

論 文

議論に基づくスケジュール調整エージェント

福本 太郎[†] 澤村 一^{††a)}

Argumentation-Based Scheduler Agents

Taro FUKUMOTO[†] and Hajime SAWAMURA^{††a)}

あらまし これまでにスケジュール調整について多くの研究がなされてきたが、多くが各スケジュールに対するユーザの選好基準を数値に変換し、これを量的に評価し調整するものだった。このアプローチには二つの問題がある。(1) ユーザの選好基準には一般的に不確実な情報が含まれ数値への変換が難しい、(2) どのような基準に従い数値に変換したのか分からなくなる。実際に我々が行うスケジュール調整では、どのような理由によってスケジュールを決定したのか、が提示されることは重要である。しかしこれには主観的な情報、不確実な情報が含まれることがあり、これらを直接用いてスケジュール調整を行うことは難しい。一方で近年、分散したあいまい性を含んだ情報をもとに合意を形成する手法として議論が注目されている。そこで本論文ではあいまいな情報をもとに議論することによってスケジュールを調整する新たなスケジュール調整エージェントシステムを提案する。

キーワード マルチエージェントシステム, スケジューリング, 議論

1. ま え が き

近年、コンピューターシステムや情報通信技術の飛躍的な発展に伴い、人間の意思決定支援を行うためのマルチエージェントシステムに関する研究が盛んに行われている。中でも、不完全な知識をもった多数のエージェントが合意を形成するための手法として対話の形式の一つである議論が注目され、これを実装した議論するマルチエージェントシステムが提案されている [1]。我々が日常生活においてありとあらゆる問題を議論によって解決を計ることからも分かるように、議論はそのための強力なツールであり、様々な分野へ応用可能である。これまでに批判的・論理的思考のための e-Learning ソフトウェア [2] や、Web サイト推薦システム [3] などが提案されている。

エージェントシステムの得意とする分散処理の応用例として、スケジュール調整を行うエージェントシス

テムがある [4]~[9]。これまでのスケジュール調整システムの多くの関心は、ユーザの好みをいかに数値に変換するかということと、数値で表現された各ユーザのスケジュールをどのように定量的に評価するかの二つに終始している。このようなシステムの場合、ユーザは各スケジュールに対して自分で重要度を割り振っていく必要がある。しかしここで実際に私たちがスケジュール調整を行う場面を考えてみると、(i) 仕事のスケジュールを何よりも優先する、(ii) 既に決まっているスケジュールを優先する、など様々な選好基準 (preference) を組み合わせ用いており、単純にすべてのスケジュールに対して一定の基準で重要度を割り振っていく作業は難しくまた煩雑でもある。更に数の大小など量的な情報に基づき個人の好みを比較することは好ましくない。そのために数値などの定量的な評価ではなく、選好基準などの定性的な評価に基づく新たなスケジュール調整方法が必要だと考えられる。

そこで本研究では、ユーザが自らのエージェントに知識として上記のような選好基準を与え、エージェントが他のエージェントとの議論を通し合意を形成することでスケジュールを調整する新たなスケジュール調整エージェントシステムを提案する。本研究ではエージェントの知識表現言語に拡張注釈付き論理プログラム (EALP: Extended Annotated Logic Program-

[†] 新潟大学大学院, 新潟市

Graduate School of Science and Technology, Niigata University, 8050-2-cho, Ikarashi, Niigata-shi, 950-2181 Japan

^{††} 新潟大学工学部, 新潟市

Institute of Natural Science and Technology Academic Assembly, Niigata University, 8050-2-cho, Ikarashi, Niigata-shi, 950-2181 Japan

a) E-mail: sawamura@ie.niigata-u.ac.jp

ming), 議論フレームワークに多値議論の論理 (LMA: Logic of Multiple-valued Argumentation) を採用する [10]. 我々の提案した EALP は, あらゆる真理値の完備束を注釈として用いることができる. この多値性から不確実な知識, 矛盾を含む知識を表現することができる. LMA はあいまいな知識に基づいた議論を可能にする. そのため不確実な情報, 不確定な情報に基づき合意を形成しなければいけないスケジュール調整には都合が良い. 更に本システムではエージェント間で行われたスケジュール調整に関する議論をユーザが分析することによって「どうして自分の提案が受け入れられなかったのか」ということを知ることもできる, そのため本研究はユーザが却下されたスケジュールに対して代案を考える上での支援効果も期待できる.

まず 2. では, 関連研究と比較し本研究の特徴と意義を示す. 3. では, 本研究で用いるエージェントシステムの知識表現言語 EALP を概説し, 続く 4. では, 多値議論フレームワーク LMA の説明をし, 議論に基づくスケジュール調整フレームワークを与える. 5. では簡単な例を用いてスケジュール調整の流れとその意義を示す. そして 6. で結論づける.

2. 関連研究

スケジュール調整についてはこれまでに多くの研究がなされ, 実際にグループウェアとして開発されているものも多い. ここでは他のスケジュール調整システムと比較し本研究の特徴と意義を述べる.

