

Pepperに体操させるアプリケーション開発のためのライブラリの提案

阿久津由嗣^{†1} 野田夏子^{†1}

近年、日常生活の場で人とコミュニケーションすることにより情報提供などのサービスを行う、コミュニケーションロボットへの注目が高まっている。コミュニケーションロボット Pepper は人間に近い上半身の構造を持っており、この特徴を活かしたアプリケーションとして介護現場で体操を教示させるといったものが考えられる。しかし現状のアプリケーション開発方法では Pepper の動きの作成・カスタマイズが容易にできないものとなっており、Pepper に体操の動きを実際に行わせる際にも体操の特性を考慮する必要がある。本稿では、介護現場のレクリエーション等で用いられるラジオ体操を Pepper に行わせるアプリケーションを簡単に開発・カスタマイズするためのライブラリを提案する。

Library for Gymnastic Exercises Application of Pepper

YUJI AKUTSU^{†1} NATSUKO NODA^{†1}

Pepper is one of the communication robots that have gotten attention recently. It has a human-like upper body and can act similar to human by using it. Applications such that let the Pepper do gymnastic exercises utilizing this feature are expected. In this paper, we introduce a library for gymnastic exercises application of Pepper that ease developing and customizing the applications.

1. はじめに

近年、日常生活の場で人とコミュニケーションすることにより情報提供などのサービスを行う、コミュニケーションロボットへの注目が高まっている。このようなコミュニケーションロボットのひとつとして、ソフトバンクロボティクス社の Pepper[1]があり、様々な場所での活用が試みられ始めている。Pepper は、人間に近い上半身の構造を持っており、その構造を用いることで人間に似た身振りさせることができるため、この特徴を生かした活用が考えられている。

その一つとして、介護現場で活用するために、Pepper にラジオ体操などの簡単な体操動作を教示として行わせ、被介護者の運動を促進させるといった使い方があられる。このような使い方においては、どの様な運動をさせるかが、介護現場毎に異なり得る、そのため、Pepper に行なわせたい体操を簡単にアプリケーションとして作成できること、またアプリケーションを簡単にカスタマイズして体操の内容(運動の強度、速度、動かす身体部位、繰り返し回数)を変更できることなどが求められる。

Pepper のアプリケーションを開発するために、開発環境 Choregraphe[2] が提供されている。しかし、Choregraphe の標準的な使用方法に則って、

Choregraphe が標準で備えているライブラリのみを使って様々な動きを開発するのは簡単ではなく[7]、開発されたアプリケーションをカスタマイズすることも簡単ではない。また、体操の動きを作成する際、自身の持つ部位同士の衝突や、体操の切り替えなどを考慮して作成する必要もある。

そこで、本稿では、ラジオ体操を Pepper に行わせるアプリケーションを簡単に開発し、また簡単にカスタマイズするためのライブラリを提案する。なお本稿では、いわゆるラジオ体操第 1 第 2[11]だけでなく、それと同程度の、リズムに合わせて行う簡単な徒手体操一般をラジオ体操と呼ぶことにする。第 2 章で Pepper にラジオ体操をさせるアプリケーションを作成する際の現状の課題について述べ、第 3 章で課題解決のために提案するライブラリについて解説する。第 4 章ではライブラリの評価及びそれを踏まえた考察を述べ、第 5 章で関連研究を提示し、第 6 章で結論と今後の課題を述べる。

2. 現状の課題

本章では Pepper の標準的なアプリケーション開発環境を紹介するとともに、それを用いてラジオ体操を行うアプリケーションを開発する際の課題について述べる。

^{†1} 芝浦工業大学
Sibaura Institute of Technology

本文中で使われているシステム・製品名は、一般に各社の商標または登録商標です。

2.1 アプリケーション開発環境 Choregraphe

Choregraphe はビジュアルプログラミング形式で Pepper のアプリケーション開発を行う環境である。Choregraphe ではアプリケーションの一連の振る舞いの基本構成要素であるボックスを接続していくことでアプリケーションを実装することを開発の基本としており[4]、様々な機能を持つボックスを、部品として再利用することができる。ボックスは図 1 のように接続され、接続された順序で振る舞いを行う。標準機能としてボックスライブラリにあらかじめ用意されているボックスのみならず、ユーザ自身が独自に定義したボックスを利用することもできる。

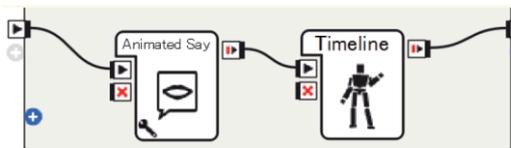


図 1 ボックス接続例

Pepper は人に近い振る舞いを行えるよう、全身にアクチュエータが配置されている[4]。各アクチュエータの値を変更することで Pepper の全身の関節部の位置を指定しポーズを作成する。

