

■7群 (コンピュータ-ソフトウェア) - 7編 (分散協調とエージェント)

1章 エージェントの定義・モデル・概念

【本章の構成】

本章では以下について解説する.

- 1-1 マルチエージェント研究の歴史
- 1-2 エージェントの定義
- 1-3 自律エージェント
- 1-4 合理的エージェント：論理的
- 1-5 合理的エージェント：経済的
- 1-6 社会的エージェント
- 1-7 マルチエージェント
- 1-8 ソフトウェアエージェント
- 1-9 インタフェースエージェント
- 1-10 モバイルエージェント

■7群-7編-1章

1-1 マルチエージェント研究の歴史

(執筆者：中島秀之) [2015年4月 受領]

マルチエージェントという研究分野は、現在ではAAMASという国際会議が開催されているが、もともとは米国、欧州、日本でそれぞれ独立に、しかも少しずつ違う方向性を持って生まれたものである。以下でそれらを概観する。

1-1-1 分散人工知能

年代的に最も早いのはやはり米国であるが、最初は分散人工知能 (Distributed AI) の名で研究されていた。First International Workshop on Distributed Artificialが1980年にMITで開催されている。

もともとは、米国国防省の要請でレーダー網からの情報を統合し、敵味方の判別などを行うために発展した分野らしい。Davisら⁴⁾はコントラクトネットをDSS (Distributed Sensing System) に応用することを示しているが、これは航空管制などにおいて、各地に分散しているレーダーなどの感知装置の情報を統合し、飛行機の感知、判別、追跡などを行うシステムである。地理的に分散するステーションは感知あるいはデータ処理の能力を持ち、これらが協力して単一のエリア地図を作成することが目標である。最終的な情報の統合と制御は一つのステーションが受け持つことになる。

古くからの人工知能 (Artificial Intelligence : AI) の探索問題は、単独プロセッサのもので、複数台のプロセッサで並列探索を行うものでも、基本的にはすべての情報を管理したうえで探索が行われていた。これに対して分散探索問題では互いの探索が独立に行われるため、それらの間の情報統合が問題となる。典型的な例としては、北米レーダー網で捉えた飛行物体の間の同一性判定がある。各々のレーダーは自分中心座標で飛行物体を追跡するため、それらの間の座標変換や、一部のレーダーから隠れた場合の情報補完などが問題となる。

これらの技術要素はそのまま複数のエージェントの協調の問題に使える。米国では独立したエージェント間の交渉が主要課題となっており、現在でもオークションの研究などが盛んである。

様々な問題設定によるマルチエージェント系の性質の解析も行われた¹⁰⁾。例えば、追跡問題²⁾においては、複数のエージェントが単一のターゲット (比較的単純な規則で逃げ廻る) を追跡するときにエージェントの組織としてどのようなものが効率が良いかが解析された。ベンダらによると、命令する指揮エージェントと、互いに通信する部下エージェントの組合せが最も効率が良いらしい。しかし、これは仕事の性質によっても大きく左右されるものであり、すべての場合にこのような軍隊型の組織が良いとは限らないが、Minskyの『心の社会』⁶⁾を含む、米国での多くの研究が軍隊や会社組織に似たものを前提としている。

複数エージェントを含む系の動作をプログラムする際、従来の“処理”あるいは“計算”としてのプログラミング言語の抽象化の方向は適切なものではない。このような場合には自律性を持った独立単位としてのエージェントどうしの、情報の伝達や、共同作業を素直に記述できることが望ましい。そこで、情報の伝達行為を定式化した言語行為⁷⁾の理論や、そもそもそういった行為を引き起こす意図の理論³⁾を基本動作原理としたプログラミング言語/システムも

提案されてきた⁸⁾¹⁾。

1-1-2 欧州の MAAMAW

欧州では First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-89) が、この分野で最初の会議となる。ワークショップの題名からも分かるように、マルチエージェント環境下でのエージェントの自律性が中心となって研究が行われてきた。

なお、欧州ではこのほかに Multi-agent Based Simulation (MABS) Workshop が 1998 年から始まっている。ここではマクロとミクロの融合問題などが議論されている。例えば、社会のシミュレーションと個人のシミュレーションをどのように結合するかというような問題である。すべてを個々の人間のシミュレーションに還元するのは効率が悪いだけでなく、マクロに存在する法則性が捉えられない。この問題はマルチエージェントの分野に限らず、気象のシミュレーション(湖水と水分子)や生命現象のシミュレーション(タンパク質と素粒子)でも同じである。

1-1-3 日本の MACC

日本ではソフトウェア科学会の研究会として筆者らが始めた MACC-91 (Multi Agent and Cooperative Computation)⁹⁾ が最初である。

日本ではやはり「三人寄れば文殊の知恵」というスローガンが表す、個々のエージェントの集まりより強い知能を目指す研究が多い。より良い社会制度を目指すものもある。もちろん、この対比は典型化のし過ぎであり、MACC の立ち上げ当初から、日本の研究者のなかには米国で DAI の指導を受けてきたものも多いし、最近では AAMAS のオークションコンテストの優勝者が日本からも出ている⁵⁾。

1-1-4 ICMAS への統合と AAMAS

3 つの地域別のワークショップは 1995 年に ICMAS (International Conference on Multi-Agent Systems) として統合された。2002 年に、欧州で 1994 年から始まった、Agent Theories, Architecture, and Languages (ATAL) workshop を開催している (マルチでない) 自律エージェント研究コミュニティと合体して AAMAS (Autonomous Agents and Multiagent Systems) となった。AAMAS から、“multi” と “agent” の間のハイフンがとれ、“multiagent” という一単語になっている (これはマルチエージェントが世界的に定着した印である) 点にも注意されたい。

■参考文献

- 1) Mihai Barbucaanu and Mark S. Fox. Cool: “A language for describing coordination in multi agent systems,” in Proc. First International Conference on Multi-Agent Systems, pp.17-24, 1995.
- 2) M. Benda, V. Jagannathan, and R. Dodhiawalla: “On optimal cooperation of knowledge sources,” Technical Report BCS-G2010-28, Boeing AI Center, 1985.
- 3) Michael E. Bratman: “Intention, Plans, and Practical Reason,” 1987. (門脇俊介, 高橋久一郎(訳): “意図と行為,” 産業図書, 1994.)
- 4) R. Davis and R. G. Smith: “Negotiation as a metaphor for distributed problem solving,” Artificial Intelligence, 20(1):63-109, 1983.
- 5) Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors: “New Trends in Agent-Based Complex Automated Negotiations,” Studies in Computational Intelligence, vol.383, 2012.
- 6) Marvin Minsky, 安西祐一郎(訳): “心の社会,” 産業図書, 1990. (原著出版: MIT Press, 1985.)