木下らは, 欠席者の数と欠席時間の合計をもとにユーザがスケジュールを受け入れることの可能な日の候補リストを作り, 既にスケジュールの埋まっている参加予定者にそのスケジュールがキャンセル可能かどうか聞くアルゴリズムを提案した [4]. このシステムでは, ユーザは何度もキャンセル可能かどうかの問いに答える必要がある.

伊藤らは, ATMS を用いたスケジュール管理システムを提案した [5], [6]. [5] では, スケジュールを仮説で記述しそれぞれユーザに重要度を割り当てさせる. そして各エージェントのスケジュールの整合性を ATMS を用いて管理することによってスケジュールの調整を行う. また [6] では, 説得によって信念を翻意 (belief revision) するスケジュール調整方法を提案している. このシステムでは説得者と妥協者の 2 人の間で数値的な情報をもとに提案を受け入れ可能かどうか評価し, 提案を受け入れられなかった場合, 交渉による簡単な説

得を試みる. ただし信念の翻意は妥協者にのみ起こり得る.

Jennings ら [7] は, それぞれスケジュールにランクを付けて高いランクのスケジュールを受け入れるという単純なスケジュール調整を行っている.

一方で, 対話によってスケジュール調整を行うシステムとしては [8], [9] などがある.

Garrido らは, エージェント間でスケジュールに対する定量的な情報に基づく交渉によってスケジュール調整を行うシステムを提案した [8].

Haynes らは, どこまで妥協できるかというユーザの好みをしきい値で表現し, 交渉によってスケジュールを自動生成するシステムを提案した [9].

このように, これまでスケジュール管理を行うエージェントシステムについて行われてきた研究の多くはユーザの好みを量的に評価している. しかし単純に各スケジュールに付けられた重要度の評価によってスケジュール調整をしてしまうと, ユーザが恣意的にすべてのスケジュールに高い重要度を与えた場合, エージェントは適切なスケジュール管理ができなくなってしまふ. また, いかにしてスケジュールにランクが与えられたのかは示されないので, ユーザがそれを知ることが難しい. 実際に我々が日常で行うスケジュール調整を考えてみると, 他の人がなぜその重要度をそのスケジュールに与えたのかという説明をせず, 提案されたスケジュールを受け入れることは難しく, 量的な評価だけでは不十分だといえる.

本研究は, 以下の 3 点において既存のスケジュール調整についての研究とは異なると考える.

(1) 選好基準を用いて, 各スケジュールを質的情報に基づき評価する: 我々のスケジュール調整エージェントは選好基準を直接用いて, 他のエージェントと議論することによってスケジュール調整を行う. そのため量的情報に基づくスケジュール調整とは異なり, 各スケジュールに重要度を割り振っていく必要はない. また意図的な高ランク付けによって適切なスケジュール調整が行えなくなることもない.

(2) 議論によってスケジュール調整を行う: 対話の一つの形式である交渉や説得を用いてスケジュール調整を行わせる研究はあるが, 議論を用いたものはまだない. 日常に行われるスケジュール調整を考えてみても, 我々は交渉や説得をだけでなく議論することによってもスケジュールを調整している.

(3) どのような調整が行われたのか確認すること

ができる：議論を用いることによって、エージェントが合意に至る過程と議論において提出された論証はすべてユーザに示される。そのためユーザはなぜ自分の提案が却下されたのかを分析することができる。

また、スケジュール調整エージェントシステムで考察されるプライバシー保護について考える。本システムでは分散されたエージェントがそれぞれ独自の知識ベースをもってそれらの和集合をとることはない。更にユーザがエージェントに入力した個々のスケジュールを公開することもない。したがって、スケジュール調整において各エージェントは必要以上に情報をさらすことはなく各ユーザのプライバシーは守られているといえる。

3. 言語と知識ベース

本章では EALP の定義を述べる。EALP は矛盾や未定義状態をも扱うことができるため、本研究で扱う各エージェントの知識は無矛盾である必要はない。

[定義 1] (注釈と注釈付き原子式) [10] 真理値の完備束 (T, \leq) を仮定し、その最小要素と最大要素を \perp と \top によって表す。また最小上界オペレータを \sqcup によって表す。注釈とは T の要素 (定数注釈), T 上の注釈変数, 及び注釈項のいずれかである。注釈項は次のように再帰的に定義される。 T の要素と注釈変数は注釈項である。加えて, x_1, \dots, x_n が注釈項ならば, $f(x_1, \dots, x_n)$ は注釈項である。ただし f は $T^n \rightarrow T$ の形の全域で連続な注釈関数である。 A を原子式, μ を注釈としたとき $A:\mu$ を注釈付き原子式という。 $\neg: T \rightarrow T$ の形の注釈関数を仮定し, $\neg A:\mu = A:\neg\mu$ とする。このとき, $\neg A:\mu$ を $A:\mu$ の認識論的な明示否定という (注1)。

