

# uDialogManager v5.0: User Interaction

Spring 2019

Michael Novitzky, novitzky@mit.edu  
Department of Mechanical Engineering, CSAIL  
MIT, Cambridge MA 02139

---

<b>1</b>	<b>uDialogManager v5.0: User Interaction</b>	<b>1</b>
<b>2</b>	<b>Using uDialogManager v5.0</b>	<b>1</b>
2.1	Typical Module Topology . . . . .	2
2.2	The States of uDialogManager . . . . .	2
2.3	Available Sentences . . . . .	2
2.4	Sentence Action . . . . .	3
2.5	Sentence Acknowledgement . . . . .	4
2.6	Vehicle Nicknames . . . . .	5
2.7	Using Wave Files Instead of TTS . . . . .	5
2.8	Rejecting Sentences with Word Confidence Scores . . . . .	5
<b>3</b>	<b>Configuration Parameters of uDialogManager</b>	<b>5</b>
3.1	An Example MOOS Configuration Block . . . . .	5
<b>4</b>	<b>Publications and Subscriptions for uDialogManager v5.0</b>	<b>6</b>
4.1	Variables Published by uDialogManager . . . . .	6
4.2	Variables Subscribed for by uDialogManager . . . . .	7
4.3	Command Line Usage of uDialogManager . . . . .	7
<b>5</b>	<b>Terminal and AppCast Output</b>	<b>7</b>

---

## 1 uDialogManager v5.0: User Interaction

The `uDialogManager` application is a module for interfacing with a user. It controls the user experience by confirming what the user has asked for which helps reduce error. Furthermore, `uDialogManager` is responsible for interacting with other applications such as triggering events or relaying information. The typical setup is the use of `uSpeechRec` for speech recognition and `iSay` or text output for user feedback.

Key parts needed for `uDialogManager` are the possible syntax and vocabulary combinations produced by speech recognition which are found in the vocabulary file and grammar files for the Julius Speech Recognition Engine. These files and combinations are described in [2.3](#).

## 2 Using uDialogManager v5.0

Typical use of `uDialogManager` has it situated in a community in which a human will interact with it using speech. In addition to `uDialogManager` another application in the same community must interpret the user's speech such as `uSpeechRec` and have a method of communicating back either through `iSay` or terminal output such as through appcasting.

## 2.1 Typical Module Topology

The typical module topology is shown in Figure 1 below. The `uDialogManager` is situated in a community in which speech will be used as a form of interaction. It is typically run alongside `uSpeechRec` and `iSay` for an interactive experience. The `uDialogManager` application subscribes to `SPEECH_RECOGNITION_SENTENCE` and `SPEECH_RECOGNITION_SCORE` and minimally publishes `SPEECH_CMDMANDED`, `SAY_MOOS`. As discussed further in the document, other variables and their values can be triggered by a speech recognition sentence. The `SPEECH_CMDMANDED` variable is used as a logging tool for speech commands that have been acknowledged by the user. The `SAY_MOOS` variable is published as a means to use `iSay` to give the user auditory feedback.

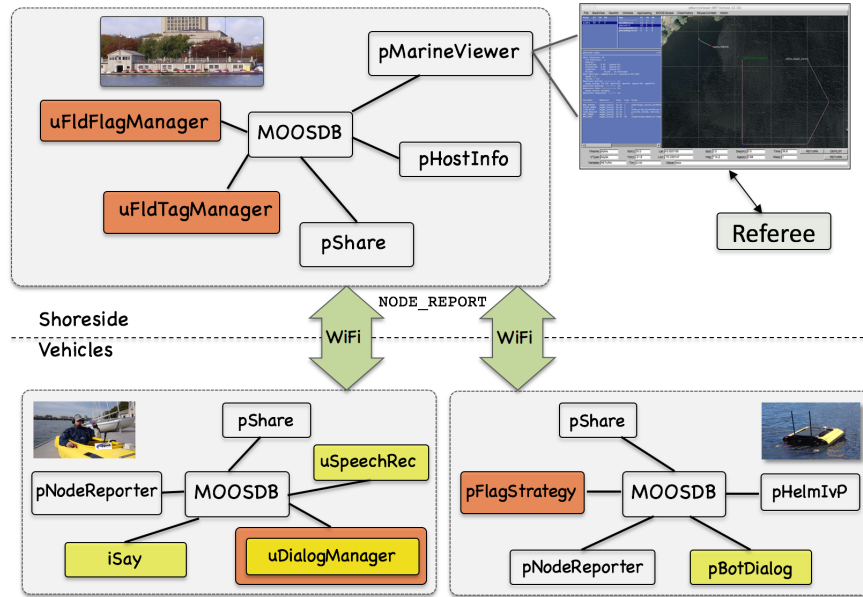


Figure 1: **Typical `uDialogManager` Topology:** This module runs in any community in which one would like to use Speech Recognition. It is typically used with the applications `uSpeechRec` and `iSay` for an interactive experience. It subscribes to `SPEECH_RECOGNITION_SENTENCE` and `SPEECH_RECOGNITION_SCORE` which comes from `uSpeechRec` and publishes `SAY_MOOS` for audio output and user specified variables based on speech sentences.

