

# 高齢者との弾む会話を行うソーシャルロボットの実現方式

平山秀昭<sup>1,a)</sup>

日本の総人口は、2018年4月1日現在で1億2,653万人、後期高齢者(75才以上の高齢者)人口は1,776万人と14.0%に及んでいる。仮に後期高齢者人口における男女の人数差が一人暮らしの後期高齢者人口に相当するとすれば、384万人で総人口の3.0%を占めることになる。一人暮らしの後期高齢者は、どうしても会話の機会が少なくなり、物忘れ等の老化が進むだけでなく寂しい生活をおくることになってしまう。一方、近年、人工知能やロボットの研究が進み、高齢者との会話を行うソーシャルロボットの活躍が期待されている。これまでに高齢者との弾む会話を行うソーシャルロボットの実現を目指した予備的検討、具体的には、特定対象者での観察に基づき、高齢者との弾む会話を行うソーシャルロボットの実現ポリシーを検討してきた。本稿では、そのポリシーを実現できるロボット(プラットフォーム)を選定し、実現方式を検討したので報告する。

## 1. はじめに

日本の総人口は、2018年4月1日現在で1億2,653万人、後期高齢者(75才以上の高齢者)人口は1,776万人と14.0%に及んでいる。また、後期高齢者人口における男女比は、男性696万人、女性1,080万人で1:1.55となっている[1]。仮に後期高齢者人口における男女の人数差が一人暮らしの後期高齢者人口に相当するとすれば、384万人で総人口の3.0%を占めることになる。一人暮らしの後期高齢者は、どうしても会話の機会が少なくなり、物忘れ等の老化が進むだけでなく寂しい生活をおくることになってしまう。

一方、経済産業省の新産業構造ビジョン[2]では、IoT、ビッグデータ、人工知能、ロボットといった第4次産業革命技術の社会実装が鍵だとされ、それにより人間中心で課題解決型の新たな社会(Society 5.0)を形成することをビジョンとしている。近年の高齢化社会に注目するなら、人工知能やロボットの研究が進み、高齢者との会話を行うソーシャルロボットの活躍が期待され、介護施設でロボットと会話をしたり、介助を受けたりすることへの利用意識は、高齢化が進む日本では特に高く、「とても利用したい」「どちらかといえば利用したい」の合計が59%となっている[3][4]。

しかし、現在のロボット・AIの技術で、一人暮らしの後期高齢者がコミュニケーションロボットと「弾む会話」ができるかという点で懐疑的である。その理由は、現状の自然言語処理分野における対話技術が十分に進展していないからである。また、仮に対話技術が十分に進展したとしても、人間は感情のない人工物との会話に心を動かさない。例えば、スナックを提供する自律ロボットをオフィスに設置した実験において、人間はロボットに対して強い命令口調で話したり、ロボットの発声を遮ってしまうという実験結果が示されている[5]。また、自律ロボットと遠隔操作ロボットとの対話実験において、人間は自律ロボットの挨拶を無視する傾向にあることが示されている[6]。このように人間

はロボットを社会的存在と見なさない傾向がある。その要因として、人間同士のコミュニケーションでは非言語表現が活用されていることが指摘されている、特に顔の表情は全体の約55%を占めると言われている[7]。

これまでに、高齢者との弾む会話を行うソーシャルロボットの実現を目指した予備的検討を行った[8]。そこでは、ポリシーを実現できるロボット(プラットフォーム)を選定すること、及び、そのロボットでポリシーを実現することが次の課題であるとしていた。本稿は、その課題設定に基づき高齢者との弾む会話を行うソーシャルロボットの実現方式について示すものである。なお、本研究はソフトバンクロボティクスのPepper[9]を活用しているが、それは目白大学メディア学部メディア学科平山秀昭研究室が独自に実施しているものである。

## 2. 特定対象者での観察

これまでの予備的検討[8]においては、高齢者との弾む会話を行うソーシャルロボットの実現を目指した特定対象者での観察を行った結果を示した。観察対象者の情報は、表1の通りである。2015年3月から2018年8月の3年6ヶ月に渡り、1週間に一度、約1時間の直接対話による観察を行なった結果である。当初2年間は、記憶力が正常だった。その後、病气入院し記憶力が低下した時期が1年間続いた。しかし、その後の6ヶ月間は、改善が見られている。

表1 観察対象者の情報

年齢	90才代
性別	女性
生活場所	リハビリホーム
観察期間	3年6ヶ月(2015年3月~2018年8月)
観察頻度	1回/週
観察時間	1時間/回

観察結果は以下の通りである。

1 目白大学メディア学部メディア学科  
Department of Media Studies, Faculty of Media Studies, Mejiro University,  
Shinjuku, Tokyo, 161-8539, Japan.  
a) h.hirayama@mejiro.ac.jp

