

## ■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)

---

### 2章 エージェントアーキテクチャ

#### 【本章の構成】

本章では以下について解説する.

- 2-1 エージェントアーキテクチャ
- 2-2 熟考アーキテクチャ
- 2-3 階層アーキテクチャ
- 2-4 即応アーキテクチャ
- 2-5 連邦アーキテクチャ
- 2-6 包摂アーキテクチャ

## ■7群-7編-2章

### 2-1 エージェントアーキテクチャ

(執筆著者：木下哲男) [2011年2月 受領]

エージェントという処理体の概念的、構造的、機能的な仕組みやその特徴などを規定したものがエージェントアーキテクチャと呼ばれ、エージェントの設計や実装におけるモデルや指針として参照・利用されたり、実現されたエージェントの類識別などで活用されたりする。

また、エージェントアーキテクチャは、これに準拠して設計・実装されたエージェントの原型を与えるものとなる。そこで、エージェントを定義する際の手段として、既知のエージェントアーキテクチャを利用することもしばしば行われる。例えば、1-2 節では、エージェントの機能や構造に着目した場合、「センサ」、「意思決定機構」、「エフェクタ」という3つの機能要素に基づいてエージェントが定義されるという例を挙げた。この定義では、上記の3つの機能要素によって構成されるエージェントアーキテクチャを暗黙的な前提として、エージェントの振舞いを説明している。

一方、エージェントアーキテクチャは、その対象となるエージェントの目的や用途、備えるべき機能や構造などの要件を考慮して、設計者の視点からトップダウン的に与えられる場合が多い。実際、これまで提案され利用されてきた種々のエージェントの事例では、それらの設計コンセプトや実装条件などの違いが反映された固有のアーキテクチャとなっており、有用なエージェントアーキテクチャやそれに関する知見を広く活用してゆくのは困難であった。そこで、これらのエージェントアーキテクチャに共通的に見られる機能や構造に着目して抽象化を施したり、アーキテクチャが立脚している基本概念に注目して類別したりすることにより、エージェントアーキテクチャの類型化が行われる。

例えば、前述した「センサ」、「意思決定機構」、「エフェクタ」に基づくエージェントアーキテクチャは、協調プロトコル「契約ネット」における処理ノード、黑板モデルにおける知識源、メッセージ通信により協調するソフトウェアエージェント、或は、身体を備えたロボットなどの事例で導入されたエージェントの機能や構造を抽象化して得られる汎用的アーキテクチャの一例である。

以下、2章の各節では、代表的なエージェントアーキテクチャとして知られている5つの類型、すなわち、「熟考アーキテクチャ」、「階層アーキテクチャ」、「即応アーキテクチャ」、「連邦アーキテクチャ」、「包摂アーキテクチャ」を取り上げて順次説明する。

## ■7群-7編-2章

---

### 2-2 熟考アーキテクチャ

## ■7 群-7 編-2 章

---

### 2-3 階層アーキテクチャ

## ■7 群-7 編-2 章

---

### 2-4 即応アーキテクチャ

## ■7群-7編-2章

### 2-5 連邦アーキテクチャ

(執筆者：峯 恒憲) [2010年10月受領]

#### 2-5-1 連邦アーキテクチャ

連邦化 (federation) は、分散システムを構築する際の基盤技術の一つである。連邦化技術を利用したシステムアーキテクチャ (以下、連邦アーキテクチャ) では、エージェントは、ファシリテータ (facilitator) と呼ばれる特別なエージェントに接続し、一つの仮想化されたコミュニティを形成する。またファシリテータは、ほかのファシリテータと接続し、相互通信を行うことができる。連邦アーキテクチャは、黒板モデルを利用したアーキテクチャと異なり、一般に共有メモリを持たない<sup>5)</sup>。

連邦アーキテクチャでは、エージェントはファシリテータに動的に接続したり離れたりする。ファシリテータに接続すると、エージェントは自身のサービス機能 (capabilities) や要求 (needs) の仕様 (specification) を ACL (Agent Communication Language) [7編4章4-1参照] と呼ばれる標準化された共通言語 (例えば、FIPA ACL や KQML<sup>2)</sup>) を利用して提供する。ACL に記述されたメタレベルの情報に加え、エージェントはまた、アプリケーションに関する応用レベルの情報をファシリテータに送信 (もしくはファシリテータから受信) する。送信の際には、ファシリテータが解釈できるよう、必要な説明書を付けて送る。ファシリテータは、このようにして提供されたエージェントの仕様や応用レベルの情報を利用して、要求に合うエージェントを発見し、その発見したエージェントが解釈できる表現に要求を翻訳して送る。要求を受信したエージェントは、その要求に合うサービスを提供する<sup>4)</sup>。