2.2 既存のモーション作成方法

標準のボックスライブラリでは Pepper の腕や腰などの動きを作成しアプリケーションに実装したい場合、Timeline というボックスを使う。Timeline ではパラパラ漫画のように何種類かキーとなる Pepper の全身のポーズを作成し、そのポーズを並べることで、ポーズの連続という形でモーションを作成する。この時、ポーズ間の動きは自動で補完される。図 2 に Timeline を用いたモーション作成例を示す。ポーズは図 2 の下にあるロボット画面上で各部位の関節を選択し、任意に関節の角度変更することで作成する。並べられたポーズの上部にある目盛は時間軸を示す。

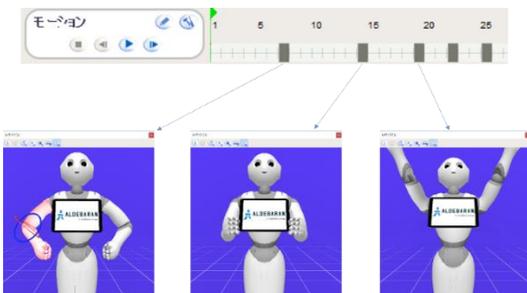


図 2 Timeline を用いたモーション作成例

2.3 ラジオ体操の特徴

ラジオ体操第 1 第 2[11]は体力向上と健康の保持や増進を目的とした一般向けの体操である。それぞれの体操は一定のリズムに合わせて運動を行い、複数回繰り返す。体操は様々なものがあるが、主に腕、腰、頭の動きを組み合わせて構成されている。体操は身体の左右をバランスよく運動させるために、左右対称に体操を行うが、腕を交差させる体操などの一部の体操は完全に対称に身体を動かす体操ではないものとなっている。本稿でのラジオ体操とは、このような特徴を持つ体操のことである。

2.4 ラジオ体操のモーション作成に生じる課題

ラジオ体操のモーションを作成する際、前述のモーション作成方法で生じる作成・カスタマイズのしにくさに関する課題、Pepper に体操を実際に行わせる際に考慮すべき課題についてそれぞれ述べる。

(a) モーションの作成・カスタマイズのしにくさ

ラジオ体操を Pepper に行わせるため、ラジオ体操の『腕を回す運動』といった 1 単位の体操の構成要素を分析したものを図 3 に示す。体操は決まった動きを繰り返すことで 1 つの体操としているため、1 つの体操を示す「体操動作」は「反復運動」を持つ。「反復運動」は例えば、『腕を振って足を曲げ伸ばす運動』であれば腕を横に振ってまた元に戻るという繰り返す動きのことを指している。「反復運動」では運動毎に身体を動かす部位が異なる、または組み合わせることから、腕・腰・頭といった部位毎の「反復運動」が存在する。また、『体を前後に曲げる運動』や『体を斜め下に曲げ、胸を反らす運動』といった、1 つの体操の中に、複数の小さな単位としての「体操動作」を持つ場合もある。

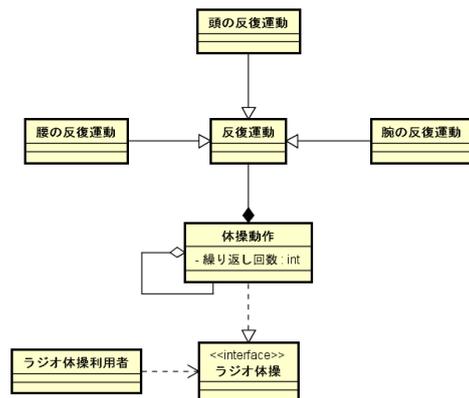


図 3 ラジオ体操の構成を分析したクラス図

この分析から、既存のモーション作成方法では、Pepper に行わせるラジオ体操のモーション作成におけるいくつかの問題点を指摘できる。主な問題点として、①ポーズの連続がどの「反復運動」を示しているか分かりづらいという点、②1 つの「反復運動」をカスタマイズするのにポーズ毎の編集を要求される点、③部位毎の「反復運動」として分割しづらい点、以上 3 点が挙げられる。これらの問題により、既存のモーション作成方法ではラジオ体操のモーションの作成やカスタマイズがしづらいため、手間や時間がかかる。