- ① テレビは主な情報の入手元である。ただし内容を会話する相手がいないと次第に関心が薄れていく。病気入院前はよくテレビを見ていたが、病気入院後、食事時間の際に着席する場所が変わり、食事時間に会話する相手がなくなると関心が薄れていった。情報は入手するだけでなく、会話することで関心を持ち続ける。
- ② 家族への関心は高いが、情報入手がされなくなると記憶が薄れていく。病気入院後、孫のことを忘れるという状況が見られたが、会話をすると関心を示してくる。
- ③ 忘れてしまった孫の記憶も、毎週繰り返し会話をしていくことで思い出してくる。ただし、一気に記憶が甦るのではなく、思い出したり、忘れてたりすることを繰り返しながら少しずつ改善していく。
- ④ 記憶力が低下しても同じ話を何度も聞かされると覚える。例えば、病気入院後、孫のことを忘れているという状況であっても、羽入結弦選手が金メダルを取ったことは記憶していた（金メダル取得の翌日）。
- ⑤ 数種類の定型文を喋るだけのソーシャルロボット（ペット型の癒しロボット）には興味を示さない。これは、病気入院後、孫のことを忘れているという状況においてもである。単に、ぬいぐるみが録音された言葉を再生しているだけだと認識している。定型文を喋るだけでは愛着は湧かない。

のみが主体となるのではなく、双方が交互に主体となるようにする。

- ④ 同じ話を何度もすることで、記憶が定着するようにする。
- ⑤ ソーシャルロボットに対する愛着が湧くような人間らしい会話や仕草を試みる。
- ⑥ 家族等に関する特定の記憶を低下させないことで、生活に楽しみをもたらす、対象者及び家族双方の会話へのモチベーションを高める。

「関心の高い家族等に関する情報を入手し、入手した情報で会話を行う」、「家族等に関する新しい情報を入手し続けることで、関心と記憶力の低下を防止する」という実現ポリシーは、2章で示した観察結果の①②と関連している。「会話はソーシャルロボットと対象者のどちらか一方のみが主体となるのではなく、双方が交互に主体となるようにする」、「同じ話を何度もすることで、記憶が定着するようにする」という実現ポリシーは、観察結果の③④と関連している。「ソーシャルロボットに対する愛着が湧くような人間らしい会話や仕草を試みる」という実現ポリシーは、観察結果⑤と関連している。「家族等に関する特定の記憶を低下させないことで、生活に楽しみをもたらす、対象者及び家族双方の会話へのモチベーションを高める」という実現ポリシーは、観察結果全般と関連している。

### 3. 観察結果に基づく実現ポリシー

これまでの予備的検討[8]において、特定対象者での観察に基づき、高齢者との弾む会話を行うソーシャルロボット実現の基本的な考え方として、ソーシャルロボットの対話技術を向上させるという方向ではなく、高齢者が知りたいと思う話題で会話を行うという課題解決アプローチをとることにした。勿論、ソーシャルロボットの対話技術が向上することにこしたことはないが、それよりも本研究では会話の内容に着目した。本研究でいう「高齢者との弾む会話」とは、「高齢者が強い関心を持つ話題での会話」とする。また、人間は感情のない人工物との会話に心を動かさないとされるが、感情のないソーシャルロボットに対して愛着を湧かせることも会話を弾ませるための一助になると考えた。上記の基本的な考え方に基づくソーシャルロボットの実現ポリシーを以下のように定めた。

- ① 関心の高い家族等に関する情報を入手し、入手した情報で会話を行う。
- ② 家族等に関する新しい情報を入手し続けることで、関心と記憶力の低下を防止する。
- ③ 会話はソーシャルロボットと対象者のどちらか一方

### 4. ソーシャルロボットの实现要件

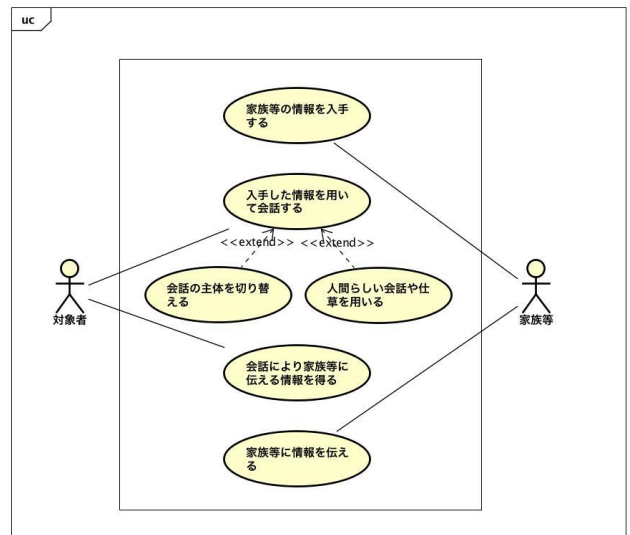


図1 ソーシャルロボットのユースケース

3章で示したポリシーを実現するソーシャルロボットのユースケースを図1に示す。ユースケースは、家族等の情報を入手するユースケース、入手した情報を用いて会話するユースケース、会話により家族等に伝える情報を得るユ

ユースケース、家族等に情報を伝えるユースケースからなる。また、会話の主体を切り替えるユースケース、人間らしい会話や仕草を用いるユースケースは、入手した情報を用いて会話するユースケースを拡張している。