連邦アーキテクチャでは、エージェントは、ファシリテータを介して要求を出すことができる代わりに、自身が受けた対応可能な要求に対しては、自身の持つサービスを提供しなければならない。この点で、エージェントは、自身の自律性を犠牲にし、システムの制約に従うことが求められる<sup>4)</sup>。

一般にエージェントは、ほかのエージェントと互いに直接通信を行い、自分たちの活動を自由に調整することもできる。このような直接通信を行うモデルとして、契約ネットプロトコル (contract-net protocol [7編3章3-10節参照]) や、互いに自身の自身の機能 (capabilities) や要求 (needs) を提供し合う仕様共有法 (specification sharing approach) がある。しかしながら、いずれの方法でも、調整相手の発見と、発見した相手との調整の手間を、エージェント自らが負わなければならない<sup>3)</sup>。

これに対して連邦アーキテクチャでは、相互調整の役目をファシリテータに任せることで、エージェントに、調整の重荷をかけさせることなく、エージェントたちの活動に必要なサービスを提供することができる。更に、エージェントたちに互いに名を明かさせることなく匿名で、エージェントの相互運用を実現することは、連邦アーキテクチャの重要な特徴の一つである<sup>4)</sup>。

#### 2-5-2 ファシリテータ

連邦アーキテクチャで利用されるファシリテータは、情報システムにおけるメディアエータ<sup>1)</sup>の一般化された概念に由来している<sup>3)</sup>。ファシリテータは、ファシリテータに接続したエー

エージェントの情報や、エージェントによって通信された情報を保持し、エージェント間の調整や、相互運用を支援するための一連のサービスを提供する。

ファシリテータが提供するサービスには、ホワイトページ(エージェントを名前で見つける)、イエローページ(ある仕事を遂行できるエージェントを見つける)、直接通信(特定のエージェントにメッセージを送る)、内容に基づいた配送制御(エージェントによって提供された仕様やほかの情報を使って、あるエージェントから出された要求に合うサービスを提供するエージェント探し、提供する。)、問題分割(分割統合法と同様に、要求が複雑な場合には、それを分割し、分割された各々の解答を得たのち、統合し、元の要求の解答として返す)、翻訳(異なる語彙を利用する異種のエージェント間での相互運用のため、各エージェント間の語彙の翻訳を行う)などがある<sup>4)</sup>。

#### ■参考文献

- 1) G. Wiederhold : "Mediators in the architecture of future information systems," IEEE Computer, pp.38-49, 1992.
- 2) T. Finin, R. Fritzson, D. McKay, and R. McEntire : "KQML as an agent communication language," Proceedings of the third international conference on Information and knowledge management, pp.456-463, 1994.
- 3) M.R. Genesereth and S.P. Ketchpel : "Software agents," Communications of the ACM, vol.37, no.7, pp.47-54, 1994.
- 4) N. Singh, M. Genesereth, and M. Syed : "A distributed and anonymous knowledge sharing approach to software interoperation," International Journal of Cooperative Information Systems, pp.125-139, 1994.
- 5) S.E. Lander : "Issues in Multiagent Design Systems," IEEE Expert, pp.18-26, 1997.

## ■7群-7編-2章

### 2-6 包摂アーキテクチャ

(執筆著者：浅田 稔) [2010年11月受領]

ロボットなどのエージェントが環境内で活動するための基本制御構造としてのアーキテクチャとして、7-2節の熟考アーキテクチャとの対比で、MITのRodney Brooksが提案し、昆虫ロボットなどに実装したことで、有名になったのが、包摂アーキテクチャである<sup>1)</sup>。包摂は“subsumption”の邦訳で、階層構造において、上位層が下位層を“subsume”（包含する）の意味から来ている。状況オートマトンの流れの上であり、反応的な行動制御器として設計されたものとみなせる。以下では、歴史的経緯、アーキテクチャとしての特徴、その後の進展について概説する。

#### 2-6-1 行動制御規範の流れ

ロボットの行動制御をどのような形で実現するかには、任意性があるが、目的にしたがった制御構造を設計することは当然ながら、その頑強性、汎用性などが求められる。最も単純なロボットの制御構造は、感覚から運動への直結写像である。この手の代表はV. Braitenbergのピークルで、センサからモータへの単純な結線のみでも、光から逃れるなどの見かけ上複雑な行動を生成可能である（例えば、文献2）第2章参照）。ただし、直結の結線構造だけでは、多様な行動を生成することが困難で、この写像部分をどのように設計するかが課題である。人間の意識的な意思決定過程を模擬した構造が7-2節の熟考アーキテクチャである。多様なセンサ群からの情報を処理することで、環境のモデルを構築し、それに基づいて推論、計画、動作生成、モータ制御の一連の系列的な処理を実現する。完全な認識による完全なモデルを前提とすることから、モデル規範型アプローチ（model-based approach）などの呼称がある。当然のことながら、完全な認識や完全な環境モデルはありえず、フレーム問題（文献3）など参照）に代表されるように、想定されない環境変動には弱い。また、系列処理により初期のエラーが後続のモジュールに伝播することで、頑強性にも欠ける。