- 7) 中島秀之(編著)：“マルチエージェントと協調計算：日本ソフトウェア科学会 MACC,” レクチャーノート/ソフトウェア学, 近代科学社, 1993.
- 8) 大沢英一, 沼岡千里, 石田 亨：“サーベイ：分散人工知能小問題集,” in マルチエージェントと協調計算 I, レクチャーノート/ソフトウェア学, 1992.
- 9) John R. Searle, 坂本百大, 土屋 俊(訳)：“言語行為,” 勁草書房, 1986.
- 10) Yoav Shoham：“Agent-oriented programming,” Artificial Intelligence, 60:51-92, 1993.

■7群-7編-1章

1-2 エージェントの定義

(執筆者：木下哲男) [2011年2月 受領]

エージェントという用語は、情報処理、情報通信、そして人工知能/知識工学をはじめとする種々の分野で用いられており、その目的や担うべき役割なども様々であることから、これらを一括りにして定義付けることはできない。そこで、「分散協調」問題解決に焦点を絞って考えると、例えば、問題解決におけるエージェントの振る舞いに着目すれば、「解くべき問題の処理を、単独もしくはほかの処理体と協力して実行する処理体」という定義が考えられる。また、エージェント自身の機能や構造に着目すれば、「センサから得られる外界の情報や自分の内部情報を利用した意思決定機構によって自分自身の動作を決定し、エフェクタを通して外界への働き掛けなどを行う処理体」とみることもできる。

一方、備えるべき特性や特徴を列挙して、エージェントの概念を規定する方法もある。例えば、「自律性、社会性、反射性、自発性、合理性、移動性、…、を備えた処理体」、というのがその一例である。これらのなかで、自律性、社会性、反射性、自発性の4つは、しばしばエージェントを特徴づける主要な特性として取り上げられる。すなわち、「自分自身の動作の目標を設定して動作したり（自律性、自発性）、ほかのエージェントと協力して組織を構成して問題解決を実行したり（社会性）、種々の変化や変動を察知して適応的に動作したり（反射性）する処理体」をエージェントとみなすことになる。

更に、特定の視点を強調した呼称を付与してエージェントを類別することも行われている。例えば、実装形態を強調する「ソフトウェアエージェント」、人間の代行や作業支援といった役割に着目した「ユーザエージェント」、あるいは、応用対象を明示した「インタフェースエージェント」などが挙げられる。

上述したように、本章の標題に掲げた「分散協調」に限った場合でも、エージェントを規定する目的や方法の違いを反映した種々の定義づけができることが分かる。しかし、同時に、これらの定義はそれぞれの文脈において有用であることから、本節では、これらを無理に統合することは避けて、以降の各節において、これまでの研究開発のなかで用いられてきた主要なエージェントの概念、すなわち、「自律エージェント」、「合理的（論理的）エージェント」、「合理的（経済的）エージェント」、「社会的エージェント」、「マルチエージェント」、「ソフトウェアエージェント」、「インタフェースエージェント」、「モバイルエージェント」を取り上げて吟味することにより、エージェントに関する考察を深めることとしたい。

■7群-7編-1章

1-3 自律エージェント

(執筆著者：木下哲男) [2011年2月 受領]

エージェントを特徴づける主要な特性として真っ先に挙げられるのが「自律性」である(1-2節参照)。その意味を辞書的に述べるとすれば、「ほかからの支配や制約などを受けることなく、自分自身で立てた規範に従って行動する性質」であり、これを有することが強調される場合に自律エージェントと呼ばれる。

ここで「規範」とは、エージェント自身の振る舞いを選択したり決定したりする際の基準となる規則や様式などを指す。したがって、自律エージェントには、こうした規則や様式を自ら保持して利用する、更には、その開発や設定なども自ら行えるような知性や能力(高度な問題解決能力)が要請される。ただし、自らの規範に従って行動するという事は、ほかのエージェントを無視して勝手気儘に振る舞うことを意味しているわけではない。

一方、自律性の基盤となる知識や機能を自ら獲得して利用していくのは、人間でも容易でないことが往々にしてある。まして人工物であるエージェントに対して、こうした能力を期待するのは、現状では極めて困難である。そこで、自律エージェントの設計開発では、実現可能な程度、あるいは、扱う問題解決において必要となる範囲での自律性の実現を目指す。つまり、自律エージェントに対する規範などは、エージェント設計者が設定/付与することにする。

すなわち、設計者は、所与の問題や問題解決に関する要求や条件を考慮して、エージェントの動作とその制御を行うための知識(エージェントの意思決定機構が利用可能な形式で表現された知識)、及び、機能(エージェントのセンサやエフェクタなどの実現に関わる種々の機能)を設計し、これらを備えたエージェントを実装/実現する。これにより、設計目標のエージェントは、対応可能な問題解決の過程で、自分自身が保持している情報や知識を参照/利用して、解くべき問題の設定、問題解決(意思決定)処理、解の生成/提示などに対応した様々な処理や動作を、自ら決定して実行していけるようになる。

なお、自律エージェントの設計と実装に関する具体的な方法について議論する際には、エージェントが利用する知識の表現形式、推論エンジン(意思決定機構)の処理方式、あるいは、エージェントに埋め込まれる機能の実装形態など、エージェントを構成する諸要素の仕様が与えられていることが必要である。

■7群-7編-1章

1-4 合理的エージェント：論理的

(執筆著：打矢隆弘) [2019年1月 受領]

