

## 模擬の記述の一手法

防衛庁 君島 浩

模擬(simulation)の企画・発注・開発などを円滑にすることを目的として、模擬をどう記述するかの手法のごく一部を述べる。模擬はあらゆる種類の学術分野に存在する。模擬には複雑な計算やプログラムを用いる場合がある一方で、できるだけ簡潔に記述したり処理したりすることも必要である。膨大な手法の中の教育学及び医学と関係が深い部分を抽出して紹介する。

### 個性(individuality)

個人、その他の個々の生物、あるいは個々の集団の特性の実例の集合である。教育学的には個性は特定の個人の特定の時期の能力である。能力の個人版と言ってもよい。生理学や病理学的には個性は個人の身体検査値、健康診断値、傷病記録に代表される。

以下は個性の類似語・関連語であり、少し異なる意味として使われる。

性格(character) 「性」が種類、「格」が度合を表すものと思われる。

品性(character) 能力種類の中の態度(情意)の部分を意味する。

人物(character) 性格の意味から転じて、ゲームなどの登場人物を表す。

人格(personality) 人の性格である。

能力(capability) 個人によらない共通の定義である。

特性(characteristics) 個性を構成する一つの項目。単数もsが付く。

個性は教育によって変化する知識・心身技能・態度の能力だけではなく、教育以外の遺伝や衣食住や日常行動などによって変化する身長などの特性、あるいは歓迎されざる傷病や老化のような変化、及び医薬や介護によって回復・軽減される特性も含む。

個性という言葉が表す特性の種類は膨大であり、その度合の幅は極めて広い。人間の脳をコンピュータに換算すると、単純計算でも数十億個の中央処理装置CPU及び数万GB(ギガバイト)の記憶装置に相当するとされ、それに人体の器官の特性を加味すると、各人が記憶している言葉の数や遂行できる運動の種目数や成績は千差万別である。模擬の時にどう分類・整理するかが大切である。

特定の個人が所有している個性は、全能という状態からはほど遠いことが当然だが、多大な可能性を持っている。各年齢での個性を尊重するとともに、どういう種類の特性をどの度合まで伸ばすかは即断・断念を避けるべきだ。

一般的な定義としての職務の個人版が職責であるのと同様に、一般的な能力の定義に対する個人版が個性である。逆に言えば、各人各様の膨大な個性の事例を、学者などが丹念に収集して大同小異を見抜き、分類や尺度を整理したも

のが、一般的な特性や能力である。

#### データの格納場所と値

模擬ゲームにおける登場人物の個性は固定にしている場合もあるが、衣食住や教育や経験に応じて変化させる場合もある。次第に身長が伸びるとか、空腹になったら体力が減少するとか、勝利経験によって自信が増すなどである。この方式はゲームの変化を富ませるだけではなく、ゲームの処理を簡潔にする効果ももたらす。

- 一般的な特性の種類をデータの格納場所の枠組とする。
- 特性の実例を、格納場所に入れるデータの値として変化させる。

どの人物も値こそ違うが格納場所の形式は共通なので、一つのプログラムで処理できる。人物ごとにプログラムを組む必要がない。

#### 能力の分類と成績

ただでさえ膨大な個性をできるだけ簡潔に扱うために、能力の枠組には信頼できるモードを活用する。各人の成績は成績基準に照らして処理する。人事学の職種・職級などの縦割り、横割りの枠組を能力の枠組にするとよい。また、ガニエやブルームなどの学習領域の横の分類や縦の水準の枠組を能力の枠組にするとよいのは言うまでもない。粘り強さなどの情意能力（態度・気力・道徳）も教育学に基づいて扱うべきである。人事学の用語を用いてウェブ検索をすると、ゲームソフトの内部設計書がヒットすることがある。ゲームソフトの個性の制御に人事学が適用されているからである。

