

コミュニケーション成立の観察実験に基づく認知モデルの設計 Design of a Cognitive Model that Forms a Communication System

森田 純哉[†], 金野 武司[†], 橋本 敬
Junya Morita, Takeshi Konno, Takashi Hashimoto

[†] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
j-morita@jaist.ac.jp, t-konno@jaist.ac.jp, hash@jaist.ac.jp

Abstract

How can we communicate with people who don't share a common language? To explore this question, this paper describes a computational model that simulates a experimental study. In the experiment, we employed a coordination task in which participants used a novel communication media that restricted usual communication methods. The results of the experiment suggest the three components of the successful communication system; routine behaviors, a symbol system, and role-sharing. The model was developed to build these components from repeated interactions of ACT-R agents. We hypothesize that the multiple learning mechanism implemented in ACT-R architecture such as instance-based learning, reinforcement learning, and production compilation provide the scaffold for building a common communication system.

Keywords — Communication, Coordination, ACT-R

1. はじめに

我々は、言葉の通じない環境においても、何らかの手段を用いて他者との意思疎通を試みる。また、同一の相手との接触を繰り返すうちに、はじめは分からなかった相手の言動を徐々に理解できるようになる。このような状況での個体間コミュニケーションは、どのようにして成り立つのであろうか。この問いを検討することは、人間のもつ複雑なコミュニケーションシステムの成立を理解する有用な試みである。

この問題を検討する有効なアプローチとして、実験室内で人工的にコミュニケーション環境をデザインする手法、および計算機上でのそのプロセスのモデル化を考えることができる。Galantucci (2005) は、コミュニケーションシステムの創発を観察・操作可能な実験を行った。この研究では、実験参加者に相対での会話をさせず、あらかじめ意味も運用方法も決まっていないコミュニケーショ

ンメディアを別途用意し、それを用いてある種の調整課題を解くことを求めた。結果、当初は共通の意思伝達の手段を持ち合わせないペアが、複雑な記号ルールを含むコミュニケーションシステムを発現させることを確かめた。そして、そのようなシステムの成立と、習慣的行動や役割分担などの社会的行動の成立が関連することを議論した。

本研究は、先行研究と比べ、より制限された状況を設定し、これらの知見を定量的に確かめることを試みる。また、同時に記号処理的な認知モデルを構築し、記号ルールの成立に関与する認知的要因を検討する。以下では、実施された実験の概略とこれまでに構築されたモデルを示す。

2. 実験

2.1 方法

2.1.1 参加者

大学院生・大学院勤務の研究者から構成される21ペアが実験に参加した¹。

2.1.2 課題

2名の実験参加者がペアになり、それぞれ別の場所から実験に参加した。実験環境のソフトウェアは、Webブラウザ上で操作可能なWebアプリケーションであった。図1にそのスクリーンショットを示す。実験参加者は、マウスを利用することでソフトウェアを操作した。

課題は、独立したラウンドの繰り返しから構成される調整ゲームであった(図2)。各ラウンドにおいて、2名の参加者は、それぞれ別々の初期位置に配置されるキャラクターを、共通の位置に移動させることを目指した。位置の移動は上下左右のみ許され、それぞれの参加者には、初期位置から移ることのできない位置が存在した。そのため、参加者は移動に先立ち、共通の移動先を事前に決定する必要があった。参加者は、意味や運用ルー