家族等の情報を入手するユースケース、入手した情報を用いて会話するユースケース、会話により家族等に伝える情報を得るユースケース、家族等に情報を伝えるユースケースは、3章で示した実現ポリシーの①②と関連している。会話の主体を切り替えるユースケースは、実現ポリシー③④と関連している。人間らしい会話や仕草を用いるユースケースは、実現ポリシー⑤と関連している。また、ユースケース全体を通して、実現ポリシー⑥と関連している。

表2は、図1のユースケースにおいて家族等から入手した情報を蓄積するデータベースの構成を示す。データベースは、5W1Hのフィールドを持つ。“who”は、それを誰がしたのかを示す。“when”は、それをいつした(する)のかを示す。“where”は、それをどこでしたのかを示す。“what”は、何をした(する)のかを示す。“how”は、それをどうやって(どんなふう)にしたのかを示す。“why”は、なぜそれをしたのかを示す。“whom”は、その情報を誰から聞いたのかを示す。

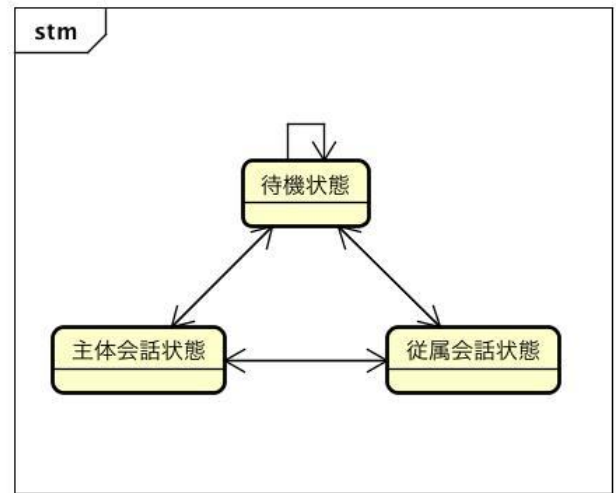
表2 データベースの構成

フィールド	概要
“who”	誰がしたのか
“when”	いつしたのか
“where”	どこでしたのか
“what”	何をしたのか
“how”	どうやってしたのか(どんなふうにしたのか)
“why”	なぜしたのか
“whom”	誰から聞いたのか

情報を表2に示す構成で持つのは、会話の主体を切り替えるユースケース、人間らしい会話や仕草を用いるユースケースと関連している。例えば、家族等から入手した情報で不足した部分を対象者に聞くことで会話の主体を切り替える。あるいは、後日、「あれはどこでしたのか?」といったことを忘れたかの如く質問することで人間らしい会話に近付けることができる。

会話の主体を切り替えるユースケースにあるように、会話は一方のみが主体となるのではなく、双方が主体となるようにする。図2は、ソーシャルロボットのステートマシン図である。ロボットは待機状態、主体会話状態、従属会話状態を持つ。待機状態は、会話をしていない状態である。主体会話状態は、ロボットが主体となって会話をしている状態である。従属会話状態は、対象者が主体となって会話している状態である。状態遷移においては、以下のような

点を考慮する。



powered by Astah

図2 ソーシャルロボットのステートマシン

(1) 待機状態→主体会話状態

待機状態から主体会話状態に遷移するトリガーは、ロボットが対象者に会話を始めることである。ロボットからの対話の内容はデータベースに記憶された5W1Hの情報に基づく。単純に家族等からの情報を提供するのではなく、5W1Hに基づく問い合わせ形式にすることで人間らしい会話に近付ける。

(2) 待機状態→従属会話状態

待機状態から従属会話状態に遷移するトリガーは、対象者がロボットに会話を始めることである。ただ単に対象者からの会話を待つのではなく、家族の新しい情報があることを示す人間らしい合図(鼻歌を歌う、手招きをする等)をすることが考えられる。

(3) 主体会話状態→従属会話状態

ロボットから対象者に話し続けるだけでなく、途中から従属会話状態に遷移させることも考えられる。なるべく対象者からの会話が行われる方が好ましい。そのためにデータベースにある5W1H1の情報に基づき、対象者が聞いてくるのを待つようにすることも考えられる。

(4) 従属会話状態→主体会話状態

なるべく対象者からの会話が行われる方が好ましいが、対象者からの会話だけになってしまうのも好ましくない。ロボットが単なる情報検索機械となってしまう、会話とは言い難くなるからである。

## 5. ロボット（プラットフォーム）の選定

4章で示した実現要件に基づきロボット（プラットフォーム）の選定を実施した。選定のための評価項目は以下の通りである。「インターネットと接続できる」及び「家族とコミュニケーションが取れる」という評価項目は、4章で示した実現要件の家族等の情報を入手するユースケース、家族等に情報を伝えるユースケースと関連している。「情報を整理して記憶できる」及び「主体・従属等会話状態を変えられる」という評価項目は、会話の主体を切り替えるユースケース、人間らしい会話や仕草を用いるユースケースと関連している。「上記を制御するためのプログラミングが行える」という評価項目は、ユースケース全体と関連している。「人間らしい仕草を表現できる」という評価項目は、人間らしい会話や仕草を用いるユースケースと関連している。

