

即応エージェントによる知的コミュニケーション

大澤一郎

通産省 電子技術総合研究所

概要

ダイナミックに変化していく状況を利用して、知的なコミュニケーションを行なう、新しいコミュニケーション行為のモデルを提案する。モデルのベースとなるエージェントにはその時々状況に応じてリアルタイムで反応する即応エージェントを用い、他のエージェントが知っている情報などのように単純なセンサでは観測不可能な情報を適切に取り扱うことのできる「因果シミュレータ」をエージェント内部に導入している。そして、状況依存性を本質とする行為の定式化に基づいて、マルチモーダルなコミュニケーション行為のモデリングを行ない、口と耳の協調によるフィードバック制御を利用した自然な知的コミュニケーションを実現する。

How can a Reactive Agent communicate with People

Ichiro Ohsawa

Electrotechnical Laboratory

ABSTRACT

We present a new computational model of reactive agent to communicate with people, in which the situation around the agent at each moment produces a great effect on the choice of the acts executed next. The architecture of agent is extended by introducing the massively parallel simulator of causal changes, so that the agent can easily manipulate the information such as intentions and beliefs, which can not be observed with any sensors. As a result, we realize an intelligent communication, which utilizes the feedback control between mouth and ear.

1 はじめに

人間社会を自由に動き回る自律ロボットが容易に製作あるいは利用できるようなにつれ、自律ロボット同士、あるいは自律ロボットと人間とのコミュニケーションに関する研究がますます重要になりつつある。すなわち、万能ではない自律ロボットが人間社会(あるいは、人間とロボットの共存社会)でうまく行動していくためには、他のエージェント(自律ロボットや人間など)をうまく利用することが不可欠である。そこで、必要に応じて他のエージェントに何らかの情報を伝達し、それによって他のエージェントの行動を自分の都合の良いように制御したい。例えば、自律ロボットが転んで起き上がれなくなってしまった場合に、周囲に大声で助けを求めることで、近くにいる他のエージェントがその自律ロボットを起こすようにしたり、何らかの情報が必要な場合に、それを知っていそうな他のエージェントに問いを発することで、そのエージェントがその情報を発話するようにしたい。

他のエージェントを外部から制御する手段としては、リモコン飛行機などのように、ある特定の行動パターンが自動的に生じるような仕組みをエージェント内部にあらかじめ組み込んでおき、それを外部からの簡単なプロトコルで起動するというのが一般的である。人間というエージェントに対しても、信号機などのように、簡単なプロトコルでその行動を制御することができる。しかしながら、不特定エージェントの多様な行動を必要に応じて制御するには、あらかじめ決められた簡単なプロトコルだけでは不十分で、日本語などの高度なプロトコルを利用したコミュニケーションが必要になってくる。

知的コミュニケーションとは、このような高度なプロトコルを利用したコミュニケーションにおいて、動的に変化する状況を上手に利用して、効率の良いコミュニケーションを行なうことを意味していると考えられる。例えば、相手の持っている情報や目標、相手の周囲の状況などに関する情報に基づいて、

- 発話しなくても良いことは発話しない
- 参照可能な時空位置にある物体に対しては「これ」「それ」などの指示代名詞を使う

などの手法により、効率の良いコミュニケーションを実現する。相手の推論パターンを利用して、仕事を婉曲に依頼するというのも知的コミュニケーションの一手法と考えられる。

そして、このような知的コミュニケーションを日常の行動において実現するためには

- 外部から多様な情報をリアルタイムで取り込み、
- 取り込んだ情報の統合などをリアルタイムで行ない、
- 大量の情報に基づく動作選択をリアルタイムで行なう

という計算が必要不可欠である。しかしながら、これまでの古典的 AI(論理指向 AI) に基づく自然言語対話モデルでは、論理的に矛盾のない行動を時間をかけて実行することは可能でも、リアルタイムに行動することは困難であった。そこで我々は、動的に変化する状況を利用して効率の良いコミュニケーションを行なう自律ロボットを実現するために、因果シミュレータを組み込んだ即応エージェントアーキテクチャ [2]¹を採用している。