## 2.2 The States of `uDialogManager`

`uDialogManager` can be in one of several states: Waiting for Command, Command Received, Waiting for Acknowledgement, and Acknowledgement Received. These states are common among dialog managers as they control the flow of interaction.

## 2.3 Available Sentences

The available sentences that can be acted upon by `uDialogManager` are delivered by `uSpeechRec`. The `uSpeechRec` uses a grammar and vocabulary file to determine the sentence structure and vocabulary available to the Julius Speech Recognition system. At the moment these are the available sentences:

```
NAME COMMAND
ACK
```

These are the available vocabulary words:

```
NAME: Arnold, Betty, Charlie, Davis, Evan, Gus
COMMAND: RETURN, FOLLOW, STATION
ACK: Yes, No
```

## 2.4 Sentence Action

A major feature of `uDialogManager_5.0` is that a user can specify the variable-value pairs that trigger based on an acknowledged speech recognition sentence. Remember that only sentences provided through `SPEECH_RECOGNITION_SENTENCE` or `SPEECH_RECOGNITION_SCORE` are acted upon.

Let's go through an example from a `.moos` file as follows:

```
sentence = Arnold_Deploy : DEPLOY = true
```

We see that a sentence action is defined with the word `sentence` first followed by an equal sign to indicate the speech recognition sentence that will trigger the following var-value pairs. Notice that the words in the speech recognition sentence are separated by an underscore instead of spaces. The end of the speech recognition sentence is indicated by a semi-colon `'.'`. Following the semi-colon is the variable name `DEPLOY` followed by an equals sign and the value `true`. In this example, once the user acknowledges the speech recognition sentence `ARNOLD DEPLOY`, `uDialogManager_5.0` will publish `DEPLOY=true`.

Now let's look at an example that published multiple variable-value pairs. In the following example, multiple variable-value pairs are published once triggered by a speech recognition sentence.

```
sentence = Arnold_Deploy : DEPLOY = true + MOOS_MANUAL_OVERRIDE = false //
+ RETURN = false
```

In the example above the variable-value pairs are separated by a plus `'+'` sign. In this case, `DEPLOY` is assigned `'true'`, `MOOS_MANUAL_OVERRIDE` is assigned `'false'`, and `RETURN` will be assigned `'false'` when the speech recognition sentence `ARNOLD DEPLOY` is acknowledged.

In the final example we will introduce the method in which we send variables to another community and can encapsulate a string with quotations. In order to send variables to another community we publish to `NODE_MESSAGE_LOCAL` which bridges to the shoreside community through `uFldNodeBroker`. This variable is then parsed on the shoreside community and sent to the proper MOOS community based on the `dest_node` community name. Once at the target MOOS community it is posted locally the the `var_name` with the value indicated by `string_val`.

```
sentence = Arnold_Follow : NODE_MESSAGE_LOCAL = //  
"src_node=mokai,dest_node=betty,var_name=TRAIL,string_val=true"
```

We can see that the `NODE_MESSAGE_LOCAL` takes as input a string with many variable-value pairs within. To make this possible we will surround the string with quotes “.

## 2.5 Sentence Acknowledgement

A major feature with `uDialogManager_5.0` is the ability to define how a sentence is acknowledged before the variable-value pairs are posted to the MOOSDB. The three options for acknowledgement are 1) the default of replying Yes/No for Confirmation/Decline 2) user specified Confirmation/Decline words or 3) no confirmation in which once a sentence is recognized, the variable-value pairs are posted immediately. Let us look at a simple default example below:

```
sentence = Arnold_Follow : FOLLOW_ARNOLD = true
```

In this example the sentence “Arnold Follow” has the variable-value pair of `FOLLOW_ARNOLD` set to `true`. By default, `uDialogManager_5.0` will ask the user “Did you mean, Arnold Follow?” At this point, the user can confirm the speech command by replying “Yes” or decline the speech command by replying “No.” Responding with anything other than “Yes” or “No” will result in `uDialogManager_5.0` responding with “Command Canceled” “Wrong Ack”.