[定義 2] (注釈付きリテラル) [10] $A:\mu$ を注釈付き原子式とする。このとき, $\sim A:\mu$ を $A:\mu$ の存在論的な明示否定という。注釈付きオブジェクトブリエテラルとは $\sim A:\mu$ か $A:\mu$ のいずれかである。存在論的な明示否定の記号 \sim は注釈付きオブジェクトブリエテラルの補リテラルを表す場合にも用いる。すなわち, $\sim\sim A:\mu = A:\mu$ とする。 L を注釈付きオブジェクトブリエテラルとしたとき, **not** L を L のデフォルト否定といい, その形を注釈付きデフォルトリテラルという。また, 注釈付きリテラルとは **not** L か L のいずれかである。

[定義 3] (拡張注釈付き論理プログラム) [10] 拡張注釈付き論理プログラム (Extended Annotated Logic

Program: EALP) は $H \leftarrow L_1 \& \dots \& L_n$. という形式の規則の集合である。ここで, H は注釈付きオブジェクトブリエテラルであり, L_i ($1 \leq i \leq n$) は注釈付きリテラルである。ただし, 各 L_i に現れる注釈は定数注釈か注釈変数である。

規則の頭部は規則の結論という。また, 本体に含まれる注釈付きオブジェクトブリエテラルは規則の前提, 注釈付きデフォルトリテラルは規則の仮定という。表 1 に真理値の完備束を $\mathcal{N}[0, 9]$ 真理値の順序を数の

表 1 知識ベース記述例
Table 1 Example of knowledge bases.

エージェント A の知識ベース (KB_A)	
$do(seminar, Date):7$	
$\leftarrow \text{not } plan(Date):8 \& \sim holyday(Date):9$	(スケジュールが存在しない日にセミナーを行う)
$\sim do(seminar, thursday):6$	
$\leftarrow group_activity(wednesday):9$	
$\& parttime_job(thursday):8$	
$\& \text{not } \sim plan(thursday):7$	(木曜日には前日にサークル活動がありバイトもあるのでどちらかといえばセミナーを行いたくない)
$\sim do(seminar, wednesday):6$	
$\leftarrow group_activity(wednesday):9$	
$(水曜日にはサークル活動があり$	
$\text{あまりセミナーを行いたくない})$	
$\sim do(seminar, monday):8 \leftarrow enjoy_holiday:9$	
$(休日を楽しみたく月曜日にはセミナーを行いたくない)$	
$enjoy_holiday:9 \leftarrow$	
$(休日を楽しみたい)$	
$group_activity(wednesday):9 \leftarrow$	
$(水曜日にはサークル活動がある)$	
$parttime_job(thursday):8 \leftarrow$	
$(木曜日にはバイトがある)$	
エージェント B の知識ベース (KB_B)	
$\sim do(seminar, tuesday):6 \leftarrow course(tuesday):6$	
$(火曜日には講義にでたいので$	
$\text{できればセミナーを行わないでほしい})$	
$\sim do(seminar, saturday):9 \leftarrow$	
$\sim do(seminar, sunday):9 \leftarrow$	
$(休日にはセミナーを絶対行いたくない)$	
$course(tuesday):6 \leftarrow$	
$\sim do(seminar, Date):7 \leftarrow have_low_energy(Date):9$	
$(月曜と金曜は元気がないので,$	
$\text{どちらかといえばセミナーを行いたくない})$	
$do(seminar, thursday):8 \leftarrow \text{not } plan(thursday):7$	
$(スケジュールが存在しない木曜日にセミナーを行う)$	
$have_low_energy(monday):9 \leftarrow$	
$(月曜日は元気がない)$	
$have_low_energy(friday):9 \leftarrow$	
$(金曜日は元気がない)$	
$\sim plan(Date):9 \leftarrow \sim parttime_job(Date):8$	
$(平日にバイトを認めない)$	
$\sim parttime_job(Date):8 \leftarrow weekday(Date):9$	
$(平日にはバイトを欠席理由に絶対認めない)$	

(注1): 注釈付き原子式の認識論的な明示否定 $\neg A:\mu$ は注釈付き原子式 $A:\neg\mu$ の形で記述するものとし, 以下では暗に扱う。

大小関係とし EALP で記述した知識ベースの例を示す。括弧内は規則の直観的な読みである。

4. スケジュール調整エージェント

本章では議論と論証について述べた後スケジュール調整フレームワークを与える。

4.1 議論フレームワーク

この小節では論証と議論、正当化について説明する。
[定義 4] (還元と極小還元) [10] P を EALP, C_i ($1 \leq i \leq k$) を次の形式の P の中の規則とする:

$$A: \rho_i \leftarrow L_1^i \& \dots \& L_{n_i}^i.$$

ここで A は原子式である。また、 $\rho = \sqcup\{\rho_1, \dots, \rho_k\}$ とする。このとき、次の形式の規則を P の還元と呼ぶ。

$$A: \rho \leftarrow L_1^1 \& \dots \& L_{n_1}^1 \& \dots \& L_1^k \& \dots \& L_{n_k}^k.$$