- インターネットと接続できる
- 家族とコミュニケーションがとれる
- 情報を整理して記憶できる
- 主体・従属等会話状態を変えられる
- 上記を制御するためのプログラミングが行える
- 人間らしい仕草を表現できる

選定の候補として「高齢者・老人向けの話し相手・会話ロボットまとめ」[10]にあるロボット（プラットフォーム）の中で会話力、値段が共に高い以下を候補とした。

- Tapia (MJI)
- unibo (ユニロボット)
- Robi (タカラトミー)
- RoBoHon (シャープ)
- Pepper (ソフトバンクロボティクス)
- Palmi (DMM.com)

上記候補の中から、ビジュアルなツール等による簡易的なプログラミングだけでなく、本格的なプログラミングが可能なPepperとPalmiに絞った。更に人間らしい仕草を表現できるという観点からPepperを選択した。Pepperが各評価項目をどのように満たしたかを以下に示す。

- インターネットと接続できる  
Pepperはインターネットと接続されていてPythonでプログラミングすることが可能である。
- 家族とコミュニケーションがとれる  
Pepperはインターネットと接続されていて、かつ、プログラミングが可能なことで、SNSを用いて家族とコミュニケーションをとることが可能である。

- 情報を整理して記憶できる  
PepperはデータベースのSQLiteを持ち、情報を整理して記憶することが可能である。
- 主体・従属等会話状態を変えられる  
Pepperにはオートノマスライフという概念があり、外部からの割り込みが可能なソリタリモードと割り込みを受け付けないインタラクティブモードを持つ等、会話状態を変えることが可能である。
- 上記を制御するためのプログラミングが行える  
PepperはChoregrapheというビジュアルなプログラミングツールに加えて、プログラムの一部をPythonで記述することが可能である。
- 人間らしい仕草を表現できる  
Pepperは容姿が人間に近い上、移動したり、身体を動かしたりすることをプログラミングでき、人間らしい仕草を表現することが可能である。

## 6. ソーシャルロボットの实现方式

4章で示した要件を満たすソーシャルロボットを実現するため、5章においてロボット（プラットフォーム）の選定を行った。本章では、選定したロボット（プラットフォーム）上での实现方式を示す。实现方式の要となるのは、以下の2方式である。

- SNSを介した家族との間接対話方式
- 人間らしい記憶に基づく主体会話・従属会話切替方式

以下に各方式について4章で示した实现要件と関連付けながら説明していく。なお、ここではSNSとしてTwitterを用いている。

### 6.1 SNSを介した家族との間接対話方式

一人暮らしの後期高齢者が家族と直接コミュニケーションをとることは難しい。家族が通常利用しているSNSを介してコミュニケーションを取れるといいのだが、一人暮らしの後期高齢者にとっては容易なことではない。そのためロボットがSNSからメッセージを取り出し後期高齢者に音声で伝える。また、後期高齢者の反応を聞き、ロボットがSNSにメッセージを送る方式を提案する。この部分は、図1の家族等の情報を入手するユースケース、入手した情報を用いて会話するユースケース、会話により家族等に伝える情報を得るユースケース、家族等に情報を伝えるユースケースと関連していて、高齢者が知りたいと思う話題で会話を行うという方向を目指している。以下では、ロボットがSNSからメッセージを取得する基本的な仕組、及び、ロボットがSNSにメッセージを送る基本的な仕組を示す。

SNS からメッセージを取得する仕組の概略フローを図 3 に示す。左から右に 5 個のボックスを順番に処理していく。一番左のボックスでは、言語設定（日本語）をしている。二番目のボックスでは、「家族からメッセージが届いているか見てみますね。」といった言葉を会話相手に発している。三番目のボックスでは、SNS からメッセージを取得する際のキーワードを設定している（ここでは、「おばあちゃん」と設定）。四番目のボックスでは、実際に SNS からメッセージを取得する処理を行っている。この処理のアルゴリズムを図 4 に示す。五番目のボックスでは、SNS から取得したメッセージを会話相手に発している。

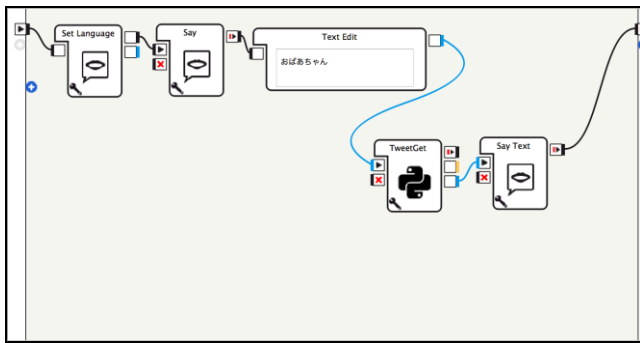


図 3 SNS からメッセージを取得する仕組の概略フロー

図 4 は、SNS からメッセージを取得する処理のアルゴリズムを示した図である。①は開始点で、メッセージを取得する際のキーワードが、ボックスの入力端子から変数 p に渡されている。②では Twitter の WebAPI にアクセスするための URL を設定している。③は Twitter の WebAPI を呼び出す際のパラメータを設定している。変数 Counts は、取得するメッセージ数を表す変数で、ボックスにマニュアルで設定できるようになっている。④は Twitter の WebAPI を呼び出す際に必要な認証処理である。⑤では、Twitter の WebAPI を実際に呼び出し、取得したメッセージを変数 acctext に詰めている。⑥では変数 acctext に詰めたメッセージをボックスの出力端子 tweeted に出力している。⑦でボックスの処理を終了させている。

