

# ネットワーク上を移動しながら知恵を授かるモバイルエージェント

## The mobile agent gifted with wisdom while migrating in the network

川上 渉\*<sup>1</sup>      梅田 勇一\*<sup>1</sup>      澤村 一\*<sup>2</sup>  
Wataru Kawakami      Yuichi Umeda      Hajime Sawamura

\*<sup>1</sup>新潟大学大学院自然科学研究科

Graduate School of Science and Technology, Niigata University

\*<sup>2</sup>新潟大学工学部情報工学科

Dept. of Information Engineering, Faculty of Engineering, Niigata University

This paper describes an approach to growing agents that migrates over the computer network and improve the quality of their logical arguments by communicating with and incorporating arguments from other agents. For the argument improvement we introduce three argument transformation rules: the rule and fact replacement rules for information refinement and diversification, and the weak literal replacement rule for information completion.

### 1. はじめに

本稿では、我々がエージェントに必要なだと考える能力、すなわち議論とモビリティについて述べる。近年では、人間科学や社会科学と同様にコンピュータサイエンスや人工知能の分野においても議論に注目されている。議論はエージェント指向コンピューティングと関係し、エージェントの相互作用、すなわち、交渉、協調、調停、合意形成、衝突解決などによるエージェント間の対話や決定のための優れたアプローチをもたらす。このことから議論は、エージェント指向コンピューティングにおける社会的計算の枠組とみることができ [5]。

一方モビリティは、コンピュータシステム間に相互作用を与えるプログラムとして研究されてきた。これまでにモバイルエージェントの利点は、技術的観察や考察によって明らかになってきた。しかしながら我々は、モビリティの技術的な面というよりはむしろモビリティやモバイルエージェントの本質的な側面に注目する。

本稿では、エージェント間議論とモビリティ能力を用いて、コンピュータネットワークで成長していくエージェントについて言及する。ここで言う成長とは、学習、進化計算、遺伝的計算などと同様の概念を意味する。これらの目的は、周囲の環境に応じて変化する個体を実現することである。本稿でいうエージェントの成長とは、他のエージェントから知識を得ることで、エージェントが不確実な要素を含む自分の論証 (argument) を反論に強いものにし、より説得力のあるものに洗練することを意味する。我々は、人間の成長と同様にエージェントの成長においても、議論とモビリティは最も基礎的なものであると確信している。オープンでかつ日々変化するネットワーク社会において、議論は、対話を目的に真実を探る方法であり、モビリティは、まだ見ぬエージェントに遭遇し、その知識や考えに触れる方法である。

### 2. 議論フレームワーク

これまでに、多くの研究者達が、各自の特徴を用いて様々な議論モデルを提案してきた [1]。同様に我々も独自の議論モデルを提案してきた [4][5]。

エージェント指向コンピューティングにおける議論の目的とし

連絡先: 川上 渉: 〒 950-2181 新潟県新潟市五十嵐二の町 8050 番地  
新潟大学大学院自然科学研究科情報理工棟 502 Tel: (025)262-7490 e-mail: kawakami@cs.ie.niigata-u.ac.jp

て、我々は、主に以下の三つを取り上げる。

- (1) 各エージェントの挙げる論証の衝突を回避し、その中から納得のいく論証を選びだすこと。
- (2) よりよい論証、または和解案を生成することによって合意を形成すること。
- (3) 不確実な要素を含む論証を反論に強いものとし、より説得力のあるものに洗練すること。

一つ目は、議論の分野におけるほとんど全ての研究の主要目的であり [1]、二つ目は、我々のこれまでの研究の主要目的である [4][5]。そして三つ目が、本稿の主要目的である。

### 3. 知識表現

本論文では、知識表現のために 2 種類の否定を導入する。

**強否定** 強否定は古典的な論理で用いる否定と同じである。これは、ある命題に対して「正しくない」という意味を表す。強否定は  $\neg$  で表記する。

**弱否定** 弱否定は否定に関する仮説を表す。これは、ある命題に対して「...とはかざらない」という意味を表す。弱否定は  $\sim$  で表記する。

以下、弱否定のついたリテラルを弱リテラルと呼び、それ以外のリテラルを強リテラルと呼ぶ。弱リテラル  $\sim L$  は、仮定として  $\neg L$  をもつ。