$\rho = \sqcup S$ となるような空集合以外の真部分集合 $S \subset \{\rho_1, \dots, \rho_k\}$ が存在しなければ、上記の還元を極小還元と呼ぶ。

[定義 5] (論証) [10] P を EALP とする。 P における論証とは次の条件を満たす規則の有限列 $Arg = [r_1, \dots, r_n]$ である。すべての i ($1 \leq i \leq n$) について、(1) r_i は P の規則、若しくは、 P の極小還元である。(2) r_i の本体にあるすべての注釈付き原子式 $A: \mu$ に対し、頭部が $A: \rho$ ($\rho \geq \mu$) となるような r_k ($n \geq k > i$) が存在する。(3) r_i の本体にあるすべての注釈付き原子式の存在論的な明示否定 $\sim A: \mu$ に対し、頭部が $\sim A: \rho$ ($\rho \leq \mu$) となるような r_k ($n \geq k > i$) が存在する。(4) 条件 1 から 3 を満たし r_1 を含んでいるような $[r_1, \dots, r_n]$ の真部分列が存在しない。

論証 Arg を構成する規則の結論を Arg の結論と呼び、その仮定を Arg の仮定と呼ぶ。 Arg の結論の集合を $concl(Arg)$ 、仮定の集合を $assm(Arg)$ と書く。また P から作られるすべての論証の集合を $Args_P$ と表す、例 1 の知識ベース KB_A からは、例えば以下の論証 Arg を作成できる。

$$\begin{aligned} Arg = & [\sim do(seminar, thursday):8 \leftarrow \\ & group_activity(wednesday):9 \\ & \& partime_job(thursday):8 \\ & \& \mathbf{not} \sim plan(thursday):7, \\ & group_activity(wednesday):9 \leftarrow, \end{aligned}$$

$$partime_job(thursday):8 \leftarrow]$$

これは木曜日のセミナーを認めないことを主張する論証である。マルチエージェントシステム $MAS = \{KB_1, \dots, KB_n\}$ のすべての論証の集合 $Args_{MAS}$ は $Args_{MAS} = \{Args_{KB_1} \cup \dots \cup Args_{KB_n}\}$ である。論証の集合が与えられると正当化された論証は、議論木を構築する対話的証明論により決定できる。

[定義 6] (議論) [10] 議論とは次の条件を満たす提議 $move_i = (Player_i, Arg_i)$ の空でない有限列である。ただし $Arg_i \in Args_{MAS}, i \geq 1$

(1) $Player_i =$ 提案者 $\Leftrightarrow i$ が偶数; $Player_i =$ 反対者 $\Leftrightarrow i$ が奇数,

(2) $Player_i = Player_j = P$ ($i \neq j$) ならば $Arg_i \neq Arg_j$,

(3) Arg_i は Arg_{i-1} を打破している。

本研究で扱うエージェントは論証を提出できる場合は必ず提出する。そして議論はすべてのエージェントから論証が提出されなくなった時に終了する。条件 1 は、議論が提案者から始まり、提案者と反対者が交互に提議することを表している。条件 2 は議論が循環することを防いでいる。ここで提案者による同じ論証の再提議を禁止したので、議論が循環するような場合は必ず反対者の提議で停止することになる。条件 3 は、提案者、反対者ともに攻撃関係として打破を用いることを表す。本研究で用いる攻撃関係の定義を以下に示す。

[定義 7] (反論) [10] Arg_α が Arg_β を反論する $\Leftrightarrow \mu_1 \geq \mu_2$ である $A: \mu_1 \in concl(Arg_\alpha)$ と $\sim A: \mu_2 \in concl(Arg_\beta)$ が存在するか、 $\mu_1 \leq \mu_2$ である $\sim A: \mu_1 \in concl(Arg_\alpha)$ と $A: \mu_2 \in concl(Arg_\beta)$ が存在する。

[定義 8] (無効化) [10] Arg_1 が Arg_2 を無効化する $\Leftrightarrow \mu_1 \geq \mu_2$ である $A: \mu_1 \in concl(Arg_1)$ と $\mathbf{not} A: \mu_2 \in assm(Arg_2)$ が存在するか、 $\mu_1 \leq \mu_2$ である $\sim A: \mu_1 \in concl(Arg_1)$ と $\mathbf{not} \sim A: \mu_2 \in assm(Arg_2)$ が存在する。

[定義 9] (打破) [10] Arg_1 が Arg_2 を打破する $\Leftrightarrow Arg_1$ が Arg_2 を無効化するか、 Arg_1 が Arg_2 を反論し、かつ Arg_2 が Arg_1 を無効化しない。

これらは議論で用いられる典型的な攻撃関係で、直観的には反論は論証の結論を否定し、無効化は論証の仮定を否定する。また反論は常に対照的で無効化は一方的である場合が多いが必ずしもそうなるとは限らない。打破は反論と無効化を組み合わせた攻撃関係で、