¹詳細は、金野・森田・橋本 (2011) を参照。

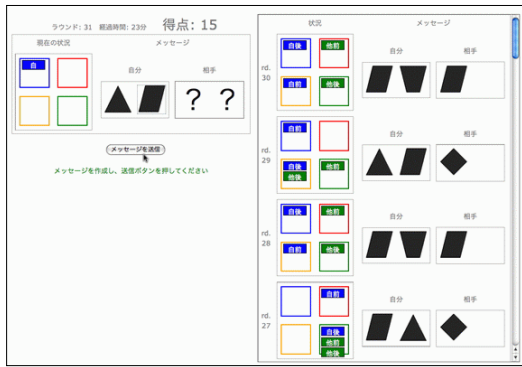


図 1 実験環境の画面

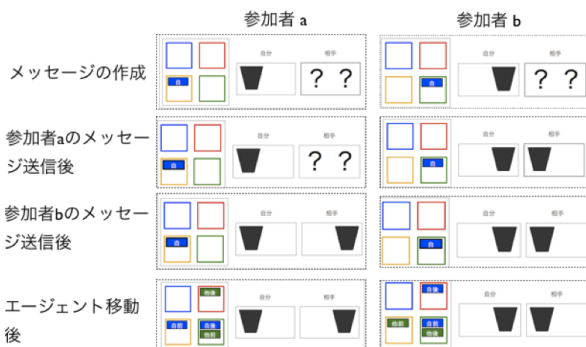


図 2 ラウンドの流れ。点線の枠内は各時点において各被験者に提示される画面を示す。各画面の左に自分のキャラクター（「自」と表示）の位置が示され、右に図形（シンボル）を組み合わせたメッセージが示される。シンボルは、6種類の中から選ぶことができた（□, ■, ▲, ▼, ◆, △）。ラウンドの最後に、自身とパートナーのエージェントの移動前後の位置が示された。

ルが事前に定められない6種類の図形を横に2つ組み合わせさせたメッセージをパートナーへ送信できた。メッセージの送信は各ラウンドにおいて一度のみ許された。また、課題においてメッセージの交換は非同期になされた。すなわち、一方の参加者がメッセージをパートナーへ送信すると、パートナーの画面に直ちにそのメッセージが送信された²。このため、本実験課題において、参加者達は、現在位置の送信者、移動先の送信者などの役割を分担することができた。

2.1.3 手続き

上記の課題環境において、共通のコミュニケーションシステムの構築を目指すトライアルセッションが行われた（1時間）。続いて、トライアルセッションにおいて生成されたコミュニケーションシステムの性質を検討するために、以下3つのテストセッションが実施された（各12ラウンド）。

²実際には通信による遅延が存在。

- T_{NM} (Test with No Message): メッセージのやりとりが完全に制限されたテスト
- T_{SM} (Test with Simultaneous Message Exchange): メッセージが同時に交換され、役割分担が制限されたテスト
- T_{IM} (Test with Immediate Message Exchange): トライアルセッションと同一の内容のテスト

2.2 結果と考察

メッセージを使用することのできたテストセッション (T_{SM} , T_{IM}) は、メッセージを使用できなかったテスト (T_{NM}) に比べ、高い成績を示した。ここから、設定された課題において、記号を使った有効なコミュニケーションシステムが生成されることが示された。さらに、役割分担が可能であったテスト (T_{IM}) は、そうでなかったテスト (T_{SM}) に比べ高い成績を示した。この結果、参加者がメッセージ交換の役割分担を有効に活用することができたことが示された。

続けて、トライアルセッション中の行動の傾向と、最終的に成立したコミュニケーションシステムの関係を検討した。結果、トライアルセッションにおいて、移動先の位置を偏らせる傾向が高く、2者間のメッセージ送信の時間差が短かったペアほど、テストセッションにおいて記号コミュニケーションを円滑に成立させたことが明らかになった。

この結果は、共通の習慣の形成やメッセージ交換の円滑さなどの協調的な行動様式が記号コミュニケーションの成立と関連することを示唆するものである (Scott-Phillips et al., 2009)。

3. モデル

3.1 アーキテクチャ

前節に示した実験環境は、記号処理的な認知モデリングとの親和性が高い。特に、著者らは、ACT-R (Adaptive Control of Thought-Rational: Anderson, 2007) の持つ複合的な学習機構により、コミュニケーションの成立と協調行動の形成の相互作用を再現できると考えている。

ACT-R は、統合認知アーキテクチャ (Newell, 1994) として構築されたプロダクションシステムである。知覚、運動、記憶、思考に関わる基礎的な心理実験の積み重ねによって構成され、人間の認知に関わる多様な現象を統合的に説明する。

ACT-R の一つの特徴は、複数のモジュールの集合として認知の構成を説明することにある。図3は、本研究において構成するモデルを ACT-R のアーキテクチャ上で図示したものである。図3下部

の中央のボックスにACT-Rを構成するモジュールが示される。ACT-Rは、プロダクションモジュールを中心として、周辺に4つのモジュールが結合される形で構成される。周辺モジュールは、情報を一時的に保持するバッファをもつ。プロダクションモジュールは、それらバッファの状態を条件として、プロダクション（if-thenルール）を選択・発火させ、バッファの状態を書き換える。