次に、SNS にメッセージを出力する仕組の概略フローを図 5 に示す。左から右に 5 個のボックスを順番に処理していく。一番左のボックスでは、言語設定（日本語）をしている。二番目のボックスでは、「どうですか？お返事して下さいね。」といった言葉を会話相手に発している。三番目のボックスでは、会話相手によって発話されるメッセージを聞いている。四番目のボックスで、「すごいね」、「よかったね」、「元気かい」、「元気だよ」、「嬉しいよ」、「ありがとう」と行ったメッセージを認識し、五番目のボックスで、そのメッセージを SNS に出力している（特定のメッセージに限定しているのは、仕組の概略を示すためである）。この処理のアルゴリズムを図 6 に示す。

```

class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        self.framemanager = ALProxy('ALFrameManager')
        self.folderName = None

    def onUnload(self):
        import sys
        if self.folderName and self.folderName in sys.path:
            sys.path.remove(self.folderName)
        self.folderName = None

# 開始 ←①
def onInput_onStart(self, p):
    import sys, os
    self.folderName = os.path.join(
        self.framemanager.getBehaviorPath(self.behaviorId),
        '../lib')
    if self.folderName not in sys.path:
        sys.path.append(self.folderName)
    for moduleName in os.listdir(self.folderName):
        if moduleName in sys.modules:
            self.logger.info('Loaded: %s, %s' % (moduleName,
                sys.modules[moduleName].__file__))
            reload(sys.modules[moduleName])
    from requests_oauthlib import OAuth1Session

# URLの設定 ←②
url = 'https://api.twitter.com/1.1/search/tweets.json'

# パラメータの設定 ←③
params = {'q': p, 'count': self.getParameter('Counts')}

# 認証の実施 ←④
twitter = OAuth1Session(self.getParameter('Consumer Key'),
    self.getParameter('Consumer Secret'),
    self.getParameter('Access Token'),
    self.getParameter('Access Token Secret'))

# ツイートの取得 ←⑤
req = twitter.get(url, params = params)
if req.status_code == 200:
    timeline = json.loads(req.text)
    acctext = 'キーワードは '+p+' です。'+ '読みますよ。'
    first = True
    for tweet in timeline['statuses']:
        if first:
            first = False
        else:
            acctext = acctext + '. 次は。'
        self.logger.info(
            tweet['user']['name'].encode('utf-8')+
            ':'+tweet['text'].encode('utf-8')+
            ':'+tweet['created_at'].encode('utf-8'))
        acctext = acctext + tweet['text'].encode('utf-8')
    acctext = acctext + '. これでおしまいです。'
    self.logger.info(acctext)

# 取得ツイートを「tweeted」に出力 ←⑥
self.tweeted(acctext)

# 終了 ←⑦
self.onStopped()
else:
    self.logger.warn('ERROR: %d' % req.status_code)
    self.onFailed(req.status_code)

def onInput_onStop(self):
    self.onUnload()
    self.onStopped()
    
```