論証の仮定が否定されればその結論を主張できなくなると考え無効化を反論より強い攻撃関係として扱う。どの攻撃関係を採用するかで正当化された論証の集合は変化するが、ここでは [10] と同様提案者、反対者とも打破を用いる。

[定義 10] (議論木) [10] 議論木とは、すべての枝 (branch) が議論となる木である。ただし、すべての提議 $move_i = (提案者, Arg_i)$ に対し、その子は $(Arg_{i+1,j}, Arg_i) \in \text{打破}$ となるようなすべての (反対者, $Arg_{i+1,j}$) ($j \geq 1$) である。

[定義 11] (証明論的正当化) [10] 議論 D が勝利議論である $\Leftrightarrow D$ の終端が提案者の提議である。議論木 T が勝利議論木である $\Leftrightarrow T$ のすべての枝が勝利議論である。論証 Arg が証明論的に正当化された論証である $\Leftrightarrow Arg$ を根とした勝利議論木が存在する。

終端が提案者であることは Arg を攻撃する論証が全く存在しないか、 Arg を攻撃する反対者のすべての論証が提案者の論証から攻撃されていることを意味する。直観的には提案者の主張を退ける論証を反対者が作れず、反対者が Arg を受け入れた状態だといえる。議論とはある論証の集合から内部に攻撃関係の生じていない論証の集合を求めることが目的であり、これによって合意を形成する。説明の簡単のために月曜日と木曜日、どちらにセミナーを行うかという議題について表 1 に示した KB_A, KB_B の知識ベースを用いて実際に議論を展開した例を示す。月曜日と木曜日どちらも休日でないと仮定すると、エージェント A が「あまりやりたくないが月曜日には予定がないのでセミナーを行おう」(Arg_{A1}) と主張する、これに対しエージェント B が「月曜日には元気がないので嫌だ」(Arg_{B1}) と主張する、 Arg_{A1} と Arg_{B1} は、定義 7 から互いに反論し合うので、定義 9 より Arg_{A1} と Arg_{B1} は互いに打破し合っている。ここで定義 6 の条件 2 からエージェント A は Arg_{A1} を Arg_{B1} を打破する論証として再提出することはできない。これ以上論証が提出されないのでこの議論は終了する。次にエージェント B が「予定がないので木曜日にセミナーを行おう」と主張する (Arg_{B2})、これに対しエージェント A は「前日にサークルがあり木曜にはバイトもあるので嫌だ」(Arg_{A2}) と主張するが、これに対し更にエージェント B が「平日のバイトを理由には認めない」(Arg_{B3}) と主張する。 Arg_{A1} と Arg_{B1} と同様 Arg_{A2} と Arg_{B2} は互いに打破し合う。更に定義 8 から Arg_{B3} は Arg_{A2} を一方的に無効化しているの

Arg_{B3} は Arg_{A2} を一方的に打破する。これ以上論証が提出されないのでこの議論も終了する。図 1, 図 2 にそれぞれの議論の議論木表現を示す。図中それぞれの黒枠には論証を木表現で示した。また defeat は打破を意味する。 Arg_{A1} を根とする議論木は終端が提案者でないので定義 11 より Arg_{A1} は正当化されない。また Arg_{B2} を根とする議論木はすべての終端が提案者なので定義 11 より Arg_{B2} は正当化される。

4.2 議論に基づくスケジュール調整の流れ

ここでは議論に基づくスケジュール調整の流れを示す。スケジュール調整エージェントは、ユーザが EALP で表現した独自の選好基準である知識ベースと共通の推論エンジン LMA をもっている、また仲介エージェント (Mediator Agent) はそれらをもたず議論の制御、監視のみを行う。図 3 にシステム構成を示す。エージェントは以下の流れにそってスケジュールを調整する。

(Step 1) ユーザは、EALP の規則の集合である選好基準とスケジュールを準備する。エージェントは、

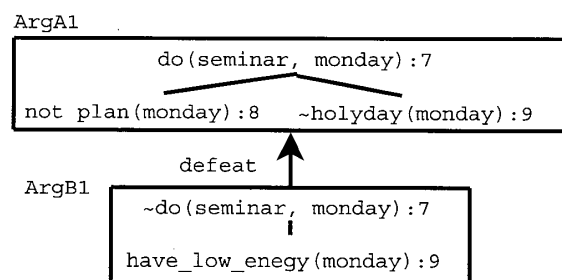


図 1 月曜セミナーについての議論木
Fig. 1 An Argumentation Tree of Monday seminar.