(b) Pepper に体操を行わせる際に考慮すべき課題

ラジオ体操では、腕を交差させるという動きがしばしば起こる。左右対称に行う体操(例:『腕を回す運動』、『腕を振り足を曲げ伸ばす運動』等)では、左右どちらか片方の腕のモーションをミラーリングしたものを再利用し、左右対称でモーションを作成することが望ましいが、腕を交差する動作があるため、そのまま左右対称に作成してしまうと、腕が衝突するような動作になってしまう。Pepper は自身の保護機能として、衝突すると考えられるモーションが入力されたとき、衝突を回避するためにモーションの連続を自動的に補正しづつからないように速度を落としゆっくりとした動きを行うが、体操を行う場合に、想定している動きと異なるような動きをしてしまうことは避けたいと考えられる。そのため、この腕が衝突してしまうことを回避するような機能を実装し、保護機能による動作の障害を防ぐ必要がある。

また、複数の体操で構成されているラジオ体操では、音楽の拍子の切り替わりによって今行っている体操の終わりから次に行う体操の最初に取るべき姿勢に移行し、スムーズに体操の切り替えを行わなければならない。Pepper のアプリケーションでは、この切り替えを伴った一連の体操動作を 1 つ 1 つの部品として分割することが考えられるが、その部品同士の継ぎ目である体操の切り替えをどのように実現するかを考慮する必要がある。

3. 提案するライブラリ

本章では、前章で述べた課題を解決するための提案について述べ、提案するライブラリの概要、使用方法を解説したのち、ライブラリの詳細について述べる。

3.1 課題解決のためのライブラリの提案

前章で述べたように、既存のモーション作成方法ではラジオ体操を作成することは可能でも、容易には作成できないものとなっている。これを解決するために、本稿ではラジオ体操を Pepper に行わせるアプリケーションを簡単に開発し、また簡単にカスタマイズするため

のライブラリを提案する。以下に課題に対するライブラリによる具体的な解決を示す。

(a) モーション作成・カスタマイズに対する解決

2.4(a)の分析に基づき、Pepper がラジオ体操の各体操動作を行う機能をボックスとしてライブラリを構成する。この部品は分析したラジオ体操の構成を基にしている。部品の構成を簡単に整理したものを図 4 に示す。

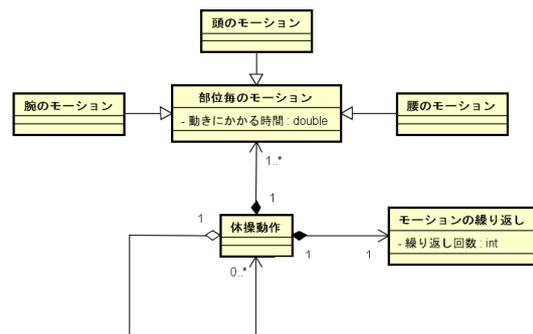


図 4 部品の構成を整理したクラス図

「反復運動」は「部位毎のモーション」に対応している。体操動作の属性である繰り返し回数は、Choregraphe のボックスの仕様上別の部品「モーションの繰り返し」として置き、体操動作はその部品を持つことで繰り返し回数を設定できる機能を持つようにしている。「体操動作」はこれら 2 種類の部品を持つようにし、Pepper に体操を行わせる機能を実装できるようにしている。この部品群により、体操の動きを部品別に細かく容易にカスタマイズ・管理することが可能になる。

(b) 体操を行わせる際考慮すべき課題に対する解決

2.4(b)で述べた、腕の交差による衝突を回避するための機能と、体操の切り替えの実現に対しては、本研究は以下のようなアプローチを通して解決を図る。

- (1) 衝突が予想される腕の交差する動きに当たる「腕のモーション」に対して、腕部の関節の 1 つである肩口の関節の角度を任意に設定可能にする。これにより、腕の交差が生じると考えられる腕を身体の内側に曲げて動く部分を、肩口の角度を変えた状態で腕を動かすことができ、腕の交差を不具合なく行えるようにできる。
- (2) スムーズな体操の切り替えを実現するため、組み合わせる「部位毎のモーション」の前後において、先に行う「部位毎のモーション」の最後を取る姿勢

を次の「部位毎のモーション」の最初に取るべき姿勢を取るように自動で補正する機能を持たせる。これにより、「部位毎のモーション」がそれぞれ固有の反復運動を持ちつつ、他の「部位毎のモーション」との互換性を保証できるようにできる。

3.2 ライブラリの概要

提案するライブラリの概要を述べる。部品は Choregraphe 上のボックスという形で存在しており、ラジオ体操の各体操はこの部品の組み合わせによって Pepper がその体操を行う機能を実現できる。また、部品毎にパラメータが割り振られており、任意に変更することで体操の一部のカスタマイズを行うことができる。