1-4-1 合理的エージェント

エージェントは、利用者やほかのエージェントと知的に相互作用する自律的ソフトウェアである。自身が有するセンサを用い外界（環境）を知覚・認識し、問題解決機構により意思決定を行い、エフェクタを介して外界への働きかけを行う。

エージェントは、状況認識や行動選択を行う際の基盤となる知識を持ち、それを問題解決の過程で利用することにより、刻々と変化する状況に自律的に対処する。一般的に、刻々と変化する状況に対して素早く反応しなければならないような場合には、実時間処理に適した即応型エージェントを用いる。一方、複雑な問題を解決する場合には、エージェントの心的状態（自己の持つ意図や信念、願望など）を含めた多様な情報・知識を収集・管理して問題解決を行う熟考型エージェントを用いる。

複雑な問題を解決する場合、エージェントが行った一連の行為が最善のものであったかどうかということが非常に重要になる。合理的エージェント¹⁾とは、エージェントが置かれている環境で最良の結果を達成するために行動するエージェントである。

1-4-2 合理性

ある時点での合理性は、成功を定義する性能指標（効用）、環境に関するエージェントの先見の知識、エージェントがとることのできる行為、これまでのエージェントの知覚系列、の4点によって決まると言われている²⁾。

すなわち、各知覚系列についてエージェントの持つ知識から分かる範囲で、合理的エージェントはその性能指標を最大化すると予想される行為を選択する（経済的合理性）。ここで、性能指標はエージェントがどれくらい成功したかを評価する基準である。性能指標はどんなエージェントにも適用できる一般的な基準はなく、エージェントを構築した設計者が基準を定める場合が多い。

また、論理的な整合性という観点から意思決定を行う合理的エージェントも存在する（論理的合理性）。論理的合理性に関しては、Shoham が論理的合理性を持つエージェントのためのエージェント指向プログラミングを提案している³⁾。また、Bratman は合理的エージェントの行動決定の核となる「意図」の理論を提案している⁴⁾。

1-4-3 合理的エージェントの設計

合理的エージェントの設計段階において、エージェントの「学習」を考慮することが望ましい。エージェントの設計者が与える先見の知識は部分的あるいは不正確であることが多く、そのみを用いて合理的な動作を行うエージェントを実現することは難しい。また、外界の環境変化により先見の知識が役に立たなくなる恐れもある。そこで、センサを用いて外界の情報を収集するだけでなく、知覚したことから可能な限り学習し、自身の動作知識を修正したり、新たな知識を獲得する仕組みが必要である。

1-4-4 BDI Logic

BDI Logic はエージェントの信念，目標，意図などの心的状態や行為などを記述することができる時相論理体系の一つであり，論理的合理性を持つ熟考型エージェントの内部アーキテクチャの表現モデルとして活用されている．BDI Logic では，様相オペレータ BEL，GOAL，INTEND が心的状態の信念，目標，意図をそれぞれ表す．また，時相オペレータ X (Nexttime)，F (Sometime)，U (Until)，E (ある未来で)，A (あらゆる未来で) などを有する．Rao と Georgeff が BDI Logic を用いて表現している心的状態の整合性公理，コミットメント戦略⁵⁾を以下に示す．ここで，コミットメント戦略とは，目標達成のための行為実行まで意図を持続させる特性を指す．

(1) 心的状態の整合性公理

$$(A1) \text{ GOAL}(\alpha) \supset \text{BEL}(\alpha)$$

意味：エージェントが α を達成するという目標を持っていれば，そのエージェントは α を達成できると信じている．

$$(A2) \text{ INTEND}(\alpha) \supset \text{GOAL}(\alpha)$$

$$(A3) \text{ INTEND}(\text{does}(e)) \supset \text{does}(e)$$

$$(A4) \text{ INTEND}(\varphi) \supset \text{BEL}(\text{INTEND}(\varphi))$$

$$(A5) \text{ GOAL}(\varphi) \supset \text{BEL}(\text{GOAL}(\varphi))$$

$$(A6) \text{ INTEND}(\varphi) \supset \text{GOAL}(\text{INTEND}(\varphi))$$

$$(A7) \text{ done}(e) \supset \text{BEL}(\text{done}(e))$$

$$(A8) \text{ INTEND}(\varphi) \supset \text{AF}(\neg \text{INTEND}(\varphi))$$

(2) コミットメント戦略

$$(A9A) \text{ INTEND}(\text{AF}\varphi) \supset \text{A}(\text{INTEND}(\text{AF}\varphi) \cup \text{BEL}(\varphi))$$

意味：エージェントが φ をあらゆる未来でいつか達成するという意図を形成していれば，あらゆる未来でエージェントは φ を達成したと信じるまでその意図を持続する．

$$(A9B) \text{ INTEND}(\text{AF}\varphi) \supset \text{A}(\text{INTEND}(\text{AF}\varphi) \cup (\text{BEL}(\varphi) \vee \neg \text{BEL}(\text{EF}\varphi)))$$

意味：エージェントが φ をあらゆる未来でいつか達成するという意図を形成していれば，あらゆる未来でエージェントは φ を達成した，または， φ は達成不可能だと信じるまでその意図を持続する．

$$(A9C) \text{ INTEND}(\text{AF}\varphi) \supset \text{A}(\text{INTEND}(\text{AF}\varphi) \cup (\text{BEL}(\varphi) \vee \neg \text{GOAL}(\text{EF}\varphi)))$$

意味：エージェントが φ をあらゆる未来でいつか達成するという意図を形成していれば，あらゆる未来でエージェントは φ を達成したと信じるか，または， φ を目標から撤回するまでその意図を持続する．