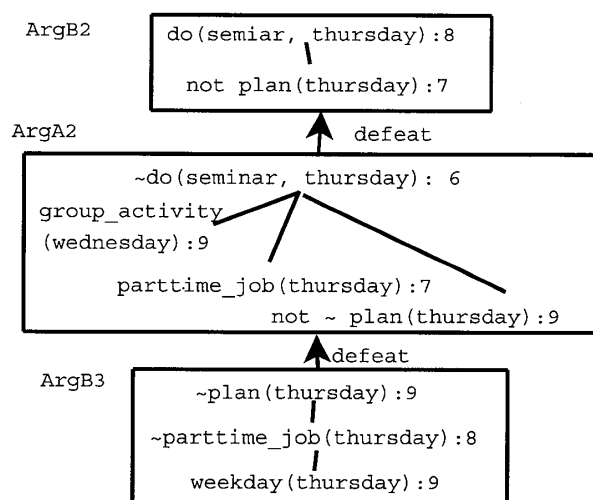


図 2 木曜セミナーについての議論木
Fig. 2 An Argumentation Tree of Thursday seminar.

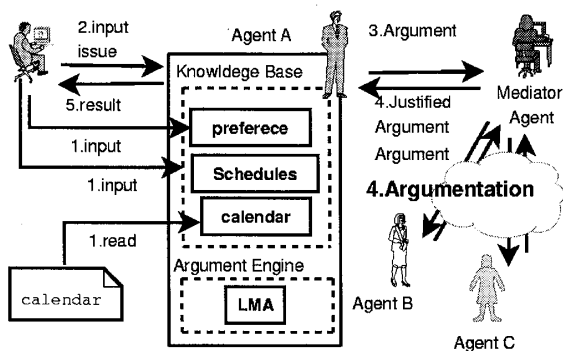


図 3 システム構成
Fig. 3 The system architecture.

ユーザの入力したスケジュールからスケジュールを述語名, 実行日を引数とした EALP の事実節, そして日を引数とした述語 $plan(Date)$ へ変換する。

(Step 2) ユーザが自身のエージェントに議題を入力すると, エージェントは, 論証 Arg を作成し仲介エージェントに送信する。

(Step 3) 仲介エージェントは, 受け取った論証の中から一つを選び, 各エージェントへ配布し定義 6 に従って議論を展開する。

(Step 4) 仲介エージェントは定義 11 に従い, 論証 Arg が正当化されたかどうか判定する。正当化された場合, 正当化された論証の集合 $JA \leftarrow Arg$ を追加する。仲介エージェントのもとにまだ議論すべき論証があった場合, 仲介エージェントは次の論証を各エージェントに配布し議論を展開する。

(Step 5) 仲介エージェントは議論結果として JA を各エージェントへ配布し議論を終了する。

(Step 6) 議論結果に基づきエージェントはスケジュール調整結果をユーザに示す。例えば $Date$ に X を行うというスケジュールがあったとすると, $do(X, Date)$ を結論にもつ論証が正当化された論証の集合に含まれていた場合, X を $Date$ に行うという主張が, 議論に参加したエージェントの間で受け入れられたことを示す。

(Step 7) ユーザはエージェントの提案を受け入れるのであればエージェントの示すスケジュールをエージェントへ入力する。

5. 実行例

ここでは 4. で与えたスケジュール調整の簡単な実行例を示す。ここでは表 1 に示したエージェント A, B と, 更に表 2 で示したエージェント C の合計 3 体

表 2 エージェント C の知識ベース
Table 2 A knowledge base of agent C.

Agent C (KB_C)	
$do(seminar, Date):8 \leftarrow$	$not\ plan(Date):8$ (スケジュールが存在しない日にセミナーを行う)
$\sim do(friday):8 \leftarrow$	$lecture(friday):9.$
$\sim do(wednesday):8 \leftarrow$	$lecture(wednesday):9.$ (水金は講義があるのでセミナーは行えない)
$parttime_job(thursday):8$	$\leftarrow working_student(agent_A):9$
$working_student(agent_A):9 \leftarrow$	(エージェント A は苦学生であり, 木曜日のバイトもやむを得ない)
$\sim do(seminar, saturday):8 \leftarrow$	
$\sim do(seminar, sunday):8 \leftarrow$	(休日にはセミナーを行わない)
$\sim have_low_energy(monday):9 \leftarrow$	(週の始まりから元気がないなんてことは認められない)
$\sim do(seminar, tuesday):6$	$\leftarrow have_low_energy(tuesday):8.$
$have_low_energy(tuesday):8 \leftarrow$	(火曜日は疲れているのでセミナーを行いたくない)

表 3 各エージェントのスケジュール
Table 3 Schedules of agents.

	日	月	火	水	木	金	土
agent A	休日	講義		サークル	バイト	講義	休日
agent B	休日		講義			講義	休日
agent C	休日		講義	講義		講義	休日

表 4 エージェント A のスケジュールの EALP 表現
Table 4 Translated schedules of agent A.

