

複数の感情の同時生起および削除を  
取り入れたエージェントの実装

奈良女子大学 理学部 情報科学科 12251520  
新出研究室 今井那緒

## 概要

近年、ロボットが人間とコミュニケーションをとることが重要視されており、その一環としてロボットに感情を持たせることが求められている。従来研究では、人間の感情を分類する OCC theory と呼ばれる理論によって、分類された 22 種の感情のうち 20 種を程度の強さの表現も含めて実装している。その従来研究を基に、本研究では時間経過による感情の程度の減衰、および複数の感情の同時生起と感情の削除を実現した。また、検証実験からエージェントが複数の感情を同時に生起し、時間経過によって程度が 0 になった感情の削除を行うことを確認した。本論文では、感情の同時生起と削除について、その実装と実験結果を述べ、最後に課題について述べる。

## 1 はじめに

近年、人間らしい感情を持つロボットの開発が進んでいる。現実世界において、ロボットが人間とコミュニケーションをとるためには、ロボットが感情を持っていることが望ましい。ロボットが人間らしい感情を持つことで、自ら人間に近い行動を決定することが求められる。

周囲の環境が常に変化する現実世界において、問題解決のために行動するロボットの実現として BDI アーキテクチャーが有効である [9]。BDI アーキテクチャーとは、人間行動を信念、願望、意図の 3 つの心的状態でモデル化した BDI モデル [4] による行為決定方式を計算機上で実現したものである。それにより、人間の合理的思考に基づいた行動をとる BDI エージェントを構築することが可能である。また、この BDI モデルは、BDI logic という論理モデルを持つ。

一方、OCC theory[3] と呼ばれる、人間の感情を分類する理論がある。この OCC theory では、信念や願望などの心的状態を用いて人間の感情を分類、特徴付けており、論理モデルによって表現が可能で、BDI logic を持つ BDI モデルとの親和性が良い。実際、OCC theory によって分類された感情は、既存の BDI アーキテクチャーの実装である Jason[2] 上の信念ベースを用いて実現が可能である。

Adam らの研究 [1] では、OCC theory の理論を BDI モデルに取り込むことで、感情を論理式でモデル化している。この研究で、BDI logic に「不確かだが起こると期待されている事象」等の新たなオペレータを複数導入することで、OCC theory で扱われている 20 種の感情を論理式として形式化し、BDI モデルに取り込んでいる。しかし、この研究においては、1 つの行動につき 1 つの感情しか生起できず、また、複合的な感情については扱っていない。それに加えて、生起する感情には度合いがなく、人間らしい感情としては不十分である。

Adam らの形式化を基に、BDI エージェントと OCC theory を用いて感情を生起し、それに伴う行動を選択するエージェントの研究 [6][5] がある。[6] では、OCC theory によって分類された 22 種類の感情のうち、20 種の感情を実装している。実装された感情には存在するか否かの判定しがなく、程度がなかった。そこで、[5] では、感情の程度を実装している。それに加えて、感情を連続して生起できるよう、生起した感情の削除を可能としている。しかし、これらの研究において、時間経過による感情の程度の減衰が無く、生起された感情が削除されないために人間らしい感情表現とは言えず、また同時に複数の感情を生起することができなかった。

そこで、本研究では、[5]の研究をもとに、時間経過による感情の程度の減衰、および複数の感情の同時生起と感情の削除を実現した。なお本研究は、奈良女子大学理学部情報科学科4回生の向井との共同研究であり、感情の程度の減衰については[10]で述べ、本論文ではそれ以外について述べる。

## 2 OCC theory

OCC theory[3]とは、Ortony,Clore,Collinsらが提唱した、感情をモデル化した理論である。人間の包括的な感情をモデル化しており、感情の特徴付けが明確であるために、比較的理解しやすく、計算機科学分野において広く用いられている。