周辺モジュールのうち、知覚モジュールと運動モジュールは外界との相互作用を受け持つ。ゴールモジュールは、課題状況の内部表現を形成することで、プロダクションの発火を制御する。宣言的モジュールは、意味記憶やエピソード記憶などの宣言的知識を格納する。ACT-Rにおける情報の単位はチャンク（スロットと値のペア）であり、宣言的モジュールは、チャンクが相互に結合されたネットワークとして情報を格納する。

ACT-Rでは、これらモジュールに関連する処理（プロダクションの発火、記憶検索、注意のシフト、リーチングなど）に要する時間を、過去の心理実験の結果に基づいて推定している。この推定値を用いることで、ACT-Rは、心理実験における実時間と対応したプロセスのシミュレーションを実現できる。

また、ACT-Rは、シンボリックな学習とサブシンボリックな学習を統合するアーキテクチャでもある。シンボリックな学習として、ゴールバッファの状態を宣言的モジュールに蓄積する事例ベース学習 (Lebiere, et al., 2007) と新たなプロダクションを生成するコンパイル学習などのメカニズムが備わる。サブシンボリックな学習として、宣言的モジュールに蓄えられたチャンクの活性値を制御するベースレベル学習とプロダクションの効用値を報酬に基づいて調整する強化学習が備わる。

著者らは、これら複数の学習メカニズムの組み合わせにより、コミュニケーションシステムの成立が導かれると考える。特に、本稿では、強化学習、事例ベース学習、プロダクションコンパイルを組み入れたモデルを構築する。強化学習を通して習慣的行動が形成され、事例ベース学習を通することで、状況と記号を対応付けるシンボルシステムが形成される。さらに、コンパイル学習を通することで、役割分担の円滑化が導かれる。ACT-Rによるモデリングにより、コミュニケーションシステムの成立、およびそれに随伴する行動傾向の発現に寄与する学習メカニズムが明らかになる。また、各学習アルゴリズムの有無を操作するシミュレーションを実施することで、学習アルゴリズム間の相互作用を検討することが可能になる。

以下では、このプロセスを具体的に議論するた

めに、ACT-R上に構築された著者らのモデルを詳細に示す。

3.2 シミュレーション課題環境

本研究におけるコミュニケーションのモデルは、ACT-Rアーキテクチャ上に構築された2体のエージェントが、実験環境を介してインタラクションするものである（以降、シミュレーションにおける個体をACT-Rエージェント、2体のエージェントを合わせモデルと呼ぶ）。本研究では、ACT-Rエージェントによって操作可能な実験環境を、ACT-R Graphical User Interface (AGI) によって構築した。

AGIは、ディスプレイ上の知覚情報を、ACT-Rによる認識が可能な記号（宣言的知識と同様の形式で表現されるチャンク）によって表現する。ACT-Rエージェントの視覚モジュールは、色や形状などの視覚的特徴を手がかりとし、ディスプレイ上のチャンクを検索する。また、ディスプレイ上の特定の座標位置に注意を向けることで、視覚バッファにチャンクを格納する。

また、シミュレーション課題環境は、モデルによる操作が可能な仮想的マウスを含む。ACT-Rエージェントはディスプレイ上の特定の位置にマウスカーソルを移動させ、マウスをクリックすることができる。これにより、ACT-Rエージェントは、実験の参加者と同様の動作で、メッセージを作成/送信し、キャラクターを移動できる。

なお、モデルの実装上、2対のACT-Rエージェントのプロセスは、単一のスレッド上で実行される（タイムシェアリング方式を採用）。そのため、片方のエージェントによる課題環境の操作は、他方のエージェントの課題環境に遅延して反映される（遅延時間: 1秒）。この設定は、2節で示した実験環境において生起する通信の遅延と対応するものとみなせる。

3.3 ゴールバッファ、および宣言的知識の蓄積

図3の下部右には、本研究のACT-Rエージェントがもつゴールバッファを示している。ゴールバッファは、各時点での課題の状態を示す複数のスロットから構成される。本研究のACT-Rエージェントがもつゴールバッファは、以下のスロットを含む。

1. 位置に関わるスロット: *upper-left*, *upper-right*, *lower-left*, *lower-right*のいずれかを値とする。
 - *self-loc-pre*: 自分のキャラクターの初期位置。ロットは、