BDI Logic を用いることで，合理的エージェントの心的状態の整合性やコミットメント戦略などを記述することが可能である．

1-4-5 合理的エージェントが備えるべき要件

文献 6) では論理的合理性を持つ合理的エージェントが備えるべき要件を下記のように定義している．

[要件 1] 整合性と目的—手段について一貫性を持った計画を信念の一部として保持できる．

[要件 2] 目的を達成するには，どのような意図を形成すれば合理的か熟考できる．

- [要件 3] 意図の目的を達成するための合理的な手段を目的一手段推論で推論できる。
- [要件 4] 目的を達成するために、実践的推論を用いて、信念と整合的で目的一手段について一貫性を持った意図を形成できる。
- [要件 5] 意図から、現状に最も合理的な現在指向的意図を熟考できる。
- [要件 6] 意欲的コミットメントを機能として実現し、未来志向的意図を実行に移すように、エージェントの振る舞いをコントロールできる。
- [要件 7] 推論の核となるコミットメントを機能として実現し、実践的推論を起動して計画や意図の部分の埋めていく。
- [要件 8] コミットメント戦略を機能として実現し、意図の持続、破棄ができる。

1-4-6 合理的エージェントの実現例 1：ルールベースエージェント

エージェントフレームワークの一つである DASH (Distributed Agent System based on Hybrid architecture) ⁸⁾ 上で動作する DASH エージェントは、if-then 型のルールに基づき行動するルールベース型エージェントであり、エージェント内部に、推論機構と呼ばれる、自身の行動を決定する仕組みを有している (図 4・1)。推論機構は、推論エンジン、ワーキングメモリ (WM)、ルール集合から成る。WM には、ファクトと呼ばれるエージェントが得た各種情報が格納されている。ファクトは OAV 型と呼ばれるデータ形式で表現されている。OAV 型とは、オブジェクト (Object) と、その属性 (Attribute)・属性値 (Value) の組で表されるデータ形式である。また、ルール集合は開発者が記述した if-then で表されるルール型の動作知識から成る。推論エンジンが WM とルール集合から実行可能なルールを選択・実行することにより DASH エージェントは動作を行う。このルールベースの動作制御の仕組みにより、DASH エージェントは環境の変化に追従した柔軟な振る舞いの実現を可能とする。

DASH エージェントは次のように前向き推論動作を行う。

1. 動作知識の制御を行う推論エンジンが WM を参照し、その内容にマッチするルールを探索する。
2. ファーストマッチしたルールを実行する。
3. ルールを実行した結果 WM の内容が更新された場合、1. へ戻る。

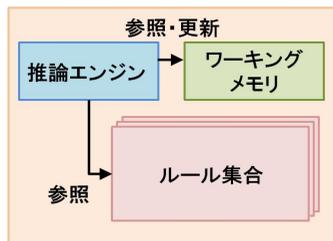


図 4・1 推論機構

DASH エージェントは、多様な情報・知識を収集・管理して問題解決を行う熟考型エージェントとして動作させることが可能である。また、推論履歴を用いて、問題解決の手順が適切であったかどうかを評価することも可能である。

1-4-7 合理的エージェントの実現例 2 : Q 学習エージェント

(1) Q 学習

Q 学習^{8),9)}は、強化学習分野における代表的な手法の一つとして知られている。強化学習とは、エージェントのとった行動の結果に対して評価を与え、エージェントがその評価を最大化するよう、試行錯誤的に学習を行う手法である。Q 学習はマルコフ決定過程 (MDP) において十分な回数の試行を繰り返した場合、最適解への収束が証明されている。Q 学習は環境にエージェントが1体のみ存在する状況に対し、特に有効に適応する。

(2) 学習のプロセス

Q 学習では、エージェントが持つ各ルールの優先度を、行動の結果を基に更新していくことで学習を行う。

1. エージェントは現在の状態 (環境) を観測し、それにマッチするルールを探索する。
2. マッチしたルールが複数存在した場合、エージェントの持つ行動選択手法に基づき、そのなかから1つを選択する。
3. で選択したルールを実行する。
4. 実行結果から更新式に従いルール優先度を更新する。
5. エージェントの動作の終了判定を行う。判定方法はエージェントにより異なるが、更にエージェントの動作が続く場合は、1. に戻る。

(3) 行動選択手法

強化学習において、学習のみを目的とする場合、本来は行動選択手法を考える必要はない。ルールを実行することで試行錯誤的に学習が進むため、全ルールが十分な回数実行されれば、学習は完了する。よって、ランダムにルールを選択しているだけでよい。しかし、実際には、学習とシステム運用が同時に求められる場合がほとんどである。そのような場合には、なるべくすべてのルールを実行しつつ、優先度の高いルールを実行するよう、行動選択手法を考える必要がある。代表的な行動選択手法として、 ϵ -greedy 法、ソフトマックス法などが知られている。 ϵ -greedy 法は確率 $1-\epsilon$ で優先度が最大のルールを選択し、確率 ϵ でランダムにルールを選択する手法である。ソフトマックス法はルールの優先度の比によって各ルールの選択確率を計算し、その確率に基づいて行動選択を行う手法である。

(4) ルール優先度の更新式

Q 学習では、以下に示す更新式を用いてルール優先度の更新を行っている。

$$AQ(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left[r + \gamma \max_{a' \in A(s')} Q'(s', a') \right]$$

更新式において、 $Q(s, a)$ は状態 s において行動 a をとるというルールの優先度、 $Q'(s', a')$ は遷移先の状態 s' において行動 a' をとるというルールの優先度を表している。また、 r は状態 s から s' への遷移において得られる報酬、 $A(s')$ は状態 s' で実行可能な行動全体の集合を表し、 $\alpha (0 < \alpha < 1)$ は学習率、 $\gamma (0 < \gamma < 1)$ は割引率を表している。優先度の更新はエージェントが行動を実行し、次の状態に遷移するたびに行われるが、エージェントは遷移によって得られた報酬 r と、遷移先で実行可能なルールのなかで優先度が最大であるルールの値 ($\max_{a' \in A(s')} Q'(s', a')$) を基にして、ルールの優先度の更新を行っている。

(5) 学習率と割引率

学習率とは、ルールの優先度の更新を行う際に、いままでの優先度の値を優先するか、得られた結果（ここでは、得られた報酬・遷移先で実行可能なルールの情報）を優先するかのバランスを表したパラメータである。更新式より、 α が 0 に近づくほどいままでの優先度の値を重視し、逆に 1 に近づくほど得られた結果を重視して優先度の更新を行うことが分かる。なお、学習率は 0.1 程度の値を設定することが一般的である。