A user can specify different words for confirming or declining a command using the following format:

```
sentence = Arnold_Follow { CONFIRM=Verify | DECLINE=Ignore } : FOLLOW_ARNOLD = true
```

In the above example, the user has specified acknowledgement options between the curly braces, “ and ”. The user has specified the word `Verify` for confirming a command and the word `Ignore` for declining a command. In this example, the user says “Arnold Follow” and the `uDialogManager_5.0` will prompt the user with “Did you mean, Arnold Follow?” The user can confirm with replying “Verify” or decline the command with replying “Ignore.” If the user replies with anything other than `Verify` or `Ignore`, the `uDialogManager_5.0` will respond with “Command Canceled, Wrong Ack.”

When the user wants a command to simply trigger posting variable-value pairs without acknowledgement they can specify this with a `NOCONFIRM` option:

```
sentence = Arnold_Follow { NOCONFIRM } : FOLLOW_ARNOLD = true
```

As in the previous example, the acknowledgement options are between the curly braces, “ and ”. In this case, the user wants the variable-value pairs to post to the MOOSDB as soon as `uSpeechRec` recognizes the sentence “Arnold Follow.” Users are cautioned when using this option as false positives in speech recognition can lead to unwanted behavior.

## 2.6 Vehicle Nicknames

Specifying nicknames between what a vehicle is called and its lab (community) name is no longer an option (since v2.0). Instead, the destination of a `NODE.MESSAGE.LOCAL` can have the vehicle's name while the speech sentence can have the name verbalized by the user.

## 2.7 Using Wave Files Instead of TTS

`uDialogManager` can either send feedback using text-to-speech (TTS) or replaying pre recorded wave files. Specifying the `use_wav_files` to either 'YES' or 'NO' sets this option. If set to yes and the wave files do not exist, then `iSay` will post an error.

## 2.8 Rejecting Sentences with Word Confidence Scores

Originally, `uDialogManager` accepted whatever the most likely sentence that the Julius Speech Recognition Engine delivered. However, this method still had a high error rate. Specifying the `confidence_thresh` parameter (range of (0.0,1]) in the .moos file switches `uDialogManager` into using a threshold on the word confidence scores produced by the Julius Speech Recognition Engine. In general, a value of 0.7 works well. It is up to the user to experiment to determine which word confidence rejection threshold works best for their scenario.

When `uDialogManager` rejects a sentence because of low word confidence scores, it will say the phrase "SAY AGAIN" to indicate your last sentence was rejected.

# 3 Configuration Parameters of `uDialogManager`

The following parameter is defined for `uDialogManager_5_0`. A more detailed description is provided in other parts of this section. Parameters having default values are indicated so.

*Listing 3.1: Configuration Parameters for `uDialogManager_5_0`.*

- `confidence_thresh`: Can reject sentences based on word confidence scores. Threshold is in range (0,1.0]. If not specified, automatically accepts the most likely sentence without consideration of word confidence scores. Section 2.8.
- `use_wav_files`: indicate whether to use local text-to-speech (TTS) or pre-recorded wave files. Options are yes or no. Section 2.7.
- `sentence`: Assignment of an incoming speech recognition sentence to trigger a set of variable value pairs. Section 2.6.

### 3.1 An Example MOOS Configuration Block

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

```
$ uDialogManager --example or -e
```

This will show the output shown in Listing 2 below.

Listing 3.2: Example configuration of the `uDialogManager` application..

```
1 =====
2 uDialogManager Example MOOS Configuration
3 =====
4
5 ProcessConfig = uDialogManager
6 {
7   AppTick    = 4
8   CommsTick  = 4
9
10  //can reject sentences based on word confidence scores
11  //threshold value range (0,-1.0]
12  //if not specified, reverts to accepting most likely sentence
13  //without considering word confidence
14  confidence_thresh = 0.7
15
16  //indicate whether to use local text-to-speech (TTS) or
17  //pre-recorded wave files. Options are yes or no
18  Use_Wav_Files=Yes
19
20  //list of vars and values to publish given speech sentence
21  //var-value pairs are ''+' separated
22  sentence = Arnold_Deploy : DEPLOY = true + MOOS_MANUAL_OVERRIDE = false + RETURN = false
23
24  //quotes around a string for a value can be used
25  sentence = Arnold_Follow : NODE_MESSAGE_LOCAL =
26                "src_node=mokai,dest_node=betty,var_name=TRAIL,string_val=true"
27
28  //sentence ack options can be specified after a sentence inside of curly braces '{' and '}'
29  //in order to skip acknowledgment and variable-value pairs post immediately
30  sentence = grab { NOCONFIRM } : FLAG_GRAB_REQUEST = ''vname = $(VNAME)''
31
32  //the user can specify words for confirming or declining commands within the curly braces
33  sentence = tag { CONFIRM=verify | DECLINE=ignore } : TAG_REQUEST = ''vname = $(VNAME)''
34 }
```