部品は Python のコードを記述し、その機能をボックスとしてアプリケーションに実装できるようにする Python ボックス[4]、複数のボックスを1つにまとめるダイアグラムボックス[4]を用いて実装する。

「部位毎のモーション」は Pepper の OS である NaoqiOS のフレームワークが提供している API の一つである ALMotion API[10]を用いて実装する。既存のモーション作成方法で作成したモーションを ALMotion API にあるメソッドに対応したコードに変換する機能があるため、これを使用しモーションを Python ボックスで扱える形にする。図 5 に Python で記述されたモーションに関するコード例を示す。

「体操動作」はダイアグラムボックスとして作成する。定義された「体操動作」には各体操に対応した「部位毎のモーション」「モーションの繰り返し」によって作成されたモーションの繰り返しパターンが含まれている。

```

from naoqi import ALProxy
names = list() #動かす関節の名前を格納するリスト
times = list() #ポーズを取る際の所要時間を格納するリスト
keys = list() #ポーズを取る際の関節の位置を格納するリスト
##頭の関節に当たるアクチュエータ群の登録
names.append("HeadPitch") #アクチュエータ名指定
times.append([0.72, 1.28, 1.36, 1.88, 1.96]) #タイミング指定
keys.append([-0.200376, -0.00124428, -0.00124428, -0.200376, -0.200376]) #アクチュエータの Rad 値指定

try:
    motion = ALProxy("ALMotion") #呼び出す API の指定
    motion.angleInterpolation(names, keys, times, True) #リストに登録したポーズの連続を再生
except BaseException, err:
    print err
    
```

図 5 モーションに関するコード例

3.3 ライブラリの使用方法

ライブラリの使用方法を述べる。図 6 に体操動作を行うアプリケーションの作成例を示す。任意の体操動作を配置・接続することで Pepper にその体操を行わせるアプリケーションを作成することができる。複数の体操を連続して行わせたい場合は、任意の順序にそれぞれの体操動作を接続する。このとき、体操動作間の切り替わりはスムーズなものになる。



図 6 体操動作を行うアプリケーションの作成例

アプリケーションに配置した体操動作は部品毎のパラメータを任意に変更することで細かいカスタマイズを可能にしている。体操動作全体での時間や、一部のモーションの調整などを行う場合、各部品のパラメータを変更する。ボックス左下に表示されているレンチマークをクリックすることでその部品のパラメータ編集画面が表示され、任意に変更することができる。図 7 はある体操の腕の反復運動に当たる腕のモーションに使用する腕の設定(use_arm)や時間の設定(beat)を行う画面例である。

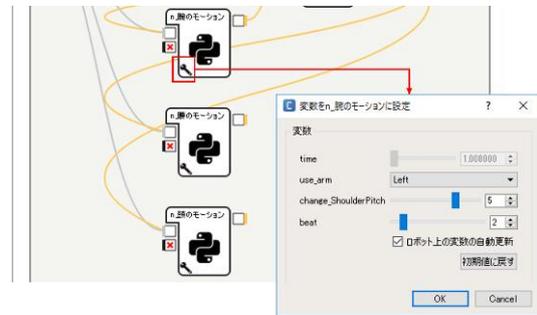


図 7 体操動作のカスタマイズ例

体操の中で体の一部を動かさないようにする場合は、動かさない部位のモーションを削除する。図 8 は『腕を振って脚を曲げ伸ばす運動』から、腰(脚部)のモーションを削除した例であり、これにより脚を動かさず、腕のみ振る運動が定義できる。

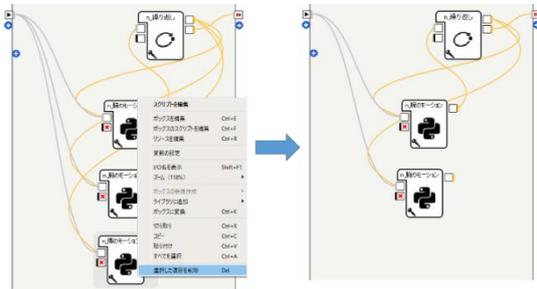


図 8 一部の部位のモーションの削除例

3.4 部品の詳細

部品毎のパラメータを中心に部品の詳細を解説する。

3.4.1 部位毎のモーション

ラジオ体操の各体操の動きを腕・腰・頭という部位のモーションとして分割した部品である。関節を動かす角度の大きさや運動の速さなどの運動強度をパラメータとして持ち、各部品に対応したパラメータを任意に変更することで部品毎のカスタマイズを可能にしている。