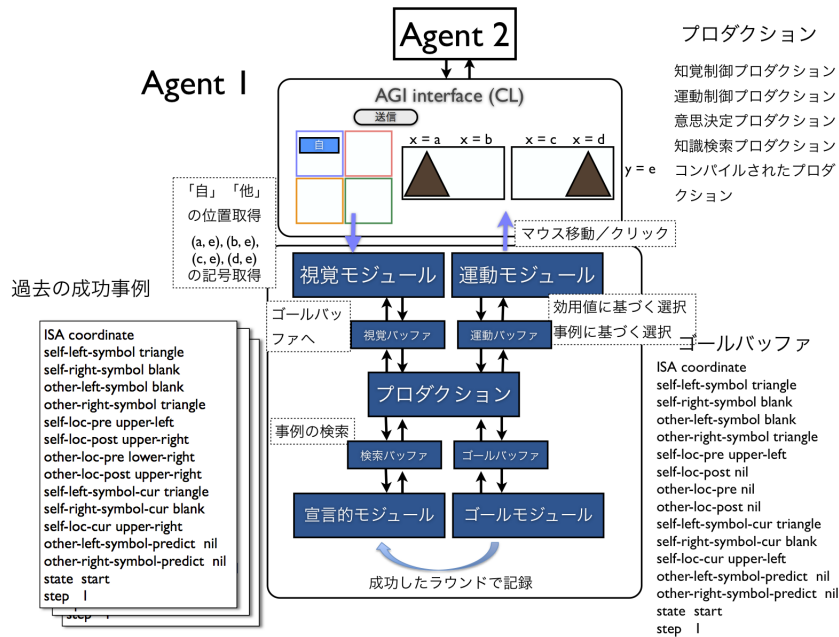


図3 ACT-Rの枠組みによるモデルの構成。2体のACT-Rエージェントが、課題インタフェースを介して結合する。ACT-Rエージェントは、課題インタフェース上の視覚情報を取得し、マウス操作を実行する。取得された視覚情報はゴールバッファに逐次記録され、ラウンドの終了時に内容が宣言的モジュールに蓄積される。エージェントは、直面するゴールバッファの状態と類似した過去の事例を検索し、合致したものが存在した場合にその行動を踏襲する。同一の宣言的知識が繰り返し検索された場合、状況とシンボルを直接的に結びプロダクションコンパイルが生じる。

- *self-loc-post*: 自分のキャラクターの移動先の位置。
 - *partner-loc-pre*: パートナーのキャラクターの初期位置。
 - *partner-loc-post*: パートナーのキャラクターの移動先の位置。
2. シンボルに関わるスロット: "blank", "square", "parallelogram", "trapezoid", "prizm", "triangle"のいずれかを値とする。
- *self-left-symbol*: 自分が送信するメッセージの左側のシンボルの値。
 - *self-right-symbol*: 自分が送信するメッセージの右側のシンボルの値。
 - *partner-left-symbol*: パートナーから受け取ったメッセージの左側のシンボルの値。
 - *partner-right-symbol*: パートナーから受け取ったメッセージの右側のシンボルの値。
 - *partner-left-symbol-predicted*: パートナーから送信されるメッセージの左側のシンボルの予測値。
 - *partner-right-symbol-predicted*: パートナーから送信されるメッセージの右側のシンボルの予測値。

ACT-Rエージェントは、各ラウンドにおいて、知覚モジュール、あるいは宣言的モジュールから取得された値を用い、これらのスロットを埋めて

いく。各ラウンドの終了時に、埋められたスロットの値は全てリセット (nil に置き換えられる) され、新たなラウンドに取り組む。

また、成功したラウンドの終了時には、その時点でのゴールバッファの状態が、事例として宣言的モジュールに格納される(図3の左下)。格納された事例は、次のラウンド以降、ゴールバッファのスロットを埋めるために用いられるようになる。

3.4 プロダクションモジュール

本研究において構築したACT-Rエージェントは、100のプロダクションを保持する。図4から図6にモデルが保持する主なプロダクションを示した。モデルのプロセスは、Step 1 (メッセージ送信と移動先の決定)、Step 2 (移動先の再考とキャラクターの移動)、Step 3 (結果の取得と学習)に別れる。この区分は、課題環境の画面遷移(図2)と対応する。以下、各ステップにおけるプロセスの概略を示す。