割引率とは、将来獲得予定の報酬($\max_{a' \in A(s')} Q'(s', a')$)を現時点でどれだけ重要視するかの度合を表したものである。更新式では、状態 s から s' への遷移によって報酬 r が得られたが、まだ状態 s' では行動していないため、その先の行動から良い結果が得られるという完全な保証はない。よって、状態 s' で実行可能なルールの優先度は考慮に入れる必要はあるが、ある程度割り引いて考える必要がある。その度合を表したものが割引率である。更新式より、 γ が 0 に近づくほど将来獲得予定の報酬を軽視し、逆に 1 に近づくほど重視することが分かる。なお、割引率は 0.9~0.99 程度の値を設定することが一般的である。

(6) DASH エージェントへの Q 学習の導入

DASH エージェントに組み込まれている推論機構と連携して運用する自動学習機構を導入する。この自動学習機構は、行動選択エンジンと学習エンジンによって構成される。

(a) 行動選択エンジン

行動選択を行う。行動選択手法として ϵ -greedy 法とソフトマックス法が実装されており、ユーザがエージェント起動時にどちらの手法を使用するかを決定する。

(b) 学習エンジン

行動選択エンジンが選択したルールを実行した結果から、更新式を用いて実行したルールの優先度を更新する。

(7) 動作の流れ

動作は、行動選択・実行フェーズ（図 4・2）、優先度更新フェーズ（図 4・3）の 2 段階に分かれる。

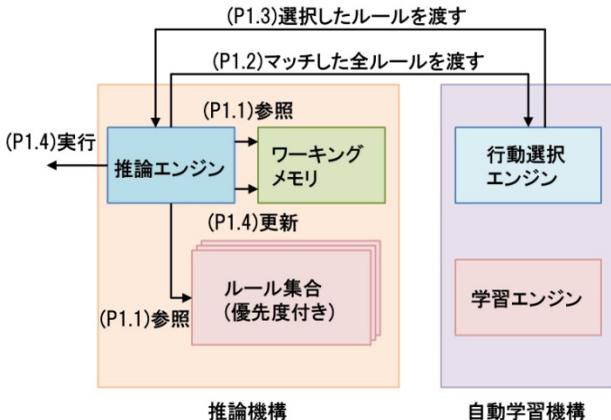


図 4・2 行動選択・実行フェーズ

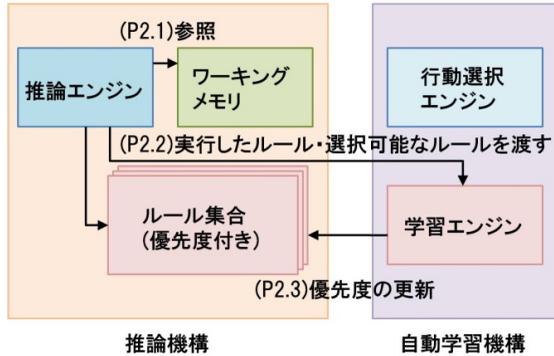


図 4・3 優先度更新フェーズ

—行動選択・実行フェーズ—

- (P1.1) ワーキングメモリ (WM) の内容を参照
- (P1.2) 選択可能なルールを行動選択エンジンへ送る
- (P1.3) 実行するルールを選択
- (P1.4) ルールを実行

—優先度更新フェーズ—

- (P2.1) WM の内容を参照
- (P2.2) 実行したルール, 選択可能なルールを学習エンジンへ送る
- (P2.3) 実行したルールの優先度を更新
- (P2.4) 終了判定

1-4-8 合理的エージェントの実現例 3 : Nash-Q 学習エージェント

(1) Nash-Q 学習

環境にエージェントが複数存在する場合, ほかのエージェントと干渉しない行動を強化することが理想的な学習と言える。Nash-Q 学習¹⁰⁾は, マルチエージェント環境においても, 他エージェントとのナッシュ均衡を学習することで最適性を持たせるために考案された学習方法である。相手のエージェントの行動を蓄積していき, 現状態からのナッシュ均衡をこれまでの履歴から予想し, 相手の行動の優先順位を予測することで, マルチエージェント環境を MDP に近づけ, 学習精度を向上する。また, 各エージェントがお互いの行動に干渉しない行動を学習することができる。Nash-Q 学習は環境にエージェントが二体存在する状況に対し, 特に有効に適応する。

(2) DASH エージェントへの Q 学習の導入

DASH エージェントの自動学習機構を, Nash-Q 学習を利用できるように拡張する(図 4・4)。Nash-Q 学習の導入により, 相手の行動を阻害しない行動を学習する。エージェントは自身が行動したあとに, 得られた報酬と相手行動を参照し選択したルールの優先度を更新する。

ルール優先度データベース: テーブルを 1 つ有する。状態ごとに自分のルールとルールの優先度を管理する。

相手行動履歴データベース: 2 つのテーブルを有する。1 つ目は状態ごとに自分のルール&

相手のルール，優先度を管理する。2 つ目は状態毎に相手のルールとルールの予想優先度を管理する。

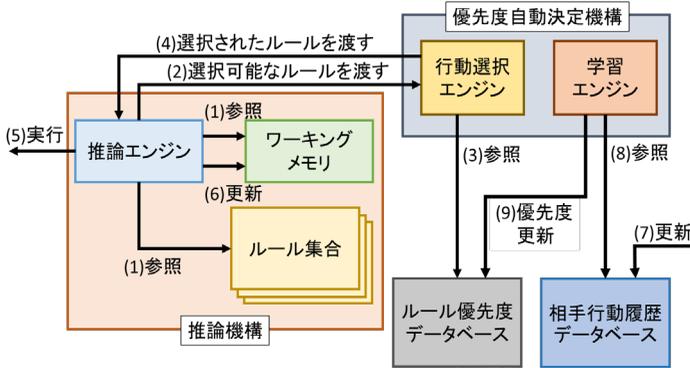


図 4・4 Nash-Q 学習用エージェントアーキテクチャの構成

【動作フロー】

自動学習機能を用いてエージェントが実行ルールを選択し優先度を更新する流れについて説明する (図 4・5)。ここで，自身のエージェントをエージェント A，相手エージェントをエージェント B とする。a は行動， α は学習率， γ は割引率である。