OCC theoryによって分類された22種類の感情を表1に示す。

1. 事象の結果の望ましさに 関して	(a) 結果の望ましさに関して (自分にとって)			(b) 結果の望ましさに関して (他者にとって)		
	i. イベントに関して	ii. 予想に関して			i. 他人が良い結果を得る	ii. 他人が悪い結果を得る
	Joy (喜び) Distress (嘆き)	A. 単なる予想  Hope (望み) Fear (恐れ)	B. 予想したことが 起こった  Satisfaction (満足) FearConfirmed (恐れていたことが確定)	C. 予想したことが 起こらなかった  Relief (安堵) Disappointment (落胆)	HappyFor (共に喜ぶ) Resentment (憤り)	SorryFor (共に残念に思う) Gloating (ほくそ笑む)
2. 行動に対する 賞賛度に対して	a. 自分の行動		b. 自分の行動と起こった事象に 関して	c. 他人の行動	d. 他人の行動と起こった事象に 関して	
	Pride (自尊心) Shame (羞恥心)		Gratification (満足) Remorse (後悔)	Admiration (賞賛) Reproach (非難)	Gratitude (謝意) Anger (怒り)	
3. 対象に対する好き嫌い			Love(好き), Hate(嫌い)			

表1 OCC theoryによる分類

## 3 従来研究

従来研究[5]では、Adamら[1]の形式化を踏まえて実装された20種類の感情に程度の強さを取り入れた論理体系[8]を基に実装を行っている。本章では、従来研究における実装について述べる  
と同時に、課題とされた点を述べることにする。

### 3.1 感情の程度の強さの実装

Adamらの形式化[1]では、感情の生起の条件をBDI logicの論理式で表している。表1における22種類の感情を以下の6つのグループに分類することによって、グループ毎に同じ形式を持つ生起条件を用いる事が可能である。

- ① Well-being emotions : Joy, Distress
- ② Prospect emotions : Hope, Fear
- ③ Confirmation emotions : Satisfaction, FearConfirmed, Relief, Disappointment
- ④ Fortunes-of-others emotions : HappyFor, Resentment, SorryFor, Gloating
- ⑤ Attribution emotions : Pride, Shame, Admiration, Reproach,  
Gratification, Remorse, Gratitude, Anger
- ⑥ Attraction emotions : Love, Hate

本節では、生起条件の形式が同じものはグループから1つを取り上げ、従来研究において実装された①～⑤の感情の実装方法について説明する。

### 3.1.1 Well-being emotions

程度の強さを持つ *Joy*(喜び) の生起条件は

$$Joy_{f(d)}^i \varphi := Bel^i \varphi \wedge Des_d^i \varphi \quad (1)$$

である。これは「エージェント  $i$  がイベント  $\varphi$  が成立することを信じ、かつそのことがエージェント  $i$  にとって  $d$  程度望ましいことであるならば、 $i$  は  $\varphi$  の成立に対し、程度  $f(d)$  の *Joy* という感情を生起する」ということを意味する。

*Joy* の程度に関係する変数  $d$  として、「エージェント  $i$  においてイベント  $\varphi$  がどの程度望ましいか ( $Des_d^i \varphi$ )」を導入している。イベントの望ましき  $d$  が大きいほど、*Joy* の程度も大きくなると考えられているため、度合いを計算する関数  $f(d)$  は、[8] では望ましき  $d$  の増加関数と定義しているが、[5] での実装では、 $d$  を受け取り、そのまま *Joy* の程度として返す関数としている。

### 3.1.2 Prospect emotions

程度の強さを持つ *Hope*(望み) の生起条件は

$$Hope_{f(d,l)}^i \varphi := Des_d^i \varphi \wedge Prob_l^i \varphi \wedge l \neq 1 \quad (2)$$

である。これは「エージェント  $i$  が  $i$  にとってイベント  $\varphi$  が成立することが  $d$  程度望ましく、かつ  $\varphi$  が  $l$  程度 ( $l \neq 1$ ) 成立しそうだと思っているのであれば、 $i$  は  $\varphi$  の成立に対し、程度  $f(d,l)$  の *Hope* という感情を生起する」ということを意味する。

*Hope* の程度に関係する変数として、*Joy* と同様に変数  $d$  を使用し、それに加え、新たな変数  $l$  として、「イベント  $\varphi$  が起きる可能性 ( $Prob_l^i \varphi$ )」も導入している。イベントの望ましき  $d$ 、イベントが起きている可能性  $l$  が大きいほど、*Hope* の強さも大きくなると考えられているため、度合いを計算する関数  $f(d,l)$  は、[8] では望ましき  $d$  の増加関数、可能性  $l$  の増加関数と定義しているが、[5] での実装では、 $d$  と  $l$  を受け取り、加算して *Hope* の程度として返す関数としている。