図 4 SNS からメッセージを取得する処理のアルゴリズム

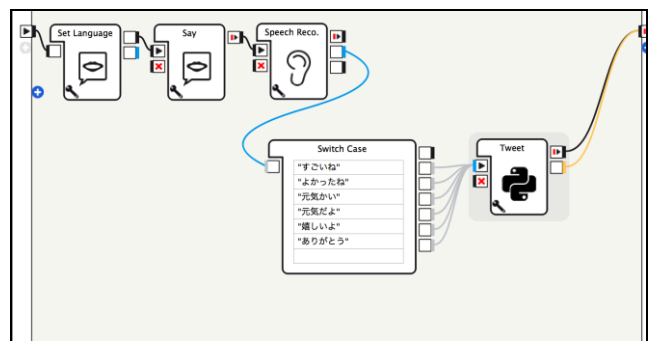


図 5 SNS にメッセージを出力する仕組の概略フロー

```

class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        self.framemanager = ALProxy('ALFrameManager')
        self.folderName = None

    def onUnload(self):
        import sys
        if self.folderName and self.folderName in sys.path:
            sys.path.remove(self.folderName)
        self.folderName = None

# 開始 ←①
def onInput_onStart(self, p):
    import sys, os
    self.folderName = os.path.join(
        self.framemanager.getBehaviorPath(self.behaviorId),
        '../lib')
    if self.folderName not in sys.path:
        sys.path.append(self.folderName)
    for moduleName in os.listdir(self.folderName):
        if moduleName in sys.modules:
            self.logger.info('Loaded: %s, %s' % (moduleName,
                sys.modules[moduleName].__file__))
            reload(sys.modules[moduleName])
    from requests_oauthlib import OAuth1Session

# URLの設定 ←②
url = 'https://api.twitter.com/1.1/statuses/update.json'

# パラメータの設定 ←③
params = ['status': p+' おばあちゃんより。']

# 認証の実施 ←④
twitter = OAuth1Session(self.getParameter('Consumer Key'),
    self.getParameter('Consumer Secret'),
    self.getParameter('Access Token'),
    self.getParameter('Access Token Secret'))

# ツイート ←⑤
req = twitter.post(url, params = params)

if req.status_code == 200:
    self.logger.info('OK')

# 終了 ←⑥
self.onStopped()
else:
    self.logger.warn('Failed: %d' % req.status_code)
    self.onFailed(req.status_code)

def onInput_onStop(self):
    self.onUnload()
    self.onStopped()
    
```

図6 SNSにメッセージを出力する処理のアルゴリズム

図6は、SNSにメッセージを出力する処理のアルゴリズムを示した図である。①は開始点で、出力するメッセージが、ボックスの入力端子から変数pに渡されている。②ではTwitterのWebAPIにアクセスするためのURLを設定している。③はTwitterのWebAPIを呼び出す際のパラメータに、出力するメッセージを設定している。④はTwitterのWebAPIを呼び出す際に必要な認証処理である。⑤では、TwitterのWebAPIを実際に呼び出し、メッセージを出力している。⑥でボックスの処理を終了させている。

## 6.2 人間らしい記憶に基づく主体会話・従属会話切替方式

SNSから取得したメッセージを単に対象者に伝えるだけでなく、一部の付加情報を対象者から教えて貰い、それもあわせて記憶する。また、ロボットが対象者から教えて貰った記憶を人間のように忘れてしまい、「あれ、なんでしたっけ?」というように問い合わせる方式を提案する。この部分は、図1の会話の主体を切り替えるユースケース、人間らしい会話や仕草を用いるユースケースと関連していて、感情のないロボットに対して愛着を湧かせるという方向を目指している。教えて貰って記憶したことを忘れてしまったり、再び思い出したりする人間らしい記憶を実現するためにデータベースに2つのフィールドを追加する。一つは、対象者に教えて貰ったことを示す learned というフィールドである。もう一つは、対象者に教えて貰ったことを忘れてしまったことを示す forgotten というフィールドである。以下に、忘れてしまった記憶を会対象者に問い合わせる基本的な仕組みを示す。

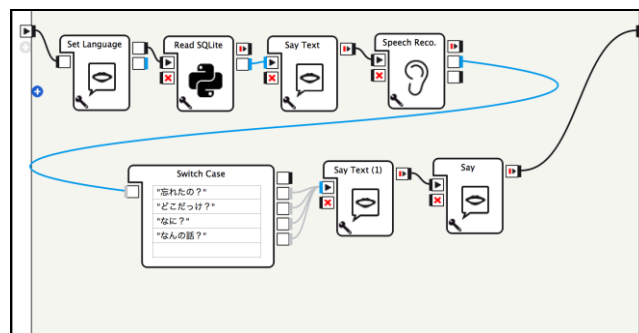


図7 会話相手に教えて貰ったが、忘れてしまったことを尋ねる仕組みの概略フロー

対象者に教えて貰ったが、忘れてしまったことを尋ねる仕組みの概略フローを図7に示す。左から右に7個のボックスを順番に処理していく。一番左のボックスでは、言語設定（日本語）をしている。二番目のボックスでは、人間らしい記憶に基づく会話の処理を行っている。この処理のアルゴリズムは、図8に示す。三番目のボックスでは、忘れてしまったことを対象者に尋ねている。四番目のボックスでは、対象者からの返事を聞いている。五番目のボックスで、対象者からの返事を認識し、六番目ボックスで、その返事を復唱している。七番目のボックスでは、思い出したことを対象者に伝えている。

図8は、人間らしい記憶に基づく会話の処理のアルゴリズムを示した図である。①は開始点で、ボックスに設定されたデータベース名、テーブル名を設定している。②では、データベースへの接続を行っている。③では、テーブルが存在することを確認している。④では、対象者に教えて貰ったのに、忘れてしまった記憶を探している。データベー

スの5W1Hのフィールドのうち、who、when、whatのフィールドはSNSから取得する。それ以外のwhere、how、whyのフィールドは対象者から聞く。対象者から聞いたことを示すのがlearnedというフィールドである。また、忘れてしまったことを示すのがforgottenというフィールドである。不定期、ランダムにforgottenのフィールドをセットし、対象者から聞いたことを忘れてしまったことを模擬する。ここでは、learnedとforgottenが共にセットされたレコードを探している。この忘れてしまった記憶を次のボックス以降で対象者に聞き直すためacctestに設定している。⑤では、全てのレコードのforgottenのフィールドをリセットし、忘れていたことを思い出すことを模擬している。⑥で忘れてしまった記憶がなかった場合の処理を行い、⑦で処理を終了させている。