以下では、我々が提案している即応エージェントアーキテクチャ上でどのようにして知的コミュニケーションが実現されるかに関して論じていく。2 節では、我々の即応エージェントアーキテクチャ

¹etlport.etl.go.jp/pub/kyocho/Papers/PostScript/MACC93.ps より anonymous ftp 可能

を概説し、3節で、そのエージェントアーキテクチャに基づいたコミュニケーション行為の実現手法を示す。そして4節で、口と耳とのあいだのフィードバック制御に基づく知的コミュニケーションのモデルを紹介する。

2 即応エージェント

2.1 エージェントアーキテクチャ

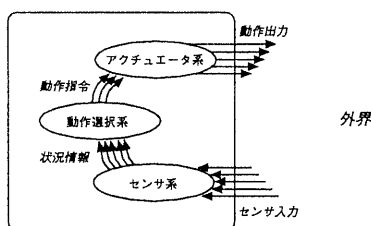


図 1: 基本的な即応エージェント

即応エージェントとは、動的に変化する状況に即応して動作するエージェントのことで、図1のように

- 多種多様な状況情報をリアルタイムで取り込むセンサ系と、
- 大量の情報に基づく動作選択をリアルタイムで行なう動作選択系と、
- 状況を変化させるアクチュエータ系

から構成される。そして、

1. その時の状況情報が、即応エージェントに取り付けられている超音波センサなどの各種センサを通じてセンサ系に入力される
2. センサ系は、センサから得た状況情報を、必要に応じて加工あるいは統合して、動作選択系に送る
3. 動作選択系では、センサ系から送られてきた状況情報に基づいて今すぐ実行する動作(群)を決定し、決定した動作(群)の実行指令をアクチュエータ系に伝達する
4. 実行指令を受けたアクチュエータ系はただちに指令された動作(群)を実行する
5. 動作(群)実行の影響などによって外界が変化する

というサイクルが短時間のうちに次々と繰り返される。

このようにして、即応エージェントは動的に変化する状況に即応して適切に動作する。しかしながら、上記の単純な即応エージェントには内部記憶が全く存在しないので、センサから得ることのできない状況情報(相手の保持している情報や目標に関する情報など)に基づく高度な動作系列(知的コミュニケーションなど)を実行するにはあまり適していない²。

²外部をうまく利用すれば実現可能であるが効率が悪い。

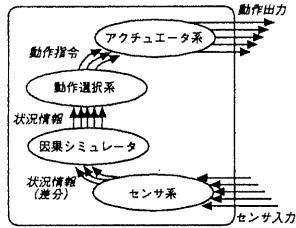


図 2: 即応エージェント + 因果シミュレータ

2.2 因果シミュレータ

我々は、上記の問題を解決するために、図 2 に示すように、センサ系と動作選択系のあいだに「因果シミュレータ」という新しい系を導入している。因果シミュレータは、直前の状況情報を系内に記憶しており、センサ系から何らかの変化 (群) が生じたという情報を受けとると、因果関係 [1] に基づく変化の拡散伝搬 [3] を、超並列ネットワーク計算によりリアルタイムでシミュレートする。例えば、即応エージェント自身が右向きに 90 度回転したという変化情報がセンサ系から入力されると、その変化から因果的に生じる、周囲の物体との相対位置関係の変化などの情報をリアルタイムで計算する。

因果関係としては、典型的な状況に関して、何らかの変化あるいはイベントの生起によって生じる因果変化を

`situation(状況, [因果関係1, 因果関係2, ...])`

という形式で記述しておく。因果関係₁ は、

`caused(因果変化, 状況)`

という形式で定義し、状況は

- 一つ前の時点で成り立っている命題、あるいは生起したイベント
- 一つ前の時点で成り立っていない命題、あるいは生起していないイベント
- 現時点で成り立っている命題、あるいは生起したイベント
- 現時点で成り立っていない命題、あるいは生起していないイベント