3.4.1 Step 1: メッセージの送信と移動先の決定

Step 1において、ACT-Rエージェントは、自分のキャラクターの初期位置を認識した後、送信するメッセージ(左右のシンボル)を決定し、自分の

キャラクターの移動先を決定する。2体のACT-Rエージェントが、パートナーにメッセージを送信した時点でこのステップは終了する。

メッセージと移動先は、過去事例の利用（図4: 左・中央のパス）、プロダクション間の競合解消（図4: 右のパス）のいずれかによって決定される。過去事例の利用による決定は、さらにパートナーのメッセージを検索の手がかりとするもの（図4: 左のパス）と、自分のキャラクターの初期位置のみを手がかりとするものに分かれる。2つの事例検索のパスのいずれが選択されるかは、*FIND-PARTNER-LEFT-SYMBOL*と*RETRIEVE-CASE*という2つのプロダクションの競合解消の結果に依存する。ACT-Rにおけるプロダクション間の競合解消は、各プロダクションに割り当てられた効用値 (utility) の比較による。効用値は後述する強化学習の手続きによって変化する。

*FIND-PARTNER-LEFT-SYMBOL*が選択された場合、ACT-Rエージェントは、知覚モジュールを用いて、課題環境中のパートナーのメッセージを探索する。これに成功した場合、ACT-Rエージェントは、パートナーのメッセージをゴールバッファの-slot (partner-left-symbol, partner-right-symbol) に格納し、現在のゴールバッファの状態とマッチする事例を、宣言的モジュールから検索する (*RETRIEVE-CASE-FROM-PARTNER*を使用)。事例の検索に成功した場合、送信するメッセージ (*self-left-symbol*, *self-right-symbol*)、移動先 (*self-loc-psot*) の値を、事例からゴールバッファにコピーする。パートナーのメッセージの取得や過去事例の検索に失敗した場合には、再度、*FIND-PARTNER-LEFT-SYMBOL*と*RETRIEVE-CASE*の競合解消がなされる。

*RETRIEVE-CASE*が選択された場合、自分の現在位置 (*self-loc-pre*) を手がかりとした事例検索が行われる。この検索に成功した場合には、上記と同様、検索された事例から送信メッセージ (*self-left-symbol*, *self-right-symbol*)、移動先 (*self-loc-psot*) の値が、ゴールバッファにコピーされる。加えて、検索された事例におけるパートナーのメッセージの値が、予測値としてゴールバッファの-slot (*partner-left-symbol-predict*, *partner-right-symbol-predict*) に格納される。この予測値は、後述するStep 2における移動先の再考に用いられる。

*RETRIEVE-CASE*による過去事例の検索に失敗した場合には、送信メッセージと移動先が、図4右に示されるプロダクション間の競合解消によって決定される。図4右側に配置される3つのボックスは、それぞれ送信メッセージの左側シンボルを決定するプロダクション、送信メッセージの右側シ

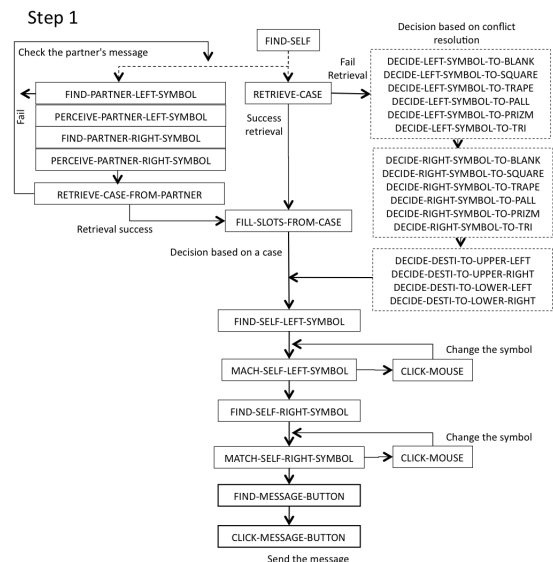


図 4 Step 1におけるプロダクションの発火系列。図中で、プロダクションがプロセスの系列にしたがって結合される。点線の矢印で結合されたプロダクションは、競合解消の結果発火するルールを示す。また、点の枠線で囲まれたプロダクションは、競合する関係にあり、各時点でいずれか1つのみ発火する。

ンボルを決定するプロダクション、移動先を決定するプロダクションを示す。これらのボックス内のプロダクションは、共通の条件節をもち、効用値による競合解消に依存して発火する。