## 7. 関連研究

高齢者は他の家族から孤立し易く、離れた家族との遠隔コミュニケーションは重要な意味を持つ。しかし、世代による生活時間帯の違いや利用可能なコミュニケーション技術の違いにより、家族間での円滑なコミュニケーションは難しいと考え、[11]は世代の異なる家族間コミュニケーションを促進することを目的にシェアエージェントシステムを提案している。本研究における「SNSを介した家族との間接対話方式」は、[11]と近い課題解決アプローチをとっていると言える。

[12]では更に、シェアエージェントシステムの祖父母世代向けインタフェースとしてロボットを試作し、それを実際に使って貰い予備実験を行なっている。高齢者とのインタフェースにロボットを用いている点も、本研究における「SNSを介した家族との間接対話方式」に近いアプローチをとっていると言える。

[13]は、ロボットを用いたコミュニケーションによる高齢者向けの対話インタフェースを提案している。対話インタフェースを持つロボットの問題として、対話コンテンツに限りがあり、ロボットとの対話に飽きてしまうことが挙げられる。[14]は、他者の対話履歴を利用したクラウド型の雑談ロボットを提案している。[15]は、ロボットとのコミュニケーションにテーマを用意するのではなく、テーマを定めない雑談を提案し実験を行なっている。長期的な利用に向けた実験では、雑談内容のカテゴリによって話し易い／話し難いものがあることが示されている。本研究は対話コンテンツとして、対象者の家族等に関する情報を扱い、更にその情報をSNSから入手し続けることで、ロボットとの対話に飽きてしまうという問題を解決する。

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #put initialization code here
        pass

    def onUnload(self):
        #put clean-up code here
        pass

# 開始 ←①
def onInput_onStart(self):
    import sqlite3
    dbPath = self.getParameter('DB Path')
    dbFile = self.getParameter('DB File')
    dbPathFile = dbPath + dbFile
    tableName = self.getParameter('Table')

    # DBへの接続 ←②
    try:
        conn = sqlite3.connect(dbPathFile)
        cur = conn.cursor()
    except Exception as e:
        self.logger.info('connect error: ' + e.message)
        self.onStopped()

    # テーブルの存在確認 ←③
    try:
        sqlCheckTable = 'select count(*) from sqlite_master where
type="table" and name="' + tableName + '"'
        result = cur.execute(sqlCheckTable).fetchall()
        if(result[0][0] == 0):
            self.logger.info('table not exist')
            conn.close()
            self.onStopped()
    except Exception as e:
        self.logger.info('fetch error: ' + e.message)
        self.onStopped()

    # 教えて貰ったのに、忘れてしまった記憶を探す ←④

    sqlSelect = 'select * from ' + tableName + ' where learned = 1
and forgotten = 1'
    list = cur.execute(sqlSelect).fetchall()
    first = True
    for row in list:
        if first:
            first = False
            acctest = '忘れてしまったので、教えてくださいか?'
            acctest += str(row[1].encode('utf-8')) + 'に'
            acctest += str(row[0].encode('utf-8')) + 'さんが言った
ことです。'
            acctest += str(row[2].encode('utf-8')) + '。これ、どこ
で話でしたっけ?'

        # 忘れてしまった記憶を思い出す ←⑤
        self.logger.info('update')
        sqlUpdate = 'update ' + tableName + ' set forgotten = 0'
        cur.execute(sqlUpdate).fetchall()
        conn.commit()

    # 忘れてしまった記憶がなかった場合 ←⑥
    if first:
        acctest = 'あれ、あれなんだったっけ?'
        self.logger.info(acctest)
        self.onAsk(acctest)