$lecture(monday):9 \leftarrow,$	$plan(monday):9 \leftarrow,$
$group_activity(wednesday):9 \leftarrow,$	$plan(tuesday):9 \leftarrow,$
$parttime_job(wednesday):9 \leftarrow,$	$plan(wednesday):9 \leftarrow,$
$lecture(friday):9 \leftarrow,$	$plan(friday):9 \leftarrow$

の知識ベースを用いる。説明の簡単のために表 3 に示した単純なスケジュールを考える。3 人とも共通でスケジュールが空いている日は休日を除いて存在しないためスケジュールを調整する必要がある。エージェントはまずユーザのスケジュールを EALP へ変換する。表 3 のエージェント A のスケジュールからは, 表 4 の規則が作成される。ここでは各スケジュールの注釈として最大値の 9 を与えている。多くの既存の研究では最高のランク付けを行うとスケジュールを適切に調整できない。

任意のエージェントから提議されると, EALP 表現へ変換されたスケジュールと選好基準を用いて論証の作成を試みる。この例ではエージェント A, C は自分のスケジュールが存在しない曜日, エージェント A は火木, エージェント C は月木にセミナーを行うことを主張するための論証を作成し, エージェント B は木曜

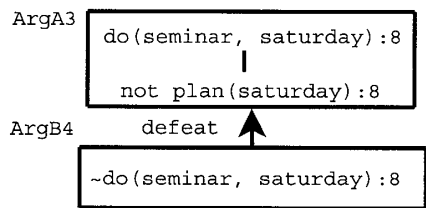


図 4 ArgA3 についての議論木
Fig.4 An argumentation tree of ArgA3.

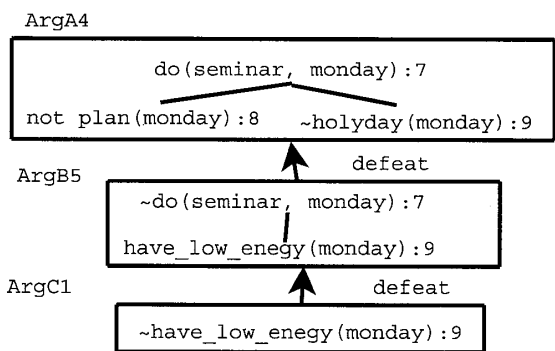


図 5 ArgA4 についての議論木
Fig.5 An argumentation tree of ArgA4.

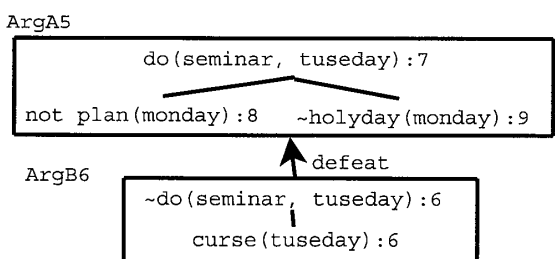


図 6 ArgA5 についての議論木
Fig.6 An argumentation tree of ArgA5.

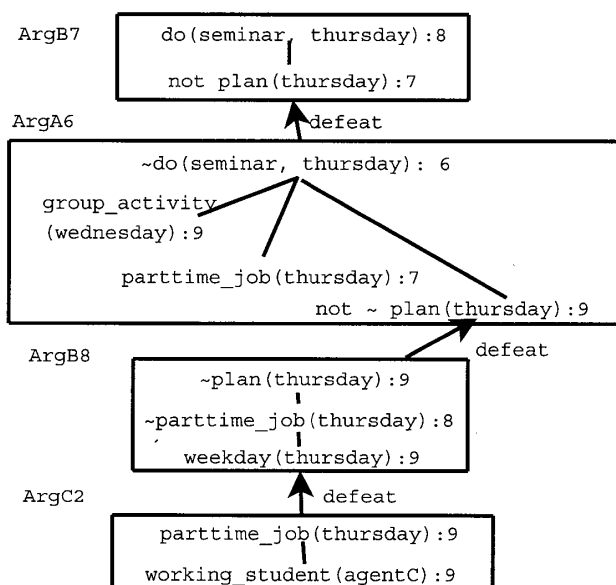


図 7 ArgB7 についての議論木
Fig.7 An argumentation tree of ArgB7.

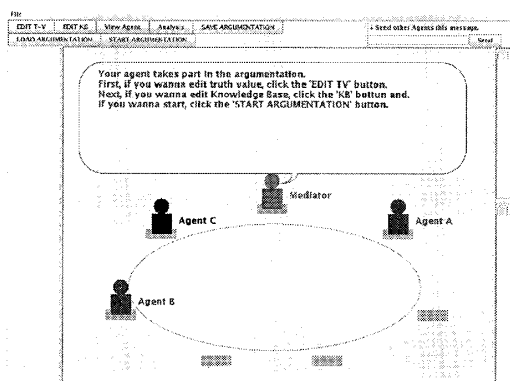


図 8 IAE のスクリーンショット
Fig.8 Screen Shot of IAE.