### 3.1.3 Confirmation emotions

程度の強さを持つ *Satisfaction*(満足) の生起条件は

$$Satisfaction_{f(d,e,l)}^i \varphi := Des_d^i \varphi \wedge Bel^i \varphi \wedge Bel^i Past_e Prob_l^i \varphi \wedge l \neq 1 \quad (3)$$

である。これは、「エージェント  $i$  はイベント  $\varphi$  が成立することが  $d$  程度望ましく、かつ、過去のある時点で  $\varphi$  の成立を  $l$  程度期待し ( $l \neq 1$ )、 $\varphi$  を達成するために  $e$  程度の努力をし、かつ、 $\varphi$  が成立することを信じているならば、 $i$  は  $\varphi$  の成立に対し、程度  $f(d, e, l)$  の *Satisfaction* という感情を生起するということを意味する。

*Satisfaction* の程度に関係する変数として、これまでと同様に変数  $d, l$  を使用し、それに加え、新たな変数  $e$  として、「 $\varphi$  を達成するために費やした努力の程度」も導入している。イベントの望ましさ  $d$ 、イベントを成立するために費やした努力  $e$ 、イベント  $\varphi$  が起きている可能性  $l$  が大きいほど、*Satisfaction* の強さも大きくなると考えられているため、度合いを計算する関数  $f(d, e, l)$  は、[8] では望ましさ  $d$  の増加関数、努力  $e$  の増加関数、可能性  $l$  の増加関数と定義しているが、[5] の実装では、 $d$  と  $e$  と  $l$  を受け取り、加算して *Satisfaction* の程度として返す関数としている。

### 3.1.4 Fortunes-of-others emotions

程度の強さを持つ *HappyFor*(共に喜ぶ) の生起条件は

$$HappyFor_{f(d_i, d_j)}^{i,j} \varphi := Bel^i \varphi \wedge Bel^i Des_{d_j}^j \varphi \wedge Des_{d_i}^i Bel^j \varphi \wedge Bel^i Deserve^i \varphi \quad (4)$$

である。これは「エージェント  $i$  がイベント  $\varphi$  の成立を信じており、かつ、 $i$  が『エージェント  $j$  が  $\varphi$  を  $d_j$  程度望んでいる』ことを信じており、かつ、 $j$  が  $\varphi$  の成立を信じていることを  $i$  は  $d_i$  程度望んでおり、かつ、 $i$  は  $\varphi$  が  $j$  にふさわしいと思っているならば、 $i$  は  $j$  が  $\varphi$  を成立させたことに対し、程度  $f(d_i, d_j)$  の *HappyFor* という感情を生起する」を意味する。

*HappyFor* の程度の強さに関係する変数  $d_i$  として、「 $i$  にとって『 $j$  が  $\varphi$  をさせること』がどの程度望ましいか ( $Des_{d_i}^i Bel^j \varphi$ )」と、変数  $d_j$  として、「 $j$  が  $\varphi$  をどの程度望ましいか ( $Des_{d_j}^j \varphi$ )」が定義されている。 $i$  と  $j$  のイベントに対する望ましさが大きいほど、*HappyFor* の強さも大きくなると考えられているので、度合いを計算する関数  $f(d_i, d_j)$  は、[8] では望ましさ  $d_i$  の増加関数、 $d_j$  の増加関数と定義しているが、[5] の実装では、 $d_i$  と  $d_j$  を受け取り、加算して *HappyFor* の程度として返す関数としている。

### 3.1.5 Attribution emotions

程度の強さを持つ *Pride*(誇りに思う) の生起条件は

$$Pride_{f(l,p)}^i \varphi := Bel^i Past Prob_l^i \varphi \wedge Bel^i \varphi \wedge Praise_p^i \varphi \quad (5)$$

である。これは「エージェント  $i$  が過去の時点でイベント  $\varphi$  の可能性は  $l$  程度と信じており、かつ、 $i$  が  $\varphi$  が成立したことを信じており、かつ、 $\varphi$  が成立したことは  $p$  程度賞賛されると判断されるならば、 $i$  は  $\varphi$  が成立したことに  $f(l, p)$  程度の *Pride* という感情を生起する」を意味する。