部品は、他の部品との接続がある場合、その部品が本来行うモーションの最後に取り姿勢を接続された部品が行うモーションの最初に取り姿勢を取るように自動で補正する機能を持つ。この機能は、部品が本来最後に行うポーズを取らずに、最後に行うポーズとその直前に行うポーズの時間差を接続された部品に送り、接続された部品ではその送られた時間を最初に行うポーズを取る時間に適用することによって実現している。部品のコード中で、この機能に関する部分の例を図 9 に示す。

```
def onLoad(self):
    ~中略~
    #next motion set
    self.next_time = self.times[-1][-1]-self.times[-1][-2]
    #最後の姿勢と最初の姿勢の時間差を宣言
    ~中略~
    def onInput_onStart(self, p = None):
        if p is not None:
            self.ftimes = [[ p ] for i in range(6)]
            self.onMotion()
            self.onStopped(self.next_time) #時間差を送信
```

図 9 他部品と接続する際のポーズの補正を実現するコード例

当部品に共通して、表 1 に示すようなパラメータを設定できる。ここで、モーションの拍数とはラジオ体操の「1,2,3,4,5,6,7,8」という掛け声を拍として扱っているものである。例えば『腕を振って脚を曲げ伸ばし運動』では腕を振って脚を曲げ伸ばし、また腕を元の位置に戻すのに「1,2,3,4/5,6/7,8」と 2 拍の動きを繰り返している。こ

のときモーションのパラメータ beat を 4 拍に変更すると 2 拍で行っていた動きを 4 拍で行わせることが可能になる。

表 1 部位毎のモーションに共通するパラメータ

パラメータ名	パラメータの概要	値
time	1 拍にかける時間	0.90~10.0[秒]
beat	モーションの拍数	1~8[拍]

(a) 腕のモーション

ラジオ体操の各体操の動きを分割したモーションのうち、腕のモーションに当たる部品である。腕のモーションに対しては、表 2 に示すようなパラメータを設定できる。change_ShoulderPitch は左右の腕が交差する際に衝突してしまうことを防ぐため、肩のアクチュエータである ShoulderPitch の角度を変更するパラメータである。ラジオ体操における腕の動きは『伸びの運動』であれば腕を左右対称に腕を動かす、『腕を振って脚を曲げ伸ばし運動』であれば腕を交差させながら腕を動かす、『身体を横に曲げる運動』であれば腕を任意の方向に腕を向けて動かす、というように体操動作毎に特徴を持つ。今回はモーション毎に適切なパラメータを割り振るために腕のモーションを分類した。左右対称の動きを行う「左右対称運動」、左右対称ながら腕を交差させる動きを行う「左右対称交差運動」、左右非対称な動きを行う「左右非対称運動」の 3 種に分類される。use_arm は「左右対称運動」「左右対称交差運動」に、change_ShoulderPitch は「左右対称交差運動」に、first_direction は「左右非対称運動」にそれぞれ割り振られる。

表 2 腕のモーションのパラメータ

パラメータ名	パラメータの概要	値
use_arm	動かす腕の選択	左 or 右
change_ShoulderPitch	肩のピッチの角度変更	-10~10°
first_direction	動き始めに向かう方向	左 or 右

(b) 腰(脚部)のモーション

ラジオ体操の各体操の動きを分割したモーションのうち、腰のモーションに当たる部品である。腰のモーションに対しては、表 3 に示すようなパラメータを設定できる。腰のモーションは腕のモーションと同様に動きを分類した結果、「屈伸する」、前方、及び横方向に腰を倒す「腰を倒す」、後方に腰を反らす「腰を反らす」、

を回す」の4種に分類される。HipPitch, HipRoll は「腰を倒す」に、first_direction は「腰を回す」に割り振られる。

「腰を倒す」は、例えば拍数が4拍(beat=4)の場合、最初に取り腰の姿勢を1_HipPitchと1_HipRoll, 2拍目に取り腰の姿勢を2_HipPitchと2_HipRoll, 4拍目に取り腰の姿勢を3_HipPitchと3_HipRollで指定することにより、腰の動きを任意に変更することができる。図10に「腰を倒す」のパラメータの変更例を示す。

「腰を回す」のパラメータ first_direction は、腰を回す動きの際に最初に取り腰の向きを指定する。

表3 腰のモーションのパラメータ

パラメータ名	パラメータの概要	値
1_HipPitch	モーションの始めにおける前方に腰を倒す角度	0~50°
1_HipRoll	モーションの始めにおける横方向に腰を倒す角度	-25~25°
2_HipPitch	モーションの中間における前方に腰を倒す角度	0~50°
2_HipRoll	モーションの中間における横方向に腰を倒す角度	-25~25°
3_HipPitch	モーションの最後における前方に腰を倒す角度	0~50°
3_HipRoll	モーションの最後における横方向に腰を倒す角度	-25~25°
first_direction	モーションの始めにおける腰を向ける方向	左 or 右