の四つ組 (およびオプションな計算項) で表現する。例えば、自分の左側に物体 X があるという状況で、自分自身が右に 90 度回転するというイベントが生起すると、物体 X の相対位置は自分の後方に変化するという因果関係は次のように定義できる (ただし、i は即応エージェント自身を、述語の前の + はちょうど今ということを表している)。

```
situation([left_of(i,X)],[],[],[[]],[
    caused(back_of(i,X),[],[],[+done(i,[rotate,-90])]),[],
    .....
]).
```

因果シミュレータは、変化しない部分に関しては直前の状況情報をそのまま現在の状況情報として継承することができる。例えば、上記の右 90 度回転の場合、周囲の物体の色や形などに関する情報

は、変化したという情報がセンサ系から入力されない限り、直前の情報をそのまま継承する。実際にはどのような種類の情報を継承するかは因果シミュレータの設定に依存しているが、情報の更新がない場合には指定された時間だけ旧情報を継承するという設定も可能である。そして、このような継承計算も上記の超並列ネットワーク計算のなかで、因果伝搬計算と同時に実行され、結果として得られたその時の状況情報が動作選択系に送られる。

2.3 意図と信念

これまでに述べてきた因果シミュレータの機能を利用することで、即応エージェント自身の意図や信念を制御することができる。すなわち、外界あるいはエージェント内部で生じる変化あるいはイベントから因果的に、即応エージェントの意図や信念に変化が生じると考えて、即応エージェントの意図や信念に関する情報を因果シミュレータ内部で管理する。例えば、即応エージェント自身のお腹が空いたという情報をセンサ系から得た因果シミュレータは、何かを食べたくなるという自分自身の意図の変化を因果伝搬計算により導き出す。そして、その自分自身の意図に関する情報を(何かを食べて空腹が癒されるまで)継承していくことで、即応エージェント自身の何かを食べたいという意図を因果シミュレータ内部に保持し続けることができる。すなわち、保持し続けている間、その意図を動作選択に反映させることができる。信念に関しても全く同様である。ただし、三段論法などのような論理的推論を経て導き出される高度な意図や信念に関しては、因果シミュレータ内部でローカルに導き出さずに、即応エージェントの動作サイクルを何段階か経ることで創発的に生じさせる。例えば、紙と鉛筆を使って外界と密に相互作用することで鶴亀算の答を紙の上に生じさせるという様である。

2.4 動作と行為

言葉の定義の問題であるが、我々は、「行為」はアトミックなものではなく、ある状況下でエージェントがある「動作」を実行した時に生じるものと考えている。例えば、サッカーボールを蹴る一連の動作でも、前にボールがあればボールを蹴るという行為になるが、前にボールがなければ単に足を振り上げるという行為にしかならない。したがって、ある行為を行なおうとしても、状況認識に誤りがあれば、全く異なる行為を行ってしまうということがあり得る。

さて、我々の即応エージェントは、センサ系と因果シミュレータから得られた状況情報に基づいて、いまずぐ実行する動作を選択する。動作選択メカニズムは、因果シミュレータで用いたメカニズムをそのまま利用している。したがって、動作選択の方式としては非常にナイーブであるが、現時点の状況情報だけではなくて、一つ前の時点の状況情報も動作選択の際に参照できるという点で他の方式よりも新しい。

以下は、動作選択系における定義例であるが、ここではわかりやすいように、別途開発中のサッカープログラムから定義を引用する。

```
situation(([], [], [ball((10,Y))], []), [
    caused(exec([kick,f]), [], [], [], [], (abs(Y)<10)),
    caused(exec([move,r]), [], [], [], [], (Y>0)),
    caused(exec([move,l]), [], [], [], [], (Y<0))
]).
```