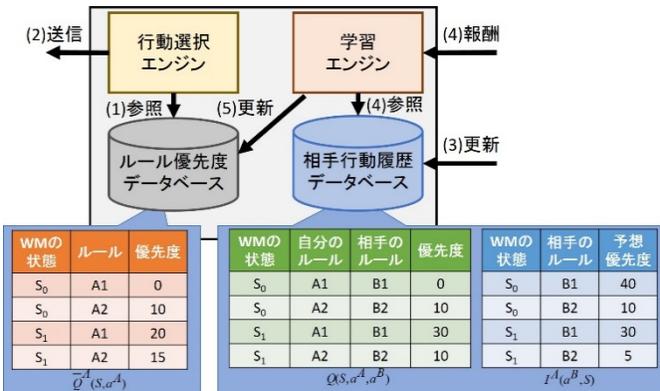


図 4・5 Nash-Q 学習の動作の流れ

1. 実行ルールの選択

エージェント A がエージェント B の政策を $I^A(a^B|S)$ とし，エージェント A が状態 S で行動 a^A をとる優先度示す関数 $\bar{Q}^A(S, a^A)$ を，

$$\bar{Q}^A(S, a^A) \equiv \sum_{a^B \in A^B} I^A(a^B|S) Q^A(S, a^A, a^B)$$

とする。 \mathcal{A} は選択可能なルールの集合である。 現在状態 $s_t \in S$ において、 エージェント A は政策 (ε -greedy など) に従い行動を確率的に選択する。

2. 選択したルールを推論エンジンに送信

エージェント A は、手続き 1 で選択した行動を実行する。 ここで、ほかのエージェントも同時に行動を選択し実行する。 両エージェントの行動により、状態は現状態 s_t から次状態 s_{t+1} に移行し、 エージェント A は環境から報酬 r_{t+1}^A を受け取る。

3. 相手の行動履歴から相手行動履歴データベースを更新

エージェント A は、状態 s_t 、行動 a_t^A 、 a_t^B に対する 行動価値関数である $Q(s_t, a_t^A, a_t^B)$ を下記式に従い更新する。

$$Q(s_t, a_t^A, a_t^B) \leftarrow (1 - \alpha)Q(s_t, a_t^A, a_t^B) + \alpha[r_{t+1}^A + \gamma \max_{a \in \mathcal{A}^A} \bar{Q}(s_{t+1}, a, a_{t+1}^B)]$$

エージェント A は状態 s_t においてエージェント B が選択可能なすべての行動 $a^B \in \mathcal{A}^B$ に対して、下記式に従い関数 I^A を更新する。

$$I^A(a^B, s_t) \leftarrow (1 - \theta)I(a^B | s_t) + \begin{cases} \theta & (a^B = a_t^B) \\ 0 & (\text{otherwise}) \end{cases}$$

ここで、 θ は観測した行動を将来の行動予測時にどれくらい考慮するかを決定するパラメータである。

4. ルール優先度の計算

環境から得られた報酬と相手行動履歴データベース、下記式で、実行したルールの優先度を計算する。

$$\bar{Q}(s, a^A) \equiv \sum_{a^B \in \mathcal{A}^B} I^A(a^B | S) Q^A(S, a^A, a^B)$$

5. 優先度の更新

ルール優先度データベースの優先度を更新する。更新後 1 へ戻る。

■参考文献

- 1) M. Wooldridge : “Reasoning about Rational Agents,” The MIT Press, 2000.
- 2) S. Russell, P. Norvig (著), 古川康一(監訳): “エージェントアプローチ 人工知能(第2版),” 共立出版, 2008.
- 3) Y. Shoham : “AGENT0: A simple agent language and its interpreter,” in Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), 1991.
- 4) M.E. Bratman : “Intention, Plans, and Practical Reason,” Harvard University Press, 1987.
- 5) A.S. Rao and M.P. Georgeff : “Modeling Rational Agents within a BDI-Architecture,” in Proc. of International Conference on Principles of Knowledge Representation and Reasoning, pp.473-484, 1991.
- 6) 高田司郎, 新出尚之: “意図に基づくエージェントアーキテクチャ,” 人工知能学会誌, vol.20, no.4, pp.433-440, 2005.
- 7) T. Uchiya, H. Hara, K. Sugawara, T. Kinoshita : “Repository-Based Multiagent Framework for Developing Agent Systems,” IGI Global, Ch.4, pp.60-79, 2011.
- 8) 謝 孟春, 立花 敦: “マルチエージェントの協調行動の取得における Q 学習に関する考察,” JAWS2007 合同エージェントワークショップ&シンポジウム論文集, pp.441-448, 2007.
- 9) C.J.C.H. Watkins and P. Dayan : “Technical note: Q-learning,” Machine Learning, vol.8, pp.55-68, 1992.
- 10) J. Hu: “Nash Q-Learning for General-Sum Stochastic Games,” Journal of Machine Learning Research 4, pp.1039-1069, 2003.

■7 群-7 編-1 章

1-5 合理的エージェント：経済的

■7群-7編-1章

1-6 社会的エージェント

■7群-7編-1章

1-7 マルチエージェント

(執筆著者：北村泰彦) [2011年3月 受領]

Wooldridge¹⁾によると、「マルチエージェントシステム (Multiagent System) は複数のエージェント (Agent) と呼ばれる計算要素から構成される。エージェントは、ある目的を達成するために自らの行動を決定する自律性を持ち、単なるデータ交換だけでなく、協調 (Cooperation)、調整 (Coordination)、交渉 (Negotiation) といった社会的行動によってほかのエージェントと相互作用する。」と定義されている。

マルチエージェントの研究は1980年頃から米国のAI研究者を中心に開始され、当初は分散人工知能 (Distributed Artificial Intelligence : DAI) や分散問題解決 (Distributed Problem Solving : DPS)²⁾ と呼ばれていた。当時からマルチエージェントという表現はみられる³⁾が、エージェントは単なるシステムを構成する要素という認識しかなく、人工知能との強い関連性を示すためにも分散人工知能という呼称が用いられていたように思える。