上記のプロセスによって、送信メッセージ、移動先が決定した後に、ACT-Rエージェントは課題環境を操作することで、メッセージを作成し、メッセージを送信する。

3.4.2 Step 2: 移動先の再考と移動の実行

Step 2では、決定された移動先に基づいて、課題環境上のキャラクターを移動させる（図2の3番目の画面に対応）。移動に先立ち、ACT-Rエージェントはパートナーのメッセージに基づいて、移動先を再考する。*JUDGE-PARTNER-MESSAGE*によって、Step 1において予測されたメッセージ *partner-left-symbol-predict*, *partner-right-symbol-predict*と実際に得られたメッセージ *partner-left-symbol*, *partner-right-symbol*を照合する。

照合がとれなかった場合、モデルは再度事例を検索する。Step 2における事例の検索は、検索の手がかりの異なる2つのプロダクションによって行われる。まず、ゴールバッファ中の自分の現在位置、自分のメッセージ、他者のメッセージを手がかりとして事例を検索することを試みる (*RETRIEVE-DESTINATION-FROM-ALL-SYMBOLS*)。これに失敗した場合、自分のメッセー

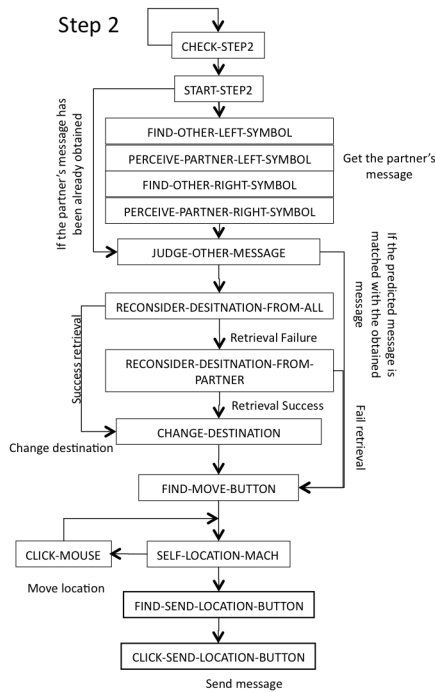


図 5 Step 2におけるプロダクションの発火系列 .

ジを検索の手がかりから除き，自分の現在位置と他者のメッセージを手がかりとした事例の検索を試みる (*RETRIEVE-DESTINATION-FROM-PARTNER-SYMBOLS*) . これらのプロダクションによる事例の検索に成功した場合，検索された事例に基づいて移動先が変更される (*CHANGE-DESTINATION*) . 検索に失敗した場合には，移動先の変更がなされない .

3.4.3 Step 3: 結果の認識と学習

Step 3は，課題環境に表示されたパートナーのキャラクターの初期位置，移動先をゴールバッファに格納する . 格納の完了後，ACT-Rエージェントは課題環境上のFinishボタンをクリックし，ラウンドを終了させる . Finishボタンのクリックを駆動するプロダクションは，ゴールバッファにおける自分とパートナーの移動先の状態 (*self-loc-post*, *partner-loc-post*) によって変化する . 自分とパートナーの移動先が一致した場合，*CLICK-FINISH-BUTTON-LEARN*が発火し，一致しなかった場合は，*CLICK-FINISH-BUTTON-NO-LEARN*が発火する .

CLICK-FINISH-BUTTON-LEARN は，現在のゴールバッファの状態を宣言的モジュールに格納し，*CLICK-FINISH-BUTTON-NO-LEARN* は，ゴールバッファの状態を宣言的モジュールに格納せずに，リセットする .

また，いずれのプロダクションが発火した場合であっても，そのラウンドで発火したプロダクション

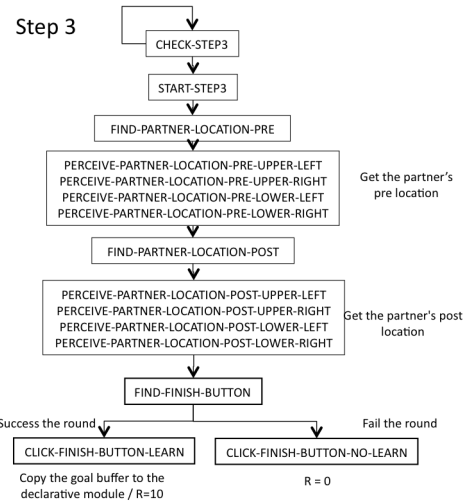


図 6 Step 3におけるプロダクションの発火系列 .

