Возможности и принципы работы dialogmanager. Данный модуль предназначен для обаботки диалогов и ответной реакции на них, **работает не только с NAO**. Итак, как что куда и для чего писать.

Библиотека состоит из одного класса, и обработчика диалогов в нем. Что к чему?

Есть 2 метода, по обработки вопросов разного типа. Вопрос с бинарным ответом(да, нет) и вопрос с множественными вариантами ответа.

* **askQ(question, answMap):**
  + **question**
    - Функция, задающая вопрос
  + **answMap**
    - словарь возможных ответов и реакция на них
* **askBinaryQ(question, yesF, noF):**
  + **question**
    - Функция, задающая вопрос
  + yesF
    - в случае если ответ да, выполняет эту команду
  + noF
    - в случае если ответ нет, выполняет эту команду

Примеры использования

# -\*- encoding: UTF-8 -\*-

import time

import socket

import select

import smtplib

import ssl

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

from email.mime.image import MIMEImage

from naoqi import ALProxy

from naoqi import ALBroker

from naoqi import ALModule

# import sys, os

# path = os.path.dirname(os.path.dirname(os.path.dirname(os.path.abspath(sys.argv[0]))))

# sys.path.append(path + '\custom\_lib')

from eventmanager import binaryPredicate, changedValuePredicate

from dialogmanager import \*

def makeAndSendPhotoToEmail(robot\_IP, robot\_PORT=9559):

    # receiver\_address="nikolafgh@gmail.com",

    # sender\_address = "tsinaolab@gmail.com", sender\_pass = "#TSIRiga2018!",

    # resol = 4, picFormat = "jpg",

    # subject="Hello from NAO", mail\_content="Hi, this is your photo"):

    global tts, memory, motion, alife, touch, vision, animation, anitext, asr

    tts = ALProxy("ALTextToSpeech", robot\_IP, robot\_PORT)

    anitext = ALProxy("ALAnimatedSpeech", robot\_IP, robot\_PORT)

    motion = ALProxy("ALMotion", robot\_IP, robot\_PORT)

    alife = ALProxy("ALAutonomousLife", robot\_IP, robot\_PORT)

    touch = ALProxy("ALTouch", robot\_IP, robot\_PORT)

    vision = ALProxy("ALPhotoCapture", robot\_IP, robot\_PORT)

    animation = ALProxy("ALAnimationPlayer", robot\_IP, robot\_PORT)

    posture = ALProxy("ALRobotPosture", robot\_IP, robot\_PORT)

    memory = ALProxy("ALMemory",robot\_IP,robot\_PORT)

    asr = ALProxy("ALSpeechRecognition", robot\_IP, robot\_PORT)

    motion.stopMove()

    posture.goToPosture("Stand",0.4)

    alife.setState("safeguard")

    memory.insertData("WordRecognized", [0,0])

    global loop

    loop = True

    global dialogHandler

    dialogHandler = DialogManager(robot\_IP)

    dialogHandler.askBinaryQ("Do you want to take a picture?", lambda: takePhoto(robot\_IP), no)

    while loop:

        dialogHandler.askBinaryQ("Do you want to take a picture again?", lambda: takePhoto(robot\_IP), no)

    asr.unsubscribe("WordRecognized")

    tts.say("It was great! Have a nice day!")

    alife.setState("solitary")

def no():

    global loop

    loop = False

def takePhoto(ip, resol = 4, picFormat = "jpg"):

    pic = getPhoto(ip, resol, picFormat)

    sender\_address = "tsinaolab@gmail.com"

    sender\_pass = "#TSIRiga2018!"

    # sender\_pass = "D9D9A675682877172EE275458E2E721407DB"

    receiver\_address=["nikolafgh@gmail.com", "nikolafgh@inbox.lv"]

    # cc = ["nikolafgh@inbox.lv", "drewzxcvbnm0@gmail.com"]

    resol = 4

    picFormat = "jpg"

    subject="Hello from NAO"

    mail\_content='''Hi, this is your photo'''

    onYes = lambda: sendEmail(\*\*preparedEmail(pic,picFormat, receiver\_address, subject, mail\_content, sender\_address, sender\_pass))

    # onNo = lambda: dialogHandler.askBinaryQ("Do you want to take a picture again?", lambda: takePhoto(ip), no)

    dialogHandler.askBinaryQ("Do you want to send to an email?", onYes, lambda: True)

def getPhoto(ip, resol = 4, picFormat = "jpg"):

    vision.setResolution(resol)

    vision.setPictureFormat(picFormat)

    tts.say("Free photo to everyone in 5 sec")

    time.sleep(1)

    tts.say("4")

    time.sleep(1)

    tts.say("3")

    time.sleep(1)

    tts.say("2")

    time.sleep(1)

    tts.say("1")

    anitext.post.say("^start(animations/Stand/Waiting/TakePicture\_1) Say cheese! ^wait(animations/Stand/Waiting/TakePicture\_1)")

    vision.takePicture("/opt/aldebaran/var/www/apps/", "image", True)

    s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

    s.connect((ip, 80))

    s.sendall(b'GET /apps/image.' + picFormat + b' HTTP/1.1\r\nHOST: ' + ip + '\r\n\r\n')

    reply = b''

    while select.select([s], [], [], 3)[0]:

        data = s.recv(2048)

        if not data: break

        reply += data

    headers =  reply.split(b'\r\n\r\n')[0]

    image = reply[len(headers)+4:]

    return image

def preparedEmail(image, picFormat, receiver\_address, subject, mail\_content, sender\_address, sender\_pass):

    tts.say("ok sending")

    message = MIMEMultipart()

    message['Subject'] = subject

    message['From'] = sender\_address

    message['To'] = ", ".join(receiver\_address)

    # message['Cc'] = ", ".join(cc)

    message.preamble = subject

    message.attach(MIMEText(mail\_content, "plain"))

    imge = MIMEImage(image, name = "pic.{}".format(picFormat))

    imge.add\_header('Content-Disposition', 'attachment; filename="pic.{}"'.format(picFormat))

    message.attach(imge)

    return {"receiver\_address":receiver\_address, "message":message, "sender\_address":sender\_address, "sender\_pass":sender\_pass}

def sendEmail(receiver\_address, message, sender\_address, sender\_pass):

    session = smtplib.SMTP('smtp.gmail.com', 587)

    session.starttls()

    session.login(sender\_address, sender\_pass)

    session.sendmail(sender\_address, receiver\_address, message.as\_string())

    tts.say("done")

    session.quit()

# ip = "192.168.253.155"

ip = "192.168.252.62"

makeAndSendPhotoToEmail(    robot\_IP = ip    )