#### 能力以外の特性の分類と度合

身長や血圧などの教育対象外の特性には、生理学や病理学の学術的な分類や尺度を活用する。

#### 状態遷移 (state transition)

プログラムを単純にする工夫の一つが状態遷移という関係論である。個性を有限の個数の状態の集合として定義する。食事、教育、経験、傷病、治療などの、ある状態から他の状態へ変化させる有限の遷移を定義する。前述のデータの格納場所と値が状態に相当する。例示としては自動販売機やビデオデッキの例がよく用いられる。ビデオデッキの静止状態を再生状態へ遷移させるのが再生ボタンを押すことである。再生状態を休止状態へ遷移させるのが休止ボタンである。血圧などの状態にはアナログ値を用いてもよい。

#### 状態遷移図 (state transition diagram)

状態遷移は状態遷移図という流れ図の一種の有向グラフとして記述する。状態を節（ノード）で表し、ある状態から別の状態への遷移を矢印付きの弧（ア

ーク)で表す。節の名称は節を表す四角や円などの枠の中に記述し、遷移の名称あるいは計算方法などは弧のすぐそばに記述する。ビデオデッキの例で状態遷移図を書く練習をするとよい。

状態遷移は直前の状態だけしか直後の状態に関係しないという模式なので、模擬の定義を整理して記述するのに適している。格別の制限のない一般的な流れ図を用いれば、模擬の可能性が広がるが、その代わり専門的な設計技能が要求される。状態遷移図であれば主題の専門家と模擬装置を開発する専門家との間で議論したり発注したりする資料として適当である。

#### マルコフ過程 (Markov process)

ある有限回以前にさかのぼった回からは影響を受けないという確率過程をマルコフ過程と言う。直前の1回の状態にしか影響を受けないのを単純なマルコフ過程と言い、n回の状態に影響を受けるのをn重のマルコフ過程と言う。

直前(1回前)の検温結果が38度だとして、2回前の検温結果が35度なら症状悪化であるし、2回前の検温結果が40度だったのなら症状は回復傾向と言える。このような場合には、状態遷移にn重のマルコフ過程の考え方を加える。模擬ソフトとしては各状態に身上歴・病歴・会社四季報などのn回分の格納場所を設けて、そこへ値を入れて先入れ先出し(first-in, first-out)の規則で古い値から消していくのである。

#### 乱数生成 (random number generation)

模擬において不確実な現象の一つが生じることの代替手段として、乱数を生成させて用いることがある。確率が一樣である場合には一樣乱数を用いる。確率事象が一樣でない分布になるなら、それぞれの分布曲線に合せた乱数を用いる。一樣乱数を生成した後にそれを分布関数やファジー関数で加工してもよい。

例えば、記憶が時間とともに薄れていくという現象は、「 $Y = -X + 1$ 」という単純な分布関数に合せて0~1の範囲の乱数を生成してもよい。乱数生成はコンピュータのプログラミング言語が関数として用意しているし、除算などで自分で生成してもよい。

#### IF-THEN規則 (if-then rule)

複雑な流れ図による処理を単純にするのが人工知能で用いられるIF-THEN規則である。いろいろな状況に応じる処理を「if 状況 then 処理」という形式に統一して一覧表として格納する。模擬ソフトウェアは事象が発生するたびにその状況をキーにしてIF-THEN規則表をいちいち走査して、それに合う処理の子プログラムを実行する。走査の手間がかかるが、プログラムは単純になり、状況の追加・変更は単純な一覧表の書き方を知っているだけでできる。

## 任務管理 ( task management )

小さな仕事の単位である任務が発生するたびに、それに適する職員を割り振るといふ労務管理の方式を任務管理という。次々に発生する事象（任務）を取りあえず待ち行列に並ばせる。任務管理プログラムは待ち行列の先頭の任務を取り出して、それに適する子プログラムを選定して実行させる。子プログラムごとに担当範囲の任務の発生を監視する必要がないので、プログラムが単純になる。

コンピュータを使っている時に、アイコンを次々とクリックするとそれを記憶していて、しばらくして次々と処理してくれるのは、各プログラムが開始時期を決めているのではなく、任務管理プログラムが一括して管理しているのである。任務管理のこの方式は別名行き当たりばったり方式という。大きな単位である業務管理は年間で計画的に、小さな単位である任務管理は行き当たりばったり方式でというのが生産管理の知恵である。