*Pride* の程度の強さに関係する変数  $l$  として、エージェントの行動の意外性 ( $Bel^i PastProb^i \varphi \wedge Bel^i \varphi$ ) が定義されている。この論理式は、過去に見込みが低いと思われていたことが、現在は真であるということの意味する。また、変数  $p$  として、イベントが賞賛に値すると判断される程度 ( $Praise_p^i \varphi$ ) が定義されている。行動の意外性  $l$ 、賞賛に値する判断される程度  $p$  が大きいほど、*Pride* の強さも大きくなると考えられているので、度合いを計算する関数  $f(l, p)$  は、[8] では意外性  $l$  の増加関数、賞賛度  $p$  の増加関数と定義しているが、[5] の実装では、 $l$  と  $p$  を受け取り、加算した値を *Pride* の強さを返す関数としている。

### 3.2 実装上の課題点

従来研究では、時間経過による程度の減衰が無く、生起された感情が削除されなかった。また、1つの感情を連続して生起することはできるが、同時に複数の感情を生起することができなかった。

次節以降で、本研究におけるこれらの改善について述べる。なお時間経過による程度の減衰については、[10] で述べる。

## 4 複数の感情の同時生起

従来研究では、エージェントが行動する時のみ、感情の生起の検査をしていた。また、行動選択の条件を判定する際に、一つの感情の生起が検査されれば終わりとし、複数の感情の生起を検査していなかった。例えば、図1のように外部入力を与えると、図2のような結果が得られる。本来ならば、このような外部入力を与えると、*sanpo* に対する Joy が生起した後に *hone* に対する Hope が生起するべきである。しかし、一つの感情の生起しか検査しないため、Hope が生起したことが得られず、それに従った行動も起きないという問題がある。そこで、本研究ではゴールの追加を行う時ではなく、感情を生起する原因となる信念の追加時に感情が生起したかを調べられるようにし、その際に複数の感情の生起を検査することで、同時生起を可能とした。本章では、その実装方法について述べる。

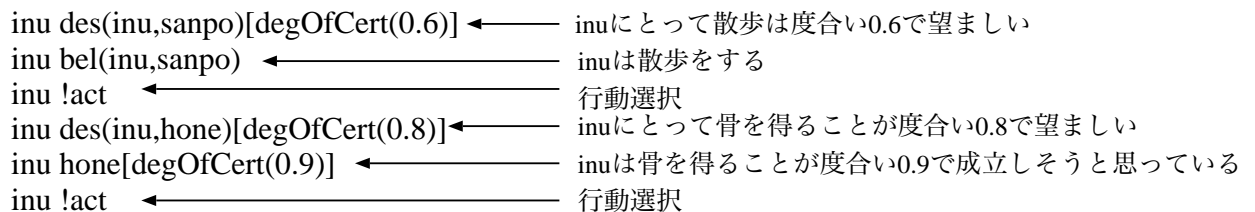


図 1



## 5 感情の削除

従来研究では、時間経過による感情の程度の減衰については考えられていなかった。そのため、感情の程度が0になることがなく、感情がなくなることはなかった。そこで、本研究では時間経過による程度の減衰によって程度が0になった感情を削除出来るようにした。本章では、その実装方法について述べる。

程度が0になった感情は存在していないとして扱うべきである。そのため、程度が0になった感情は削除する必要がある。本研究では、エージェントが行動を起こす度に、信念中で感情が生じた状況と時間を保持している述語 `time` を更新するようにした。個々の感情について、今まで保持していた述語を信念から削除し、新たに行動した時間とその時の度合いを持つ `time` 述語を信念に追加することで更新している。この更新された度合いが0ならば、追加をせずに削除のみ行うようにした。

しかし、信念を削除しただけだと、次に新たな信念を追加した時に削除したはずの感情が再度生起する可能性がある。例えば図1の例で、*sanpo* に対する Joy が削除された後に、`hone[degOfCert(0.8)]` と入力すると、*sanpo* に対する Joy の生起に使われた信念が残っているため、削除したはずの Joy が生起してしまう。そのため、感情の生起に使用される信念は最新のものを使うようにする必要がある。そこで、追加される信念全てに追加時刻を表すアノテーションをつけることで、その信念がいつ追加されたかがわかるようにしている。実際の実装形式は図4、図5となる。