の効用値が，強化学習によって更新される . ACT-Rにおけるプロダクション*i*の効用値 (U_i) は，式1によって制御される .

$$U_i(n) = U_i(n - 1) + \alpha[R_i(n) - U_i(n - 1)] \quad (1)$$

α は，パラメータによって設定される学習率であり， $R(n)_i$ は時刻*n*においてプロダクション*i*が受ける報酬値である . 学習率としてデフォルトの $\alpha = 0.2$ を用い，プロダクションの効用値の初期設定は，先行研究 (Lovett & Anderson, 1996) を参考に $U(0) = 10$ とした .

この式による効用値の更新は，そのラウンドにおいて発火した全てのプロダクションに対して適用される . 本研究では，*CLICK-FINISH-BUTTON-LEARN*が発火した際に， $R = 10$ の報酬値を与え，*CLICK-FINISH-BUTTON-NO-LEARN*が発火した際に， $R = 0$ の報酬値を与えた . つまり，ACT-Rエージェントは，自分とパートナーの移動先が一致した場合に正の報酬を受け取り，一致しなかった場合に実質的に負の報酬を受け取った .

3.5 プロダクションコンパイル

3.4.3に示されるように，ACT-Rエージェントは，ラウンド終了時に事例を獲得し，プロダクションに対する報酬を得る . これらの学習に加え，ACT-Rエージェントは，プロダクションコンパイルにより，新たなプロダクションを学習する . 以下は，プロダクションコンパイルによって構成されるプロダクションの例である .

```
(P PRODUCTION6
"RETRIEVE-CASE&FILL-SLOTS-FROM-CASE-G1-0"
=goal>
```



```

ISA coordinate
self-loc-pre upper-right
self-left-symbol nil
self-right-symbol nil
state start
step 1
==>
=goal>
self-loc-post lower-right
self-left-symbol "square"
self-right-symbol "square"
partner-left-symbol-predict "triangle"
partner-right-symbol-predict "blank"
state start
)

```

このプロダクションは、*RETRIEVE-CASE*と*FILL-SLOTS-FROM-CASE*の系列的な発火からコンパイルされた。より正確に言えば、*RETRIEVE-CASE*が、チャンク *G1-0*を宣言的モジュールから検索し、*FILL-SLOTS-FROM-CASE*によって、ゴールバッファを書き換える」という一連のイベントから生成された。

上記プロダクションに示される条件節(==>の前の部分)は、*RETRIEVE-CASE*から引き継がれたものであり、実行節(==>の後の部分)は*FILL-SLOTS-FROM-CASE*を引き継ぐものである。また、もともとのプロダクションにおいて変数であった部分(*self-loc-pre*, *self-loc-post*, *self-left-symbol*, *self-right-symbol*, *partner-left-symbol-predict*, *partner-right-symbol-predict*)が、コンパイル時に束縛された値によって置き換えられている。この値は、チャンク *G1-0*からコピーされたものであり、過去の成功事例を反映している。

このように、プロダクションコンパイルは、宣言的知識の検索をスキップし、環境に対してダイレクトに反応するプロダクションを生成する(宣言的知識の内容が手続き化される)。このようなコンパイルは、宣言的知識の検索が成功する度におきる。コンパイルされたプロダクションの効用値は、初回のコンパイル時は低いですが、繰り返し同様のコンパイルが起きる度に強化されていく。その結果、宣言的知識の検索がスキップされるようになり、初期位置やパートナーのメッセージの提示などの環境の変化に対する速やかな反応が形成される。

3.6 モデルによって説明されるコミュニケーションシステムの成立

ここで示したモデルの特徴は、複数の学習メカニズムが関与する段階的な学習にある。

事例が蓄積されていない初期のラウンドにおいて、ACT-Rエージェントは、競合解消に基づく決定(図4右のパス)のみを行う。競合解消に基づく決定が成功を導いた場合、その決定に関与したルールが強化され、共通の習慣(メッセージ選択、移動先の決定)が形成されるようになる。

共通の習慣の形成と同時に、モデルは成功事例を蓄積していく。蓄積された事例は、シンボルと状況を対応付ける知識である。この知識が貯えられることで、事例の検索による決定(図4左・中央のパス)がなされるようになる。さらに多くの事例が蓄積されていく中で、パートナーのメッセージを考慮した移動先の再考がなされるようになる(図5)。