この定義は、ある一人のフィールドプレイヤーのプログラムで、ballという関係はボールとの相対距離を表している。スタジアムは600×400程度の整数値で表現され、プレイヤーやボールの大きさは10×

10である。したがって、上記の定義で、ボールがX軸上で10単位離れているという状況は、プレイヤーの直前、あるいはその横方向上にボールが位置していることを意味する。そして、その状況においてY軸上で絶対値が10未満の位置、すなわちプレイヤーのほぼ直前にボールが位置していれば蹴る動作を実行し、Y軸上で正(あるいは負)の位置、すなわちプレイヤーのほぼ左(右)にボールが位置していればそちらの方向に移動するという動作を実行する。なお、各時点で選択される動作は一般に複数で、資源競合が起こらない限り、選択された全ての動作が実行に移される。上記の例であれば、蹴りながら移動するという複合動作もあり得る。資源競合が生じる場合には、競合した動作集合の中からランダムに一つの動作が選択される。

3 コミュニケーションの実現

3.1 発話動作

言葉を喋るというのも、歩いたり食べたりするのと全く同様に、自律ロボットが実行できる動作の一つである。すなわち、日本語などを発話する動作は、声が届く範囲に他のエージェントが存在している状況では、そのエージェントに何らかの情報を伝達するという行為を生む。したがって、あるエージェントに何らかの情報を知らせたいという意図があつて、そのエージェントが目の前にいる状況では、その情報を発話するという動作を実行すれば良い。我々の即応エージェントは、以下のような動作定義によって、この発話動作を実行している。

```
situation(([],[],[speaking(i,J)],[]), [
    caused(exec([say,PRDC]),[],[],[goal(i,bel(J,PRDC))],[]),
    .....
]).
```

すなわち、あるエージェントJと話している状況で、PRDCという情報をそのエージェントに知らせたいという意図があれば、その情報を発話するというを上記の定義は表現している。ただし、実際の定義では、対話の流れをある程度制御しなければならないので、もう少し条件が付与されている。なお、あるエージェントに何らかの情報を知らせたいという意図があつても、そのエージェントが目の前にいないような状況下では、そのエージェントの位置を知っていればその付近に移動するなどの一般的な動作系列が即応的に実行される。

3.2 相手の意図と信念

コミュニケーションの相手が持っている意図や信念に関する情報は、知的コミュニケーションを行なう際の最重要ファクターの一つである。我々の即応エージェントアーキテクチャでは、この情報を因果シミュレータ内で計算し、管理している。例えば、他のエージェントがそばにいて、自分の発話に耳を傾けているという状況で何かを発話すれば、因果的に相手はその発話内容を聞くことになる。そして、その発話内容を聞いた相手のエージェントは、即応エージェントのことを信頼していれば、因果的にその内容を信じることになる。以下がその因果関係の定義である。

```
situation(([],[],[speaking(i,J)],[]), [
    caused(+done(J,[hear,i,PRDC]),[],[],[+done(i,[say,PRDC])],[]),
    caused(bel(J,PRDC),[],[],[+done(J,[hear,i,PRDC]),reliable(i)],[]),
    .....
]).
```

そして、相手のエージェントが、ある状況で何かを意図している、あるいは何かを信じている場合、同時に信じるようになると考えられる別の意図や信念に関しては、

```
situation(([],[],[状況],[]), [
    caused(goal(J, 命題 2), [], [], [goal(J, 命題 1)], []),
    caused(bel(J, 命題 2), [], [], [bel(J, 命題 1)], [])
]).
```

という形式で、意図や信念に関する因果関係を定義している。

4 フィードバック制御 — 口と耳の協調

即応エージェントは、状況を介した動的なフィードバック制御により、個々の状況に応じて最適な動作系列を実行することができる。すなわち、即応エージェントは、個々の独立した動作定義から、状況との相互作用によって動的に適切な動作系列を生成することができる。そこで、「口と耳の協調」によるフィードバック制御を利用することで、状況に応じた適切な日本語文が発話可能なことを簡単な例によって示す。