当時の研究アクティビティの中心となっていたのは、1980年から開始されたDAIWorkshopである。第1回の会議で示された定義は、「分散人工知能は単一の問題解決機、単一の計算機、単一の計算点では不適切とみられるような問題を対象とする。」という非常に広義なものとなっており、並列AIやコネクショニストモデルといった細粒度の要素からなる分散システムもその研究対象となっていた。「DAIはAIの発展に必要不可欠である。」⁴⁾や「DAIとAIの違いは本質的なものではなく、強調点の違いにある。」⁵⁾といった主張がなされるなど、AIとのつながりが強調されていた。その一方で、分散処理との違いを主張している。(当時の)分散処理システムではデータの分散を対象としているのに対し、DAIはデータだけでなく制御の分散も対象とする。また、分散処理システムは同時実行制御のように個別のタスクを矛盾なく処理することを対象とするのに対して、DAIは一つの問題を複数のノードが協力して解決することを対象とする。

当時、DAIの応用領域とみなされていた分野は、マルチロボット、センサネットワーク、工場スケジューリング、航空管制、オフィスシステムなどであり、複数のエージェントが如何に協力して、共通の問題を解決するかという分散問題解決が主な研究課題であった。DAI黎明期の研究成果に関してはChandrasekanがエディタを務めたIEEE SMCの分散問題解決特集号にまとめられている²⁾。

DAIWorkshopはほぼ毎年開催されてきたが、研究の範囲が広すぎて、だんだんと議論がかみ合わなくなってきた。また、参加者が常に入れ替わり、研究の手法やツールに関する連続性がなくなっていることも指摘された⁶⁾。そこで、DAIのアイデンティティを確立すべく、研究会の成果をまとめた2冊の会議録⁷⁾⁸⁾と、DAIに関する論文集⁹⁾が相次いで出版された。そのなか⁹⁾のBondとGasserによるサーベイでは、分散人工知能を分散問題解決、マルチエージェントシステム (Multiagent System : MAS)、並列人工知能 (Parallel Artificial Intelligence : PAI) の三分野に分類した。DPSは「ある問題を分割して、複数のモジュールに割り当て、協調的に解決する方法」を議論し、MASは「(あらかじめ存在する) 自律的な知的エージェントの集まりにおける振る舞いの調整」を議論する。PAIは「AIのための並列コンピュータアーキテクチャ、言語、アルゴリズムの開発」を議論するが、サーベイのなかで議論の対象としないと述べ

ている。すなわちコネクショニストモデルなどの細粒度のシステムは DAI の研究対象から外された。

Bond と Gasser は DPS と MAS を区別したが、その大きな違いはエージェントの博愛性 (Benevolence) を前提とするかどうかにある。従来の DPS では一つの問題を解決するために、すべてのエージェントが協力するという博愛性がその前提であった。その一方で、MAS ではその前提を取り去り、エージェント同志が敵対するような場合もその議論の対象とする。この発端となったのは Rosenschein らによる Deals Among Rational Agents の研究である¹⁰⁾。ここではエージェントの行動基準が効用として明示的に表現され、自らが利得を得られる行動を選択する。エージェント同志が協調するかどうかはゲーム理論をベースとして議論されるようになった。従来の DPS は人工知能、計算機科学、分散処理といった分野の枠組みを用いて研究が進められていたのに対して、MAS の研究では、投票やオークションなどの意思決定理論、市場計算などの経済理論、組織論といった社会科学の枠組みが盛んに導入されるようになった。

DAI の研究は、欧州では MAAMAW (European Workshop on Modelling Autonomous Agents in a Multi-Agent World)、日本では MACC (Multi-Agent and Cooperative Computation) という研究会が行われていたが、それらのアクティビティを統合する形で、1995年には国際会議 ICMAS (International Conference on Multi-Agent Systems) が始まった。更に、2002年には自律エージェント系の会議である AA (International Conference on Autonomous Agents)、ATAL (International Workshop on Agent Theories, Architectures, and Languages) とともに統合され、AAMAS (International Joint Conference on Autonomous Agents and Multiagent Systems) となった。現在では分散人工知能や分散問題解決よりも、マルチエージェントシステムという用語が一般的に用いられるようになった。逆に、Bond と Gasser が示したほどの狭義の意味づけではなくなっている¹¹⁾。2000年代になると Ferber¹²⁾、Weiss¹³⁾、Wooldridge¹⁾、Shoham¹⁴⁾などの教科書や Autonomous Agents and Multi-Agent Systems といった論文誌も相次いで出版されるようになった。

■参考文献

- 1) M. Wooldridge: "An Introduction to MultiAgent Systems," Wiley, 2002.
- 2) B. Chandrasekaran: "Natural and social system metaphors for distributed problem solving: Introduction to the issue," IEEE SMC, vol.11, no.1, pp.1-5, 1981..
- 3) K. Konolige and N.J. Nilsson: "Multiple-agent planning systems," Proc. AAAI 80, pp.138-142, 1980.
- 4) R. Davis: "Report on the Workshop on Distributed AI," ACM SIGART Bulletin, no.73, pp.42-52, 1980.
- 5) M. Fehlin and L. Erman: "Report on the Third Workshop on Distributed Artificial Intelligence," ACM SIGART Bulletin, no.84, pp.3-12, 1983.
- 6) N.S. Sridharan: "1986 Workshop on Distributed AI," AI Magazine, Fall, 1987.
- 7) M.N. Huhns: "Distributed Artificial Intelligence," Pitman, London, 1987.
- 8) L. Gasser and M.N. Huhns: "Distributed Artificial Intelligence, Volume II," Pitman, London, 1989.
- 9) A. Bond and L. Gasser: "Readings in Distributed Artificial Intelligence," Morgan Kaufmann, 1988.
- 10) J.S. Rosenschein and M.R. Genesereth: "Deals among rational agents," Proc. IJCAI 85, pp.91-99, 1985.
- 11) N.R. Jennings, K. Sycara, and M. Wooldridge: "A road map of agent research and development," Autonomous Agents and Multi-Agent Systems, vol.1, pp.7-38, 1998.
- 12) J. Ferber: "Multi-Agent Systems," Addison-Wesley, 1999.
- 13) G. Weiss: "Multiagent Systems," MIT Press, 1999.
- 14) Y. Shoham and K. Leyton-Brown: "Multiagent Systems," Cambridge, 2009.