また、宣言的知識の検索は、プロダクションコンパイルによるルールの生成を導く。繰り返し同じ宣言的知識が使用されることで、環境やメッセージに対する素早い反応が形成される。この結果として、実験において観察されたメッセージ交換の円滑化が生起する。

このように、本研究において構築したモデルは、シンボルコミュニケーションシステムの形成と習慣的行動の形成やコミュニケーションの円滑化などの協調的行動の形成プロセスを説明するものである。そして、本研究によってなされた、このことの説明は、ACT-Rの持つ複合的な学習メカニズムを十分に生かしたものになっている。

なお、ACT-Rをコミュニケーションシステム成立のアーキテクチャとして利用した研究は、本研究が初めではない。Reitter (2010) は、本研究と同様、ACT-Rを用い、記号システムの成立に関わるシミュレーションを行っている。彼らの用いた課題は、コンセプトをパートナーに伝えるグラフィカルなシンボルを生成する課題(Garrod et al., 2007)であった。Reitter (2010) のモデルは、基本的に宣言的モジュール内部の学習(ベースレベル学習)に基づくものであり、本研究が用いたような複合的な学習メカニズムは用いていない。

また、Reitter (2010) の用いた課題は、2者間での役割が固定化されたもので、本研究が用いたような自発的な役割分担が必要になる課題ではない。本研究のモデルは、図4に示されるように、2者間での役割調整を行う機構をもっている。Step 1におけるパートナーメッセージの取得に関わるルールは、自身の位置のみに基づいて事例を検索する

ルールと競合する。パートナーのメッセージの取得に関わるルールが、発火し続けることは、そのACT-Rエージェントがパートナーのメッセージを待ち続けることに対応する。

だが、このような役割調整の機構を設けることで、両者がパートナーのメッセージを待ち続ける状況が生起する危険も生じる。予備的なシミュレーションの結果、実際にそのようなデッドロックが生起することを確かめている。このデッドロックを回避するためには、競合解消のノイズパラメータを大きく設定するか、*RETRIEVE-CASE*、あるいは*PERCEIVE-PARTNER-LEFT-SYMBOL*に関わる効用値の初期設定をACT-Rエージェント間で変える必要がある。このようなパラメータの初期設定の妥当性については、今後検討する必要がある。

4. まとめ

本稿では、先行研究と異なり、メッセージに用いられる記号の種類、送信のタイミングが制限された実験環境を用意した。実験の結果、習慣の形成やメッセージ交換の円滑さが記号コミュニケーションの成立と関連することが示された。

また、この課題を解決するために構築を進めているACT-Rによるモデルを示した。本稿ではモデルの実行結果を示すことができなかったが、現在までに実行可能なモデルを構築している。当日の発表では設計されたモデルを用いたシミュレーション結果を示したいと考えている。

謝辞

本研究は文部科学省科学研究費補助金(課題番号 21120011)の助成を受けた。ここに記し謝意を表します。

参考文献

- [1] Anderson, J. R. (2007). How Can the Human Mind Occur in the Physical Universe? Oxford University Press, USA.
- [2] Galantucci, B. (2005). An Experimental Study of the Emergence of Human Communication Systems. *Cognitive Science*, 29(5), 737-767.
- [3] Garrod, S., Fay, N., Lee, J., & Oberlander, J. (2007). Foundations of representation: Where might graphical symbol systems come from *Cognitive Science*, 31, 961-987.
- [4] 金野 武司, 森田 純哉, 橋本 敬. (2011). 調整課題における記号コミュニケーションシステムの形成実験. 電子情報通信学会技術研究報告言語理解とコミュニケーション研究会(NLC2010-39), No.400, Vol.110, pp.49-54.
- [5] Lovett, M. C., & Anderson, J. R. (1996). History of success and current context in problem solving. *Cognitive Psychology*, 31, 168-217.
- [6] Lebiere, C., Gonzalez, C., & Martin, M. (2007). Instance-based decision making model of repeated binary choice. *Proceedings of the 8th International Conference on Cognitive Modeling*.
- [7] Newell, A. (1994). *Unified theories of cognition*. Harvard Univ Press.
- [8] Reitter, D. (2010). How groups develop a specialized domain vocabulary: A cognitive multi-agent model. *Cognitive Systems Research*.
- [9] Scott-Phillips, T., & Kirby, S. (2009). Signalling signalhood and the emergence of communication. *Cognition*.