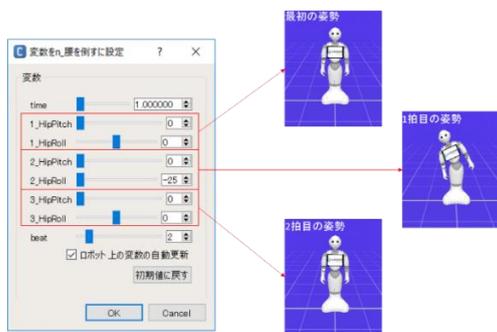


図10 「腰を倒す」のパラメータの変更例

(c) 頭のモーション

ラジオ体操の各体操の動きを分割したモーションのうち、頭のモーションに当たる部品である。頭のモーションでは、表4に示すようなパラメータを設定できる。頭のモーションは、「頭を傾ける」、「頭を回す」の2種に分類される。HeadPitch, HeadYaw は「頭を傾ける」に、first_direction は「頭を回す」に割り振られる。

「頭を傾ける」のパラメータは拍数が例えば4拍(beat=4)の場合、「腰を倒す」のパラメータと同様に、最初に取り頭の位置を1_HeadPitchと1_HeadYaw, 2拍目に取り頭の位置を2_HeadPitchと2_HeadYaw, 4拍目に取り頭の位置を3_HeadPitchと3_HeadYawで指定することにより、頭の動きを任意に変更することができる。

「頭を回す」のパラメータ first_direction は頭を回す動きの際に最初に取り腰の向きを指定する。

表4 頭のモーションのパラメータ

パラメータ名	パラメータの概要	値
1_HeadPitch	モーションの始めにおける前後に頭を傾ける角度	-30~20°
1_HeadYaw	モーションの始めにおける左右に首を回転する角度	-60~60°
2_HeadPitch	モーションの中間における前後に頭を傾ける角度	-30~20°
2_HeadYaw	モーションの中間における左右に首を回転する角度	-60~60°
3_HeadPitch	モーションの最後における前後に頭を傾ける角度	-30~20°
3_HeadYaw	モーションの最後における左右に首を回転する角度	-60~60°
first_direction	モーションの始めにおける頭を向ける方向	左 or 右

3.4.2 モーションの繰り返し

モーション、またはモーションの連続を何回繰り返すかを設定するための部品である。モーションの繰り返しに対しては、表5に示すように繰り返し回数をパラメータとして設定できる。

表5 モーションの繰り返しのパラメータ

パラメータ名	パラメータの概要	値
repeat	モーション繰り返し回数	1~10[回]

3.4.3 体操動作

部位毎のモーション、モーションの繰り返しの組み合わせによって構成された一連のモーションを1つの体操として定義する部品である。体操動作に対しては、表6に示すようなパラメータを設定できる。体操動作のパラメータのtimeとrepeatはそれぞれ「部位毎のモーション」のパラメータtimeと「モーションの繰り返し」のパラメータrepeatと同一のパラメータであり、下位に持つそれぞれの部品のパラメータに継承される。これにより、体操動作のカスタマイズを包括的に行うことを可能にしている。

表 6 体操動作のパラメータ

パラメータ名	パラメータの概要	値
time	1 拍にかける時間	0. 90~10. 0[秒]
repeat	モーションを繰り返す回数	1~10[回]

4. 評価

提案したライブラリを評価するため、ライブラリを使用したアプリケーションの作成と、作成したアプリケーションのカスタマイズの実験を行った。本章では、提案したライブラリの正常動作の確認と、作成やカスタマイズの容易性における評価について述べる。

4.1 ラジオ体操第 1 のアプリケーションの作成

提案したライブラリによってラジオ体操を行うアプリケーションを作成できるかどうか評価するため、Pepper にラジオ体操第 1 を行わせるアプリケーションの作成を行った。その結果、Pepper のハードウェアが行うことのできない『両足で跳ぶ運動』を除く 12 個の体操において、脚以外の部位を使ってそれらを構成することができた。表 7, 8, 9 にラジオ体操第 1 のために定義した 12 個の体操動作を構成するモーションを示す。

ラジオ体操第一には基本的な腕・腰・頭の動作が含まれており、また介護の現場で一般的に利用されている体操であることから、本ライブラリは基本的な表現力を持つと考えられる。