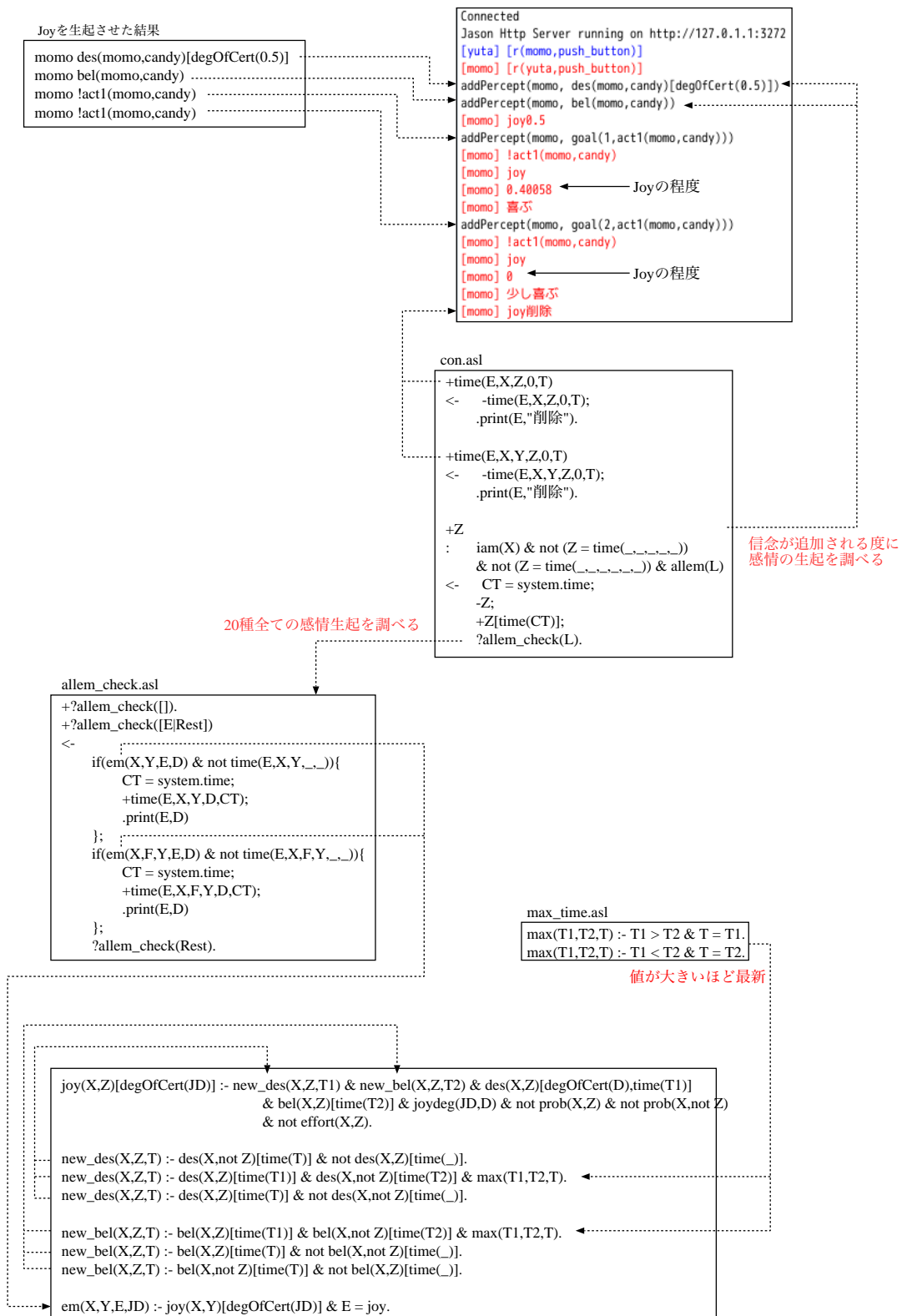


図4 感情の削除

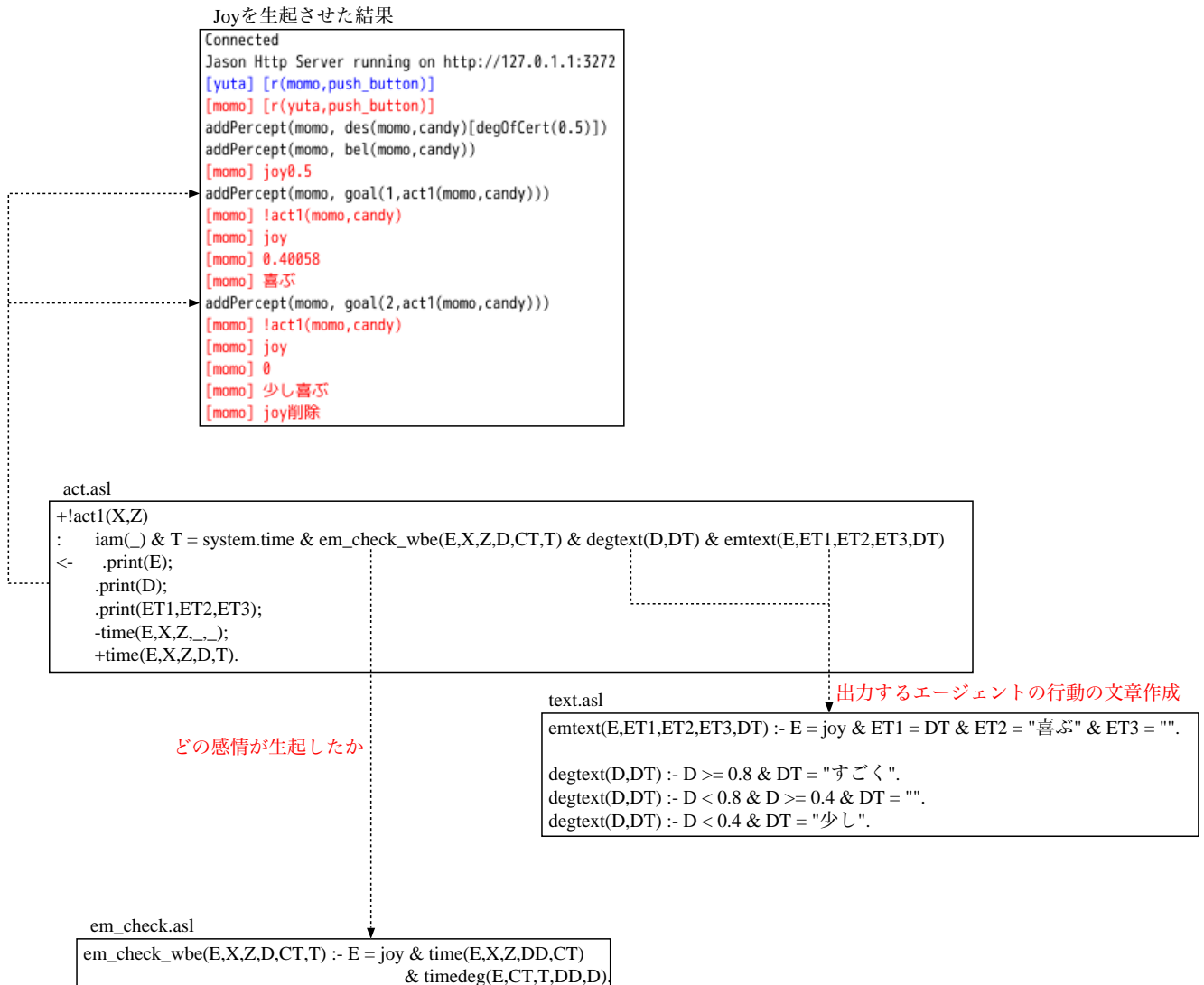


図5 感情の程度の更新

## 6 実験と結果

本章では、本研究で実現した感情の同時生起および削除の妥当性に関する検証について述べる。本研究は BDI エージェントの感情生起について扱うため、検証も、感情という心的状態に基づいてエージェントが行為を決定する部分が期待通り機能するかどうかまで含めて行う。生起した感情の妥当性に関する実験は、本来、被験者を用いた心理学的実験で行うことが理想的である。しかし、それには大規模な調査が伴い、困難である。よって、本研究ではその代替として、文学作品のストーリー通りに感情が生起されるかどうかの検証を行う。これで充分とは言えなくとも、妥当かどうか判断する一つの指針になると考える。文学作品としては、児童文学「せかいにひとつだけの

ケーキ」を用い、そのストーリーの一部をもとに準備したシナリオに従って、ストーリー通りの感情生起と行動決定ができるかを確認する。

実験は、生起した感情をもとに行動を決定するプランを Jason 上で記述することによって行った。感情の生起にはエージェントに対して任意のタイミングで外部から信念やゴールの追加を行うイベントを与える必要があるため、従来研究で用意されていた外部から TCP の 49999 番ポートに接続し、指定したエージェントにこれらのイベントを与えられるような環境を使用した。

