mentioned above, or toggle between the last-used user-configured layout and the mode of rendering all scope elements, by using the 'a' or 'A' key.

By default the output produced to the terminal is refreshed on each iteration of the uFldScope application. This may be useful for watching a trend as time passes. The user may also wish to pause the output to take a careful look at the data. This may be done by hitting the 'p' or 'P' keys, or simply the spacebar. Subsequent similar keystrokes will keep the refresh mode in the paused mode, but the output will be refreshed to their current values before again pausing. Returning to the streaming mode may be done by hitting the 'r' or 'R' keys. At any time the user may also hit the 'h' or 'H' keys for a help menu.