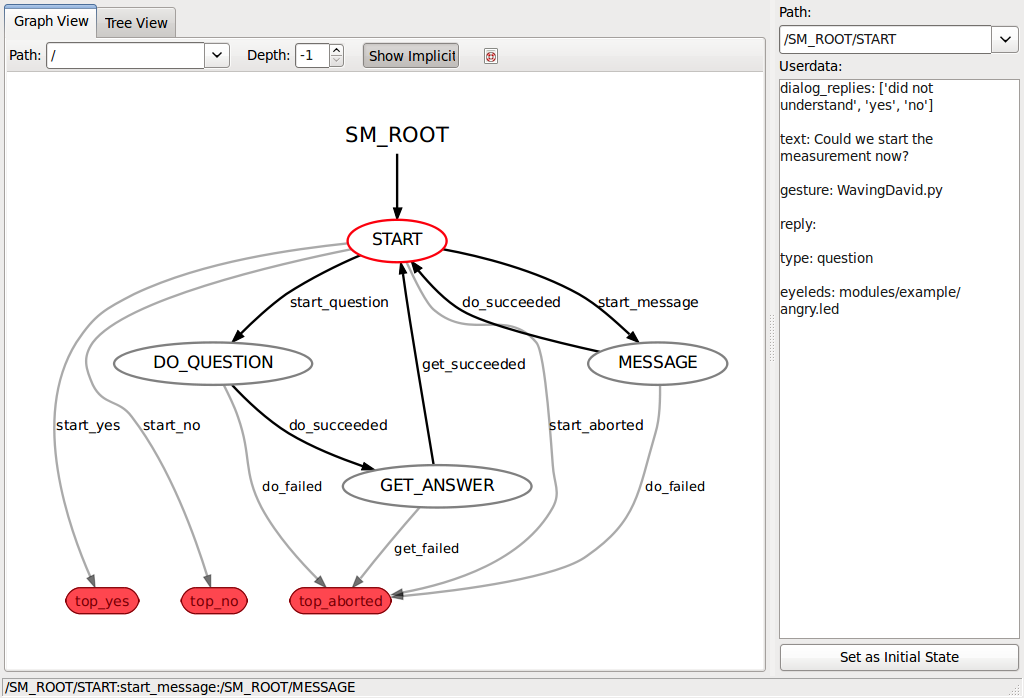
Dialogue State Machine

Below is the preliminary description of the DialogStateMachine.py. It invokes the StartState.py, DoQuesion.py and GetAnswer.py. All reside in the smach\_tutorials/KSERA\_nodes/SM\_PT2/Client folder. The old versions were called sm\_dialog1.py and sm\_dialog\_class.py which contained all parts in one file. Several additions have been made in terms of parsing message\_logic. Otherwise both versions are the same

Dear Marco and others,

Here is an update on the dialog node.



I have updated and fixed all known issues (but two, see below) of sm\_dialog1.py. As it is a module that needs to be imported it cannot be used to create multiple instances. Therefore I made a new version called sm\_dialog\_class.py. Because it is a Class many instances can be created from the same file, and each instance can be initialized with the class constructor. This is somewhat different from the old version, so I made a new file.

### Testing the node directly:

[Obsolete]: Use the DialogTesterGui.py if you want to test dialogs.

If your nao is not responding to "nao.local", edit the IP address of Nao at the bottom of sm\_dialog\_class.py to the correct value (look for line 372 sm\_top=SMDialog("ask\_measurement.csv").sm\_top and change it to sm\_top=SMDialog("ask\_measurement.csv", yourip, yourport).sm\_top, where yourport is optional as it defaults to 9559 if omitted).

Type the following commands in the Terminal (you can use shift+ctrl+t or ctrl+alt+t to create new shells).

roscore

(new shell or type ctrl+z and bg)

roscd ASRsphinx/bin

./ASRsphinx

(new shell)

rosrun smach\_tutorials sm\_dialog\_class.py

### Testing the stripped down version of Client\_IF1.py:

[Obsolete]: Use DialogTest\_1\_AskMeasurement.py etc. or DialogTesterGui.py

Replace the last command by:

rosrun smach\_tutorials Client\_IF1\_DialogOnly\_test.py

The Client\_IF1.py reads the config/ksera\_config.prop file to find the Nao IP address and creates two instances of sm\_dialog\_class.py. They are initialized with two different dialogfiles (one of which is especially designed for Marco ;-)

So Marco, please check if it works with the real robot. Pay attention to the timing, because that may need some tweaking.

### Here is how it works:

[Almost the same]: replace sm\_dialog\_class by DialogStateMachine

Suppose you have dialog files dialog1.csv, dialog2.csv, etc use the following python code

#import relevant stuff

from sm\_dialog\_class import SMDialog

from readconfigfile import ReadConfigKey

#Read nao ip

mykey=ReadConfigKey('IP\_NAO')

mynao\_ip=mykey[1]

#Create two instances of the state machine class SMDialog() and extract the statemachine sm\_top:

sm\_dialog1=SMDialog("dialog1.csv", mynao\_ip).sm\_top

sm\_dialog2=SMDialog("dialog2.csv", mynao\_ip).sm\_top

Optionally you can add nao port number as a third argument.

# add them to your own state machine or run them directly using sm\_dialog1.execute()

...see Client\_IF1\_OnlyDialog\_test.py as example

### Output:

[More or less the same]:

First the statemachine initializes its states:

If Roscore and ASRsphinx are not running it will wait for them until they are launched

If Nao is not found you will get either a quick tcp\_ip() failed to connect or a slow sequence of ALxxx timeout errors

You should see Nao's ip address if successfully connected

Then it reads the dialog file. If found you will see Parsing keys: ..., Parsing logic: ..., Parsing Message\_logic: ... where the dots list the keys. This shows that the dialog file is read correctly.

Then the states are executed and you will see the following output:

Entering Start:

...

Entering DoQuestion:

...

Nao: Could we please do a measurement now?

Nao LEDs: angry.led

Nao movement: WavingDavid.py

...

Entering GetAnswer:

[empty line] This is where the speech recognition is active

Press (y)es, (n)o, (r)epeat or (a)bort

You pressed yes

...

Entering Start

...

Entering Message

Nao: Thank you, I will be back tomorrow

Nao eyeLEDs: happy.led

Nao movement:

...

State Machine terminating do\_succeeded->top\_yes bla bla

...

The rest of the output is from the parten sm in Client\_IF1, which continues with the next dialog.

### Solved issues:

- The getch() command does not return when not run directly. Some issue with unix terminals I do not understand. Possible solution is to use opencv's cv.WaitKey()?

- If you use the simulated nao by running ./naoqi from the terminal, the nao.Say("some text") crashes the program. For now, comment out this line if you want to do this.

Kind regards,

Raymond