    # 終了 ←⑦
    conn.close()
    self.onStopped()

def onInput_onStop(self):
    self.onUnload() #it is recommended to reuse the clean-up as the
box is stopped
    self.onStopped() #activate the output of the box
```

図8 人間らしい記憶に基づく会話の処理のアルゴリズム

施設に入居した認知症高齢者は、十分に他者とのコミュニケーションが取れなく脳をあまり使わないことで認知症が進行する恐れがあり、テレビ電話による会話が期待される。[16]は、遠隔操作型ロボット(テレノイド)による実験の結果、高齢者はテレノイドに初見から関心を示し愛着を持つことを報告している。[17]は、テレノイドを介した対話と直接対面しての対話とを比較し、テレノイドの有用性について検討し中等度の認知症の高齢者ではテレノイドの方が良い印象であることを報告している。また、[18]は、

ロボットに「人らしさ」を感じるかどうかのフィールド実験を行なっている。スーパーマーケットで高齢者の買い物を支援するロボット（荷物を運ぶカート型ロボットと会話をする人型ロボット）の実験において、特に人型ロボットに対して「誰かと一緒に買い物をしているようで楽しい」と感じられたという結果が得られている。また、デイケアセンターでの実験において、毎回、訪問者に挨拶し、ちょっとした雑談をするロボットに対して「いつも挨拶してくれて元気がでる」、「口答えしないので可愛い」と感じられたという結果が得られている。更に興味深いのは、実験期間の最後にロボットのお別れ会を開き、寄せ書きを送った、後でロボットに会いにきた人もいたという事実である。本研究では、感情のないロボットに対して愛着を湧かせるという方向を目指しているが、[16][17][18]はそれが可能であることを示唆していると言える。

## 8. おわりに

これまでの予備的検討[8]において、特定対象者での観察に基づき、高齢者との弾む会話を行うソーシャルロボットの実現ポリシーについて検討した。本研究における高齢者との弾む会話を行うソーシャルロボット実現の基本的な考え方は、ソーシャルロボットの対話技術を向上させるという方向ではなく、高齢者が知りたいと思う話題で会話を行うという課題解決アプローチをとっている。すなわち「高齢者との弾む会話」とは、「高齢者が強い関心を持つ話題での会話」である。また、人間は感情のない人工物との会話に心を動かさないとされるが、感情のないソーシャルロボットに対して愛着を湧かせることも会話を弾ませるための一助となる。本稿では、そのポリシーを実現できるロボット（プラットフォーム）を選定し、実現方式を検討した。今後は本実現方式に基づく試作を行い、実際の施設等での実験・評価を行う必要がある。実際の施設等で実験・評価を行わせて貰うのは容易なことではない。しかし、冒頭でも記載した通り、日本の総人口は、2018年4月1日現在で1億2,653万人、後期高齢者人口は1,776万人と14.0%に及んでいる。この現状に鑑みると本研究は超高齢化社会を迎えている日本の喫緊の課題への取り組みであり、その意義は大きいと考えている。

## 参考文献

- 1) “人口推計 -平成30年4月報-”，総務省統計局，平成30年4月。
- 2) “新産業構造ビジョン”，経済産業省産業構造審議会新産業構造部会，平成29年5月。
- 3) “ロボット・AIに関する日・米・独インターネット調査”，野村総合研究所，2015年11月。
- 4) “ロボット・AI技術の導入をめぐる生活者の受容性と課題”，日戸浩之，谷山大介，稲垣仁美，知的資産創造，2016年5月号。

- 5) “Ripple effects of an embedded social agent: a field study of an social robot in the workplace”，Min Kyung Lee, Sara Kiesler, Jodi Forlizzi, Paul Rybski, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 695-704, May, 2012.
- 6) “Teleoperated or Autonomous?: How to Produce a Robot Operator’s Pseudo Presence in HRI”，Kazuaki Tanaka, Naomi Yamashita, Hideyuki Nakanishi, Hiroshi Ishiguro, Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction, pp. 133-140, March, 2016.
- 7) “非言語コミュニケーション”，A. マレービアン(著)，西田司，津田幸男，岡本輝人，山口常夫(訳)，1986.
- 8) “高齢者との弾む会話を行うソーシャルロボットの実現を目指した予備的検討”，平山秀昭，マルチメディア通信，分散，協調とモバイルシンポジウム (DICOM02018)，pp. 1693-1697，2018年7月。
- 9) “Pepper ドキュメント・SDK ダウンロード”，<https://www.softbank.jp/robot/developer/dev-support/docume.doc>.
- 10) “【介護】高齢者・老人向けの話し相手・会話ロボットまとめ”，<https://会話ロボットまとめ.com>.
- 11) “A Shared-Agent System for Encouraging Remote Communication over Three Generations: The First Prototype”，Yohei Noguchi, Fumihide Tanaka, Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pp. 229-230, March, 2017.
- 12) “ロボットを介した遠隔コミュニケーションにおける高齢者の事故開示の調査”，野口洋平，田中文英，2017年度人工知能学会全国大会（第31回），2017年5月。
- 13) “高齢者向け対話インタフェース -雑談継続を目的とした話題提示・傾聴の切替式対話法-”，横山祥恵，山本大介，小林優佳，土井美和子，情報処理学会研究報告，vol. 2010-SLP-80，no. 4，2010年2月。
- 14) “A cloud based chat robot using dialogue histories for elderly people”，Eri Sato-Shimokawara, Shun Nomura, Yoko Shinoda, Haeyeon Lee, Tomoya Takatani, Kazuyoshi Wada, Toru Yamaguchi, Proceedings of the 24th IEEE International Symposium on Robot and Human Interactive Communication, pp. 206-210, August, 2015.
- 15) “高齢者と音声対話ロボットの雑談履歴の解析”，下川原(佐藤)英理，篠田遥子，李海妍，高谷智哉，和田一義，山口亨，日本ロボット学会誌，Vol. 34, No. 5, pp. 309-315, 2016年6月。
- 16) “Inconsistency of Personality Evaluation Caused by Appearance Gap in Robotic Telecommunication”，Kaiko Kuwamura, Takashi Minato, Shuichi Nishio, Hiroshi Ishiguro, Interaction Studies, Vol. 16, No. 2, pp. 249-271, January, 2015.
- 17) “認知症高齢者を対象としたロボットによる対話支援”，桑村海光，西尾修一，佐藤真一，2016年度人工知能学会全国大会（第30回），2016年6月。
- 18) “ロボットに「人らしさ」を感じる人々 -フィールド実験での事例-”，神田崇行，日本ロボット学会誌，Vol. 31, No. 9, pp. 860-863, 2013年11月。