## 4 Publications and Subscriptions for `uDialogManager` v5.0

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

```
$ uDialogManager --interface or -i
```

### 4.1 Variables Published by `uDialogManager`

- **APPCAST**: Contains an appcast report identical to the terminal output. Appcasts are posted only after an appcast request is received from an appcast viewing utility.
- **SPEECH\_COMMANDED**: A sentence that has been acknowledged by a user.
- **DIALOG\_ERROR**: posted when a sentence is rejected using word confidence score threshold.

- **SAY\_MOOS**: Either a wave file to be played for the user or a sentence to be uttered by **iSay**.

## 4.2 Variables Subscribed for by uDialogManager

The **uDialogManager** application will subscribe for the following four MOOS variables:

- **APPCAST\_REQ**: A request to generate and post a new appcast report, with reporting criteria, and expiration.
- **SPEECH\_RECOGNITION\_SENTENCE**: The most likely sentence recognized by the Julius Speech Recognition Engine.
- **SPEECH\_RECOGNITION\_SCORE**: Includes the most likely sentence and word confidence scores produced by the Julius Speech Recognition Engine.

## 4.3 Command Line Usage of uDialogManager

The **uDialogManager\_5\_0** application is typically launched as a part of a batch of processes by **pAntler**, but may also be launched from the command line by the user. To see command-line options enter the following from the command-line:

```
$ uDialogManager --help or -h
```

This will show the output shown in Listing 3 below.

*Listing 4.3: Command line usage for uDialogManager.*

```

1 =====
2 Usage: uDialogManager file.moos [OPTIONS]
3 =====
4
5 Options:
6   --alias=<ProcessName>
7       Launch uFldHazardMetric with the given process name.
8   --example, -e
9       Display example MOOS configuration block.
10  --help, -h
11       Display this help message.
12  --interface, -i
13       Display MOOS publications and subscriptions.
14  --version,-v
15       Display release version of uDialogManager.
```

## 5 Terminal and AppCast Output

*Listing 5.4: Example uDialogManager console output.*

```

1 =====
2 uDialogManager_3_0 mokai                                0/0(655)
3 =====
4
```

```

5 Sentence Action: ARNOLD_DEPLOY : DEPLOY=true + MOOS_MANUAL_OVERRIDE=false + RETURN=false
6
7 Sentence Action: ARNOLD FOLLOW : NODE_MESSAGE_LOCAL=src_node=mokai,
8                               dest_node=betty,var_name=RETURN,string_val=true
9
10 sentence = grab { NOCONFIRM } : FLAG_GRAB_REQUEST = "vname = $(VNAME)"
11
12 sentence = tag { CONFIRM=verify | DECLINE=ignore } : TAG_REQUEST = "vname = $(VNAME)"
13
14
15 CURRENT STATE:
16 Ready for Command.
17
18 CONVERSATIONS:
19
20 DM: Command Sent
21 User: Yes
22 DM: Did you mean arnold follow
23 User: Arnold Follow
24

```

Line 5 shows a sentence action that was defined in the .moos file. In this case, the speech recognition sentence of ARNOLD DEPLOY triggers the three variables `DEPLOY`, `MOOS_MANUAL_OVERRIDE`, and `RETURN` to be published with the values true, false, and false, respectively. Line 7 shows another sentence action that was defined but in this case the value for `NODE_MESSAGE_LOCAL` is a string. As described above, `NODE_MESSAGE_LOCAL` allows a local variable to be published that then contains a var/value pair to be sent to another community. Line 10 shows a sentence that skips the acknowledgement step with the NOCONFIRM option. Line 12 shows a sentence that has changed the default words of Yes and No to Confirm and Decline a command to Verify and Ignore.

Line 15 displays the CURRENT STATE of `uDialogManager`, which can be in one of several states such as Ready for Command or Waiting for an ACK. Line 20 begins the display of the last 10 sentences between the user and the `uDialogManager` with the most recent sentence at the top. In this example, the user started by saying "Arnold Follow" which is displayed at Line 23. The `uDialogManager` responds with "Did you mean arnold follow" which is shown in Line 22. Lines 21 and 20 demonstrate that the user acknowledged with a "Yes" and then `uDialogManager` responded with "Command Sent." If there is a configuration or runtime warning then Line 4 would be replaced with .moos file issues.