表 7 腕のモーション

モーションの名称	含まれる体操動作
腕を前から上げ横から下ろす	伸びの運動、深呼吸
腕を横に振る	腕を振り脚を曲げ伸ばす運動、胸を反らす運動
腕を回す	腕を回す運動
腕を斜め後ろ上に反らす	胸を反らす運動
腕を横から振り上げる	身体を横に曲げる運動
腕を前に伸ばす	身体を前後に曲げる運動
腰に手を当てる	身体を前後に曲げる運動
腕を巻き込み気味に振る	身体をねじる運動
腕を斜め上に振り上げる	身体をねじる運動
腕を上下に伸ばす	腕を上下に伸ばす運動
腕を斜め下に伸ばす	身体を斜め下に曲げ胸を反らす運動
腕を斜め後ろ下に反らす	身体を斜め下に曲げ胸を反らす運動
両腕を回す	身体を回す運動

表 8 頭のモーション

モーションの名称	親となる体操動作
頭を傾ける	身体をねじる運動
頭を回す	身体を回す運動

表 9 腰のモーション

モーションの名称	含まれる体操動作
屈伸する	腕を振り脚を曲げ伸ばす運動
腰を倒す	身体を横に曲げる運動、身体を前後に曲げる運動、身体を斜め下に曲げ胸を反らす運動
腰を反らす	腰を反らす運動、身体を前後に曲げる運動、身体を斜め下に曲げ胸を反らす運動
腰を回す	身体を回す運動

また、ラジオ体操第 1 の順序に沿って体操を行うアプリケーションを作成し実行した結果、体操から体操へのスムーズな移行が実現できることを確認した。これにより、本ライブラリでは連続で複数の体操を行うラジオ体操を正常に動作することができることを示していると考えられる。

4.2 従来のモーション作成方法とのカスタマイズ比較

本研究で提案したライブラリによって簡単に体操を行うアプリケーションのカスタマイズができるかどうかを検証するため、従来のモーション作成方法で作成された体操と本ライブラリで作成された体操のカスタマイズを比較評価する。

評価の方法としてアプリ開発初心者には本ライブラリで作成されたラジオ体操の一部と従来のモーション作成方法で作成されたラジオ体操の一部のそれぞれに同じカスタマイズを行ってもらい、カスタマイズにかかった時間、カスタマイズの正確さで比較する。カスタマイズの正確さはカスタマイズ後のファイルを例題の正答に当たるファイルと比較して評価する。

本評価では以下のようなラジオ体操のカスタマイズの例題を用いる。

<カスタマイズの例題>

- ・「伸びの運動」の腕を前から振り上げ横から下ろす動きの繰り返す回数を 2 回から 3 回にする。
- ・「腕を振り脚を曲げ伸ばす運動」で脚部(腰)を動かさないようにする。
- ・「腕を回す運動」を 2 倍の時間かけて行うようにする。
- ・「身体を横に曲げる運動」で、身体を横に曲げる角度を 10°にする。

この例題について、全てのカスタマイズにかかる時間を計測、カスタマイズの正確さを目視で比較することで評価を行う。

今回の被験者として4人の大学生にラジオ体操のカスタマイズを行ってもらった。被験者にはChoregrapheでどのようにPepperのアプリケーションを作成するかということを実際にChoregrapheに触れてもらいながら説明した後、本ライブラリの使用方法、従来のモーション作成方法の使用方法を各例題の直前に説明し、例題にあるカスタマイズを行ってもらった。例題を行う際もそれぞれの操作に関するマニュアルを用意し、カスタマイズで不明なことがあった時、マニュアルを閲覧するように指示した。また、例題の行う順番によるChoregrapheの操作の慣れを考慮し、被験者A、Dには最初に本ライブラリで作成されたラジオ体操のカスタマイズ、次に従来のモーション作成方法で作成されたラジオ体操のカスタマイズの順番で行ってもらい、被験者B、Cには被験者A、Dとは逆の順番で行ってもらった。

表10にカスタマイズにかかった時間の結果を示す。

表10 カスタマイズにかかった時間

被験者	提案ライブラリ利用	従来方法
A	6分55秒	33分30秒
B	8分27秒	12分15秒
C	7分37秒	24分
D	10分45秒	16分57秒

表10から、従来のモーションの作成方法で作成したラジオ体操のカスタマイズよりも本ライブラリで作成したラジオ体操のカスタマイズの方が短い時間で達成できているため、本ライブラリを用いることで体操を容易にカスタマイズできると思える。

また、カスタマイズ後のファイルの比較では、ライブラリ利用では被験者全員が正答のカスタマイズと同様のカスタマイズを行っていたが、従来のモーションの作成方法で作成したラジオ体操では「伸びの運動」「腕を回す運動」「身体を横に曲げる運動」の例題において全員異なった、正確でないカスタマイズが行われていた。カスタマイズの中には、元々のラジオ体操の動きとは全く異なる動きをするようなものもあった。