実験に使用したシナリオは、以下の通りである。

エージェント yuta, momo がいるとする。yuta と momo はバスに乗っており、降りるバス停ではないため yuta は momo にバスの降車ボタンを押してほしくないと思っていた。momo がバスの降車ボタンを押さないだろうとも思っていたが、momo がボタンを押したことで、yuta は怒る。

yuta が怒ったことに対し、momo は怒られたくないと思っていたため、悲しくなり泣き出す。yuta は momo が泣き出したことに対し、泣き止ませるためにアメを買う約束を momo とする。アメを買う約束をしてもらった momo は、嬉しくなり喜ぶ。

その後、降りるバス停になったため、yuta が降車ボタンを押すと、降車ボタンを押したいと思っていた momo が yuta がボタンを押したことに怒り出す。

## 6.1 結果

シナリオに基づいて、入力を図 6 のように与え、結果は図 7 のようになった。なお、両エージェントに目標 `act_rep_check` が生成されると、`reproach` が生起するかどうかにより、異なるプランが選択されるようになっている。

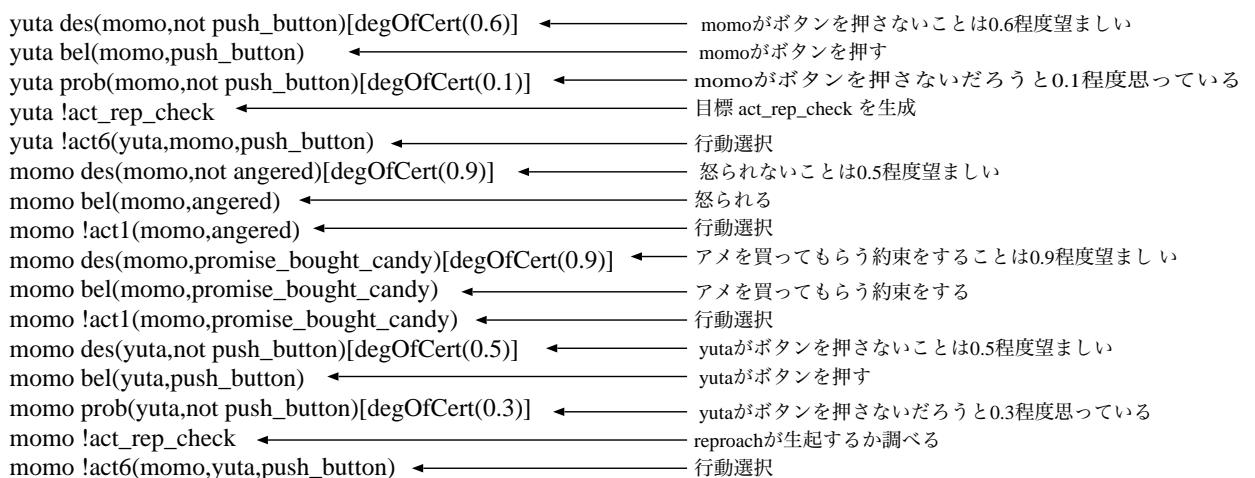


図 6 外部入力

```

Connected
Jason Http Server running on http://127.0.1.1:3272
[momo] [r(yuta,push_button)]
[yuta] [r(momo,push_button)]
addPercept(yuta, des(momo,not (push_button))[degOfCert(0.6)]) ← momoがボタンを押さないことは0.6程度望ましい
addPercept(yuta, bel(momo,push_button)) ← momoがボタンを押す
[yuta] distress0.6
addPercept(yuta, prob(momo,not (push_button))[degOfCert(0.1)]) ← momoがボタンを押さないだろうと0.1程度思っている
addPercept(yuta, goal(1,act_rep_check)) ← 目標 act_rep_check を生成
[yuta] !act_rep_check
[yuta] okyuta[r(momo,push_button)]
[yuta] anger1
[yuta] reproach1
[yuta] push_button holds
addPercept(yuta, goal(2,act6(yuta,momo,push_button))) ← 行動選択
[yuta] !act6(yuta,momo,push_button)
[yuta] anger
[yuta] 0.86244
[yuta] すごく怒る
addPercept(momo, des(momo,not (angered))[degOfCert(0.9)]) ← 怒られないことは0.9程度望ましい
addPercept(momo, bel(momo,angered)) ← 怒られる
[momo] distress0.9
addPercept(momo, goal(3,act1(momo,angered))) ← 行動選択
[momo] !act1(momo,angered)
[momo] distress
[momo] 0.7782
[momo] 泣く
addPercept(momo, des(momo,promise_bought_candy)[degOfCert(0.9)]) ← アメを買ってもらう約束をすることは0.9程度望ましい
addPercept(momo, bel(momo,promise_bought_candy)) ← アメを買ってもらう約束をする
[momo] joy0.9
addPercept(momo, goal(4,act1(momo,promise_bought_candy))) ← 行動選択
[momo] !act1(momo,promise_bought_candy)
[momo] joy
[momo] 0.80654
[momo] すごく喜ぶ
addPercept(momo, des(yuta,not (push_button))[degOfCert(0.5)]) ← yutaがボタンを押さないことは0.5程度望ましい
addPercept(momo, bel(yuta,push_button)) ← yutaがボタンを押す
[momo] distress0.5