**定義 1 (規則)** 規則を下記の形で定義する。

$$L_0 \Leftarrow L_1 \wedge \dots \wedge L_j \wedge \sim L_{j+1} \wedge \dots \wedge \sim L_n.$$

ここで  $L_i$  ( $0 \leq i \leq n$ ) は強リテラルである。 $\wedge$  は連言 (and) を表し、 $\Leftarrow$  は含意を表す。含意の左辺の強リテラルを結論、右辺のリテラルの集合を前提という。この規則は、 $\neg L_{j+1}, \dots, \neg L_n$  という仮定を持つ [4][5]。また、事実を表す規則 (事実節) は、前提部分を必要としない。

エージェントの知識ベースは、規則の集合である。

### 4. 論証

**定義 2 (論証)** 次の条件を満たす規則の列  $[r_1, \dots, r_n]$  を論証とする。

- (1) 規則  $r_i$  の前提に含まれる強リテラル  $L$  はそれぞれ規則  $r_k$  の結論に現れる ( $1 \leq k < i$ )。

(2)  $r_i$  ( $1 \leq i < n$ ) の結論  $L$  は規則  $r_j$  ( $i < j \leq n$ ) の前提に現れる。

(3) 一つの議論に同じ結論を持った2つ以上の規則は含まれない。

定義 3 (部分論証) 論証  $A_1 = [r_1, \dots, r_n]$  に対し, その部分列  $A_{sub} = [r_z, \dots, r_k]$  が論証の定義を満たすとき,  $A_{sub}$  を  $A_1$  の部分論証という。

定義 4 (論証木の定義) 論証  $[r_1, \dots, r_n]$  の論証木は以下の条件を満たすリテラルの木である。

- (1) 規則  $r_n$  の結論を論証木の根とする。
- (2) 各ノード  $L_i$  の子は  $L_i$  を結論とする規則の前提に含まれる各リテラルである。

例 1 (論証) 次の規則の列  $Z$  は, 論証である。

$$Z = \left[ \begin{array}{l} C. \\ B \Leftarrow C \wedge \sim D. \\ A \Leftarrow B. \end{array} \right]$$

$Z$  を論証木で表したものを図 1 に示す。

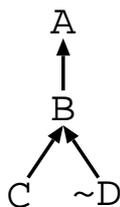


図 1: 論証木の例

## 5. 論証変換

我々は, ネットワーク上に存在するエージェントの行動シナリオを次のように考える。エージェントは, 自分の問題点を論証の形で提案する。しかし, 多くの場合論証は, 不確かな信念や知識より構成される。エージェントはその論証を自分自身, つまり, 自分の信念のためにより説得力のあるものに変換しようとする。

そのため, エージェントは, 知識ベースと推論エンジンをともない, ネットワーク上に存在する様々なエージェントの元を訪れる。ネットワーク上に存在するエージェントは, 自分の信念と知識を持ち, 自分の目的を達成するために行動をしている。それらのエージェントと議論し, 得られた論証を用いて論証変換を行なうことで自らの論証を以前よりもよりよくすることを試みる。ただし, 論証変換は, 自分の論証にいつでも適用できるわけではない。他のエージェントから提案された論証を受け入れるには, 以下のような受理条件を満たす必要がある。

定義 5 (論証受理条件) 自分以外エージェントから提供された論証を  $A$  とする。

- (1) エージェントの持つ規則の結論と仮定を否定する結論が,  $A$  の規則に含まれる, もしくは  $A$  に含まれる規則の仮定を否定するものがエージェントの知識ベースに含まれる場合,  $A$  を受理しない。
- (2) 変換の対象となる論証中の事実節の数が変換によって増加するときに限り, エージェントは, 論証変換を行なう。

(3) 変換の対象となる論証に含まれる弱リテラルの数が変換によって増加しないときに限り, エージェントは論証変換を行なう。

(4) エージェントの知識に含まれるリテラルと,  $A$  中のリテラルに同じものが存在した場合,  $A$  は受理しない。

(1) は, エージェントが自分の知識ベースを無矛盾状態に維持するために必要である。また, 議論において無効化や反論 (これは議論システムにおいてよく用いられる典型的な攻撃関係である [3]) が他のエージェントによって行なわれるのを防ぐ。(2) は, 論証中の事実数が多いほど論証の説得力があるという考えに基づく。(3) は, 論証が不確実なものに変換されるのを防ぐ。なぜなら, 変換以前よりも変換後の論証に含まれる弱リテラルの数が増えた場合, 置き換えによって根拠の薄い証拠が増えるからである。(4) は, 変換によって, 規則のループが生じたり, 論証の定義の条件 (3) を満たさなくなることを防ぐ。