これに対し、生物の行動規範、特に下位（無意識的、自動的なレベル）の行動の並列処理機構にならって、この問題を解決しようとするアプローチが台頭し、行動規範型アプローチ（behavior-based approach）と呼ばれている。“behavior”という言葉に表されているように、センサからモータへの処理の流れを各行動単位で構築し、これらの単位を並列に処理することで、直列型の弊害を回避するものである。ただし、どのようにモータ出力を統合するかによっていくつかのバリエーションがある。ある種の重み付けで総和をとるもの、各単位への評価（環境の状況への適合度など）により、勝者を選びもの、そして、事前に定められた行動単位の優先度により階層構造を構成し、上位が下位を抑制する構造が包摂アーキテクチャである。更には、それぞれの行動単位をノードとし、行動単位の遷移を有限状態オートマトンとして表すことも可能である（例えば、文献2）第2章参照）。

行動規範型は、下位の並列性により頑強に駆動するが、高次の推論や計画がもともと意図されていないので、結局、モデル規範型との妥協案がハイブリッド型などと称されている。一見、両方の良いところをまとめたアプローチであるが、それと同時に両方の欠点も引き継いでいることになる。



## 2-6-2 包摂アーキテクチャの特徴

包摂アーキテクチャが前提とする条件は、

1. 事前にセンサからモータへの行動制御が規定された行動単位が用意されていること、
2. 各行動単位の優先度が明確に規定されていること、

であり、これらが満足されていれば、頑強に動作する。生物にとっては、捕食者から逃れることが最優先と考えられるが、ロボットの場合は、障害物回避などに相当する。その上で、うろついたり、目標物をさがしたりなどの行動が実現可能なので、優先度が高い行動が階層構造の下位に位置し、この行動が保証されている条件のもとで優先度が低い行動が上位に位置して、動作する構造である。

タスクや環境が複雑になると、階層構造の各層は、行動単位レベルから、より複雑化した行動モジュールからなり、行動モジュールの内部はより下位の行動単位のネットワークなどから構成されることが多い。更に、先に挙げた前提条件を緩和する研究が多くなされて来た。強化学習に代表される機械学習法により、各行動単位を獲得したり、行動の前提となるセンサ情報の選択、更には、行動間の調停機構も学習により獲得する例などがある（ロボカップサッカーにおける例は文献4）に詳しい）。

## 2-6-3 その後の進展

行動規範型はもともと「表象無き知能<sup>5)</sup>」を標榜していたので、モデル規範型とのハイブリッドは意図するところではなく、シンボルがどのようにして、環境との相互作用から創発するかなどの基本問題がクローズアップされ、身体を持つ意味構造が議論されるようになってきた（文献6）など。近年のイメージング技術などの進展により、大きな発展を遂げている脳科学などの分野の知見を組み込みながら、新たなアーキテクチャを模索するもの（例えば、理研の谷らのアプローチ<sup>7)</sup>）、更には、人間の認知発達過程を構成的に明らかにすることで、新たな人間理解を目指す方向のアプローチ（認知発達ロボティクス<sup>8)</sup>）などに進展してきた。

### ■参考文献

- 1) R.A. Brooks : “A robust layered control system for a mobile robot,” IEEE J. Robotics and Automation, vol.RA-2, pp.14-23, 1986.
- 2) 浅田 稔, 國吉康夫 : “ロボットインテリジェンス,” 岩波書店, 2006.
- 3) 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.
- 4) 浅田 稔(編著) : “RoboCupSoccer: ロボットの行動学習・発達・進化,” 共立出版, 2002.
- 5) R. Brooks : “Intelligence without representation,” Artificial Intelligence, vol.47, pp.139-159, 1991.
- 6) R. Pfeifer and J. Bongard (著), 細田 耕, 石黒章夫(翻訳) : “知能の原理—身体性に基づく構成的アプローチ—,” 共立出版, 2010.
- 7) Y. Yamashita and J. Tani : “Emergence of functional hi-erarchy in a multiple timescale neural network model: a humanoid robot experiment,” PLoS Comput. Biol., vol.4, no.11.
- 8) M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida : “Cognitive developmental robotics: a survey,” IEEE Transactions on Autonomous Mental Development, vol.1, no.1, pp.12-34, 2009.