ここでは、コミュニケーションの相手 J に今の時刻 (9 時 45 分) を知らせたいと仮定する。すなわち、我々の即応エージェントは

```
goal(i, bel(J, is(now, 9 : 45)))
```

という意図を持っており、現在 J と対話中であるとする。すると、3.1 節に示した簡単な発話動作の定義によればすぐさま `is(now, 9 : 45)` という内容の発話動作を実行してしまうが、ここではその動作定義を用いずに、次の動作定義を利用することにする。

```
situation(([],[],[speaking(i,J),goal(i,bel(J,is(X,Y)))],[]), [
    caused(exec([say,focus(X)]),[],[],[],[focus(X)]),
    caused(exec([say,is(*,Y)]),[],[],[focus(X)],[]),
    .....
]).
```

すなわち、コミュニケーションの相手に `is(X,Y)` という形式の内容を知らせたい場合、その時点の焦点が X でなければ「X は」と発話し、その時点の焦点が X であれば「Y である」と発話する。

さらに、誰かが「X は」と発話した場合に生じる焦点の変化を以下のような因果関係を用いて因果シミュレータで管理する。

```
situation(([],[],[speaking(i,J)],[]), [
    caused(+focus(X),[],[],[+done(J,[say,focus(X)])],[]),
    .....
]).
```

上記の定義は、即応エージェントと対話中のエージェント J が「X は」と発話したとすると、因果的に焦点が X になるということを表現している。そして、即応エージェントは自分自身の発話に耳を傾けているという命題 `speaking(i,i)` は常に成り立っているとする。すると、我々の即応エージェントは、自分が「X は」と発話すると、それを耳で聞き、焦点が X になったことを認識することになる。

以上の設定により、我々の即応エージェントがコミュニケーションの相手 J に今の時刻 (9 時 45 分) を知らせたい場合、

即応エージェント	ユーザ (人間)
「おはようございます」 「手紙が来ています」 手紙の差出人を調べる ... done 「ohsawa さんからです」 「9 時 45 分です」 「またね」	「誰からですか」 「それは」 「あっ」 「今は」 「何時ですか」 「うっ」 「まずい」 「またね」

図 3: 試作システム RASCAL による対話例

- (1) 焦点が”今”になっていないので「今は」と発話し、
- (2) 「今は」という発話を聞いて焦点が”今”になったことを認識し、
- (3) 焦点が”今”になっているので「9 時 45 分です」と発話する

というフィードバック制御によって、「今は」「9 時 45 分です」という適切な発話系列を実行する。そして、最後の発話によって、J が今の時刻を知ったと認識し、それにより J に今の時刻を知らせたいという意図が因果的に消滅することになる。

なお、この節の冒頭で、即応エージェントは状況を介した動的なフィードバック制御により、その時々状況に応じて最適な動作系列を実行することができると述べたが、即応エージェントが J に今の時刻を知らせたいと思った時点ですでに焦点が”今”になっていたとすると、上記の動作系列のうちで(1)と(2)は全く考慮されずに、すぐさま(3)が生じることになる。すなわち、「今は」と主語を発話せずに、「9 時 45 分です」とだけ発話することになる。この現象は、図 3 に示す試作対話システム rascal の対話例のように、相手から「今は何時ですか」と尋ねられた場合などに生じる。

参考文献

- [1] M. P. Georgeff. Actions, processes, and causality. In *Reasoning about Actions and Plans*. Morgan Kaufmann, 1987.
- [2] 大澤 一郎. 人間と対話する即応エージェントのモデル. マルチエージェントと協調計算に関するワークショップ (MACC'93), 1993.
- [3] 大澤 一郎, 中島 秀之. 自律エージェントのための行為の定式化. 石田亨 (編), マルチエージェントと協調計算 II. 近代科学社, 1993.