我々は, 論証変換として, 以下の三つの方法を導入する。

定義 6 (規則変換) 論証  $A_1$  とその部分論証  $A_{sub} = [r_z, \dots, r_k]$  を仮定し,  $r_k$  の結論を  $L$  とする。次に, 論証  $A_1$  から  $r_k$  を抜き取り, さらに, そこから定義 2(2) を満たさない規則を抜き取った規則列  $A_2$  を得る。最後に,  $r_k'$  の結論が  $L$  であるような論証  $A' = [r_z', \dots, r_k']$  を他のエージェントから得て,  $A_2$  の先頭にくわえ論証  $A_3$  を構成する。この  $A_3$  を以前の論証  $A_1$  と置き換える。

規則変換は, ある結論  $L$  に対して自分が持っていない根拠を他のエージェントから受けとり, その新しい根拠  $A'$  を用いて異なる論証を生成するという変換である。

規則変換によって, 今までには無い多角的な根拠より成り立つ論証を構成することができる。

定義 7 (弱リテラル変換) 弱リテラルは, 仮定に基づく根拠の無い前提である。そのため, 弱リテラルを強リテラルに変換し, さらに必要となる根拠を加える。つまり, ある論証  $A_1$

$$A_1 = \left[ \begin{array}{l} \vdots \\ r_g : L_0 \Leftarrow L_1 \wedge \dots \wedge L_j \wedge \\ \quad \sim L_{j+1} \wedge \dots \wedge \sim L_m \wedge \dots \wedge \sim L_n. \\ \vdots \end{array} \right]$$

中の規則  $r_g$  において, 前提の弱リテラル  $\sim L_m$  を強リテラル  $\neg L_m$  に変換し, できた規則列を  $A_1'$  とする。そして,  $\neg L_m$  を論証の末尾の規則の結論としてもつ論証  $A_2$

$$A_2 = \left[ \begin{array}{l} \vdots \\ r_z : \neg L_m \Leftarrow L_1' \wedge \dots \wedge L_j'. \\ \vdots \end{array} \right]$$

を規則列  $A_1'$  の先頭に付加することで, 論証  $A_3$  を得る。なお, 論証受理条件 (3) より,  $r_z$  ならびにそれを成り立たせるための論証  $A_2$  内の規則は, 前提に弱リテラルを持たない。

$$A_3 = \left[ \begin{array}{l} \vdots \\ r_z : \neg L_m \Leftarrow L_1' \wedge \dots \wedge L_j'. \\ \vdots \\ r_g' : L_0 \Leftarrow L_1 \wedge \dots \wedge L_j \wedge \\ \quad \sim L_k \wedge \dots \wedge \neg L_m \wedge \dots \wedge \sim L_n. \\ \vdots \end{array} \right]$$

弱リテラル変換によって論証中の不完全な箇所を減じることとなり、論証が完全化される。

定義 8 (事実変換) ある論証  $A_1$  中から事実節  $r_z : L_z$  を取り除き、できた規則列を  $A_1' = [r_1, \dots, r_{z-1}, r_{z+1}, \dots, r_n]$  とする。そして、 $L_z$  を結論として持つ規則  $r_z' : L_z \Leftarrow L_i \wedge \dots \wedge L_j$  を成り立たせるための論証  $A_z = [r_y', \dots, r_z']$  を、 $A_1'$  の先頭に付加する。その結果が論証として成り立つ場合、以前の論証  $A_1$  を新しい論証  $A_2 = [r_z', \dots, r_y', r_1, \dots, r_{z-1}, r_{z+1}, \dots, r_n]$  に置き換える。これにより、証拠の多い洗練された論証を構成することができる。なお、論証受理条件 (3) より、 $r_z'$  ならびにそれを成り立たせるための論証  $A_z$  内の規則は前提に弱リテラルをもたない。

他のエージェントから論証を貰い、この三つの変換を適用することで、エージェントの論証を補強し、よりよくなることができる。これは、協調の提示の一種として捉えることができる [5]。