■7 群-7 編-1 章

1-8 ソフトウェアエージェント

(執筆者：服部文夫) [2010 年 10 月 受領]

1-8-1 ソフトウェアエージェントとは

ソフトウェアエージェントとは、人やほかのシステムの代行、手助けをする自律的なソフトウェアである。

そもそもソフトウェアエージェントの定義は人によって様々である。ブラッドショー (Bradshaw) は文献 1) で、ソフトウェアエージェントを Agent as an Ascription と Agent as a Description の 2 つの面から定義している。前者はエージェントを役割から捉えたもので、誰かの代わりにその仕事を遂行するもの、と定義している。その場合でも、エージェントにはユーザの要求や状況を理解する能力が必要となる。一方、後者はエージェントをその作りから捉えたもので、環境を理解して自律的に判断を行うソフトウェアとして定義される。本稿は前者の立場に立っており、関連する概念を図 8・1 のように分類する。

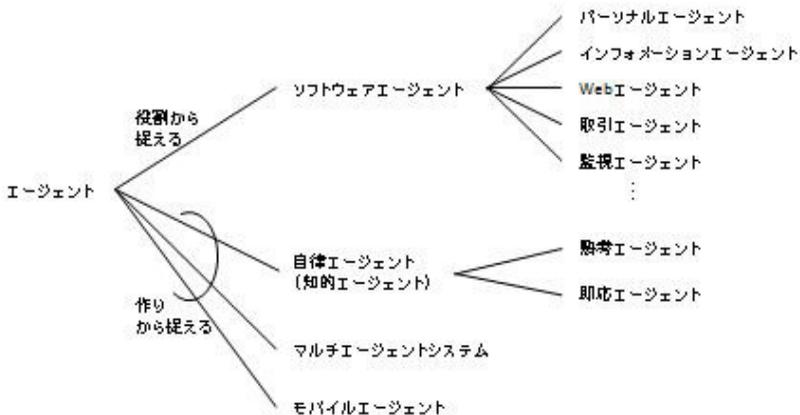


図 8・1 ソフトウェアエージェントの位置づけ

1-8-2 ソフトウェアエージェントの研究

ユーザの仕事を実行するエージェントの概念は、既に 1984 年のアラン・ケイ (Alan Kay) の論文 2) に現れている。また、1987 年に発表された Apple 社のコンセプトマシン Knowledge Navigator³⁾ では画面に秘書としてのエージェントが登場する。1990 年代にはソフトウェアエージェントの研究の全盛期を迎え、数々の論文や本が出版された。なかでも、メーズ (Maes) によるインタフェースエージェント⁴⁾ や、エツィオーニ (Etzioni) らの Softbot⁵⁾ が有名である。日本においても、1997 年に電子情報通信学会主催の“ソフトウェアエージェントとその応用シンポジウム”が開催され、連動した論文特集号が発行された⁶⁾。このシンポジウムはマルチエージェントと協調計算研究会などと統合されて、合同エージェントワークショップ&シンポジウム (JAWS) として継続的に開催され、ソフトウェアエージェントに関する数多くの論文が

発表されている。

しかし、2000年代に入るとソフトウェアエージェントを名乗った研究は急速に姿を消していく。これは汎用的なエージェントの研究から、問題に特化した機能を組み込んだシステムの研究へと移っていったため、ソフトウェアエージェントの概念が必要なくなったというわけではない。

1-8-3 なぜソフトウェアエージェントか

インターネット及び WWW の発展に伴い、人間を取り巻く情報量が急激に増大しただけでなく、多様なサービスを発見し、使いこなすことが困難になりつつある。このような環境で、ユーザのタスクを代行するソフトウェアエージェントの概念がますます重要になってきていることは言うまでもない。更に、ユビキタスコンピューティングの進展にともない、システムや環境がユーザに働きかける主体として、エージェントの概念は必須となっている。

今後ソフトウェアエージェントの概念を取り込んだ研究分野として注目されているものは、以下のようなものがある。

- (1) **情報検索・統合、情報推薦**：ユーザの意図や状況に応じた情報収集、推薦など。
- (2) **Web マイニング**：膨大な Web 上のデータからの有益な情報のマイニング。
- (3) **エレクトロニックマーケット**：代理取引やオークションのためのエージェント。
- (4) **セマンティック Web サービス**：Web 上のサービスをユーザに代わって利用、統合する。
- (5) **ソーシャルサポート**：ネットワークコミュニティや実世界でのユーザの行動の支援。

■参考文献

- 1) J.M. Bradshaw : "An Introduction to Software Agents," in Software Agent, J.M. Bradshaw (ed.), AAAI Press/The MIT Press, Menlo Park/Cambridge, pp.3-46, 1997.
- 2) A. Kay : "Computer Software," Scientific American, vol.21, no.3, pp.53-59, 1984.
- 3) J. Sculley and J. Byrne : "Odyssey: Pepsi to Apple: A Journey of Adventure, Ideas, and the Future," HarperCollins, New York, 1987.
- 4) Pattie Maes : "Agents that Reduce Work and Information Overload," Commun. ACM, vol.37, no.7, pp.30-40, 1994.
- 5) Oren Etzioni and Daniel Weld : "A softbot-based interface to the Internet," Commun. ACM, vol.37, no.7, pp.72-76, 1994.
- 6) 服部文夫 : "ソフトウェアエージェントとその応用論文特集の発行にあたって," 信学論(D-I), vol.J81-D-I, no.5, pp.431-432, 1988.

■7 群-7 編-1 章

1-9 インタフェースエージェント

■7 群-7 編-1 章

1-10 モバイルエージェント