addPercept(momo, prob(yuta,not (push_button))[degOfCert(0.3)]) ← yutaがボタンを押さないだろうと0.3程度思っている
addPercept(momo, goal(5,act_rep_check)) ← reproachが生起するか調べる
[momo] !act_rep_check
[momo] okmomo[r(yuta,push_button)]
[momo] anger1
[momo] reproach1
[momo] push_button holds
addPercept(momo, goal(6,act6(momo,yuta,push_button))) ← 行動選択
[momo] !act6(momo,yuta,push_button)
[momo] anger
[momo] 0.94496
[momo] すごく怒る

```

図7 ストーリーを再現した結果

赤文字はエージェント yuta、青文字はエージェント momo からの返答、黒文字は信念が追加されたことを示している。図7より、確かにシナリオ通りの感情生起と行動決定が行われていることが確認できた。

## 7 終わりに

本研究では、複数の感情の同時生起、感情の削除を取り入れたことで、より人間らしい感情表現の実現を目指した。感情が同時に複数生起することで、より複雑な行動選択を可能とした。

今後の課題として、感情の削除において、その感情が生起するのに必要とした信念の削除ができるようになる必要がある。感情を削除した後に、他の感情を生起するための信念の入力の順番を変えることで、削除したはずの感情が再度生起されることがあり、これは不適切である。また、感情の同時生起において、20種の感情を1つ1つが生起できるかどうかを調べているために、感情が生起されるまでに処理時間がかかる。現実世界において、状況というのは刻一刻と変わるものなので、感情が生起するまでに時間がかかることは望ましくない。これらの課題を解決することで、現実世界において、より人間らしい感情を生起できるよう目指したい。

## 謝辞

本研究の遂行および本論文の作成にあたり、指導教員の新出尚之准教授から丁寧なご指導、ご助言を賜りました。心からの感謝の気持ちと御礼を申し上げたく、謝辞にかえさせていただきます。

## 参考文献

- [1] Carole Adam, Andreas Herzig, and Dominique Longin. A logical formalization of OCC theory of emotions. *Synthese*, Vol. 168, No. 2, pp. 201-248, 2009.
- [2] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming MultiAgent Systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.
- [3] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.
- [4] Anand S. Rao, Munindar P. Singh, and Michael P. Georgeff. *Formal Methods in DAI: Logic-Based Representation and Reasoning*. Massachusetts Institute of Technology, 1999.
- [5] 石川葉子. 感情に程度の強さを取り入れたエージェントの実現. 2014年度卒業論文. 奈良女子大学理学部情報科学科, 2015.
- [6] 山根瑞樹. OCC theoryによる感情表現を持つエージェントの実現. 2013年度卒業論文, 奈良女子大学理学部情報科学科, 2014.
- [7] 池之内彰子. OCC theoryに基づくエージェントの感情表現の改良. 2012年度卒業論文, 奈良女子大学理学部情報科学科, 2013.
- [8] 池之内彰子, 新出尚之. OCC theoryに基づくエージェントの感情表現の論理モデルについて. In *Proc. of JAWS2014*, 2014.
- [9] 藤田恵, 片山寛子, 新出尚之, 高田司郎. 実世界の多様性に適応したBDIロボットについて. 情

報処理学会論文誌数理モデル化と応用, Vol. 5, No. 1, pp. 50-64, 2012.

- [10] 向井香里. 時間経過による感情の程度の減衰を取り入れたエージェントの実現. 2015 年度卒業論文, 奈良女子大学理学部情報科学科, 2016.