また、エージェントは部分論証を受け入れることによって知識や規則を学ぶことができる。これは、議論 (コミュニケーションや対話) とモビリティを通して得られた知識獲得、知識発見といえるかもしれない。上記で導入された三種類の変換は、図 2 のように、木の形で示すと分かり易い。

例 2 (論証変換) 図 2 に論証変換によって変化した論証木の交換前と交換後の様子を示す。

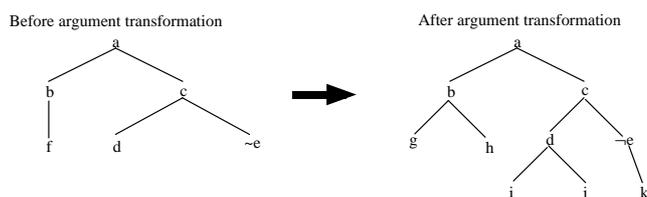


図 2: 論証木を用いて表した論証変換の例

図 2 では、ノード  $b$  を含む部分木が同一のノード  $b$  を含む新たな部分木によって変換されている (規則変換)。また、弱リテラル  $\sim e$  を含む葉は、強リテラル  $e$  と証拠  $k$  を持つ新たな部分木によって拡張されている (弱リテラル変換)。さらに事実  $d$  も、二つの証拠  $i$  と  $j$  を持つ新たな部分木によって拡張されている (事実変換)。以上より、ノード  $c$  を含む部分木は、上記の変換を取り入れたより大きな木へと変形された。

## 6. 適用例

我々は、本稿で提案したモバイルエージェントを以下の問題に適用することに成功した。

### (1) データマイニングへの適用

モバイルエージェントは、遺伝子組み換え食品 (遺伝子組み換え大豆など) に反対という自分の信念を議論を通し確認するため、様々な根拠を持ち、自分の信念をサポートしてくれるエージェントの元を訪れる。

このアプローチは、ユーザが与えたキーワードに沿ったものを検索するインターネット検索のような、通常の検索とは明らかに違う。論証形式の検索では、エージェント自身に自分なりの信念が存在し、その信念を補強する論証を他のエージェントから集めてくる。これにより論点が明らかになり、エージェントの目的が洗練され、論証が完全化し、様々な根拠を持つ論証を見つけることができる。このモデルでエージェントは、他のエージェントの



図 3: 知恵を授かるモバイルエージェントの実行例

元へ訪れ議論を交わす間、他のエージェントからの意見で自分の論点が変わることはない。

### (2) 協調定理証明への適用

我々は、完全な知識を持っているエージェントが存在しない分散環境で生じる推論問題をとって、定理の証明方法のみを知るエージェントと公理のみを知るエージェントが知識獲得による協調によって証明を完成させるという問題にシステムを適用した。

どちらの例でも、モバイルエージェントは、多くのエージェントの元を訪れ、他のエージェントから提供された論証を受け取った後にその所有者の元へ戻ってくる。それにより、他のエージェントの知識や知恵を得て自身の論証をより説得力のあるものにする。これによりエージェントは、成長したといえる。

## 参考文献

- [1] Chesñevar, C. I., Maguitman, A. G. and Loui, R. P. Logical Models of Argument. *ACM Computing Surveys*, 32(4): 337-383, 2002.
- [2] Kawakami, W., Umeda, Y. and Sawamura, H. Agent that Grow by means of Arguability and Mobility. in *Proc. of the Sixth Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems (AI2002)*, pp. 33-40. Nov 2002.
- [3] Prakken, H. and Sartor, G. Argument-based Extended Logic Programming with Defeasible Priorities. *J. of Applied Non-Classical Logics*, 7(1):25-75, 1997.
- [4] Sawamura, H., Umeda, Y. and Meyer, R. K. Computational Dialectics for Argument-based Agent Systems. in *Proc. of the Fourth International Conference on Multi-Agents Systems (ICMAS2000)*, pp. 271-278. IEEE Computer Society, July 2000.
- [5] Umeda, Y., Yamashita, M., Inagaki, M. and Sawamura, H. Argumentation as a Social Computing Paradigm. in *Proc. of 3rd Pacific RIM Int. Workshop on Multi-Agents (PRIMA2000)*, Lecture Notes in Artificial Intelligence, Vol. 1881, pp. 46-60. August 2000.