以上のような結果より、本研究で提案したライブラリは体操のカスタマイズにおいて既存のモーション作成方法よりも簡単であり、その結果正確性が改善・向上していると思える。

5. 関連研究

Pepperのアプリケーションの開発方法の側面に関する研究は、2016年6月現在筆者の知る限りない。これはPepper自体が新しいものであるため、その上で動くアプリケーションの研究がまだないと考えられる。ソフトウェア開発上の問題は、同じOSを使いソフトウェアの作成方法も似ているNaoというロボットでも起こると考えられる。Naoに関する研究は

- 転倒を防ぎ安定な状態を保つための姿勢制御に関する研究[6][9]
- 音声認識・言語処理に関する研究[3][8]
- 人とのコミュニケーションにおいて与える影響に関する研究[1][2]

以上の研究があることは確認したが、コミュニケーションロボットのソフトウェア開発、特に体操を行うアプリケーション開発に関する研究は行われていない。

本研究ではPepperの腕や腰などの動きに着目し、Pepperというコミュニケーションロボットで身振りを行う際、どのような再利用ができるか、またどのように部品化すれば良いかを検討した。

6. 結論と今後の課題

本稿では、Pepperの体操動作を作成・カスタマイズをやすくするためのライブラリを提案した。体操を教示するアプリケーションを想定したとき、体操に様々なカスタマイズを施したい場合を考慮し、モーションを適切な粒度に分割し部品化することによって、既存のモーション作成方法より細かいカスタマイズを容易にすることを目指した。ライブラリは、Pepperに体操を行わせる上で考慮される腕の衝突回避、体操間のスムーズな移行を機能として実装している。

また、本稿では提案したライブラリが正常にラジオ体操を作成することができ、既存のモーション作成方法よりも使いやすさが向上し、Choregrapheの操作に慣れていない人でも例題のカスタマイズを適切に行うことが可能であることを確認した。

今後の課題として、第4章2節で行った実験がごく限られた人数のものであり評価として十分でないことが考えられるので、評価実験の回数を増やし十分なデータをとってより適切な評価をすることが挙げられる。また、体操を教示するアプリケーションにおいて、行いたいカスタマイズや作成したい体操を調査し、それが本ライブラリで実現可能かを検証することが挙げられる。

謝辞

本論文の作成にあたり、有益なご議論とご助言を頂きました早稲田大学岸知二教授に感謝致します。

参考文献

- [1] Angelica Lim, 尾形哲也, 奥乃博. The DESIRE Model: Cross-modal emotion analysis and expression for robots. 第74回全国大会講演論文集. 2012
- [2] 胡思楊, 山崎敬一, 長谷川紫穂他. 人間・NAO・Robovie-R3の質問に対する人間の反応の比較(コミュニケーション支援及びコミュニケーション一般, ヒューマン情報処理一般). 電子情報通信学会技術研究報告. HIP, ヒューマン情報処理. 2014
- [3] 駒谷和範, 中島大一, 杉山貴昭. ロボット自身のマイクを介したNaoとの音声対話. 研究報告ヒューマンコンピュータインタラクション(HCI). 2013
- [4] 村山龍太郎, 谷沢智史他. Pepperプログラミング-基本動作からアプリの企画・演出まで. SBクリエイティブ. 2015
- [5] 沢田篤史, 平山雅之他. 組み込みソフトウェア開発技術. CQ出版. 2011
- [6] 内種岳詞, 島中利治. ロボカップ3Dサッカーシミュレーションにおける歩行パラメータ調整へのPSOの適用. 日本知能情報ファジィ学会 ファジィシステムシンポジウム講演論文集. 2011
- [7] 田口優希. Pepperモーション制作ハッカソン ~チームラボ流モーション制作術~【前編】:
http://www.slideshare.net/teamLab/pepper-41743582?qid=d9704b11-1023-4db8-ac8b-81a73ba29dce&v=&b=&from_search=14
- [8] 田村優樹, 長崎達也, 中野雅広, 原田実. 意味解析に基づくロボット指示システムAthena2011. 研究報告自然言語処理(NL). 2012
- [9] 渡辺光, 中村克彦. RoboCup用ロボットNAOのゼロモーメントポイントを用いた姿勢制御. 第75回全国大会講演論文集. 2013
- [10] Aldebaran Documentation: <http://doc.aldebaran.com/>
- [11] 「ラジオ体操」図解:
<https://pid.nhk.or.jp/event/taisou/img/radioweb.pdf>