にセミナーを行うことを主張するための論証を作成する。エージェント A, C が土曜日にセミナーを行うという主張 (*do(seminar, saturday)*) もするが、図 4 に示すとおりエージェント B から打破され正当化されない。日曜日も同様。月曜日についての議論を図 5 に示す。図 1 に示した議論と異なり、エージェント C が新たに加わることによって $ArgA1 = ArgA4$ が正当化されるようになる。次に火曜についての議論を図 6 に示す。図 6 に示したとおり火曜日にセミナーを行うという論証はエージェント B に打破され正当化されない。最後に木曜日についての議論を図 7 に示す。図 2 に示したものと異なり、新たにエージェント C が $ArgB8$ へ打破するため *do(seminar, thursday)* についての論証は正当化されない。本システムでは議論参加者によって正当化を受ける論証の集合は変化する場合があります。議論の結果、月曜日にセミナーを行うとい

う結論をもつ論証 $ArgA4$ がこの 3 体のエージェントの間で受け入れられた。議論全体を見るとエージェント A はもともとあった講義をあきらめ、エージェント B は月曜日には元気がないがしぶしぶセミナーを行うことになる。

図 8 は我々の開発した IAE (Integrated Argumentation Environment) [11] に議論に基づくスケジュール調整フレームワークを実装し、実行したものである。

6. むすび

本論文では議論に基づきスケジュール調整を行うエージェントを提案した。これまでに行われたスケジュール調整を行うエージェントに関する研究は、重要度をもとに、各スケジュールに対して量的な評価を行うことによってスケジュールを調整していたが、本

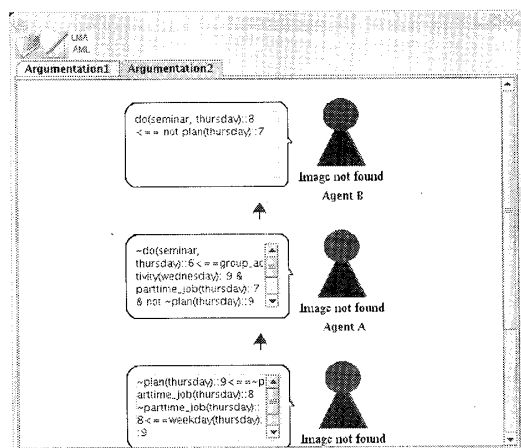


図9 議論木ビューア

Fig. 9 Screen Shot of argumentation tree viewer.

研究では今まで行われた研究と異なり、数値の比較ではなく選好基準を直接戦わせ、エージェント同士が議論することによってスケジュール調整を行う。本論文で扱った真理値の完備束 $\mathcal{N}[0, 9]$ は、これまで扱うことが難しいとされた不確実な選好基準を記述することができる。また真理値の完備束に $\mathcal{N}[1, 31]$ のべき集合を採用することで時間情報を表現することも、古典的な2値論理を用いることも可能であり本研究のエージェントは高い多様性をもつ。また、5.で示したとおりに単純に各スケジュールへ極端に大きな注釈を与えたからといってすべてのエージェントに受け入れられるとは限らない。これは本システムが数の大小ではなく論証間の攻撃関係に基づく選好基準の優劣によってスケジュールを質的に評価しているからである。スケジュールに重要度を与える場合、ユーザはスケジュールごとに数値をじっくり考えて記述する必要があるが、本研究では一度選好基準を自身のエージェントに与えてしまえば注釈を意識する必要はなく、ユーザがすべきことはスケジュールを入力することだけである。しかし現在の我々のスケジュール調整エージェントは、選好基準を入力する際にユーザはEALPを習得している必要があり、この点において量的な重要度を与えるよりは煩雑であり難解であるといえる。しかしこれは我々の提案するIAE [11]を用いることで解決できる。一方で、議論を用いることによってこれまで注目されなかった各エージェントが合意に至る道程を分析することが可能になる。ユーザは議論木からどの論証がだれからどのように攻撃されたのか詳細に知ることができる。これは再スケジュールリングの役に立つだろう。今後の課題としては以下のものを挙げることで

きる。

(1) ユーザが用意した選好基準の適切さを評価するプロセスが存在しないため、「 $1+1=2$ だからセミナーを行わない」など不適切な論証を正当化してしまう場合がある。

(2) エージェントの信頼度を考慮していないため、嘘のスケジュールを正当化してしまう場合がある。

(3) 本研究のエージェントはスケジュールを提案するだけで学習することはないが、他のエージェントのスケジュールや選好基準を知ることが効果的なスケジュール調整を実現する。

(1) 論証の適切さについては、[12]などの研究があり、議論への導入を現在研究中である。(2)については、エージェントの信頼性というエージェント技術の問題点として広く認識され、近年盛んに研究されている。これを導入することによって問題を改善できると考え現在研究中である。エージェントの議論に基づく学習については[13]で述べた。また利他的や利己的、信じやすい疑り深いなどのエージェントの性質や議論戦略を導入することによって、単純にスケジュールを調整するのではなくユーザにとって無理のないスケジュールを組むエージェントなどを実現できるだろう。

文 献

- [1] C. Chesñevar, G. Maguitman, and R. Loui, "Logical models of argument," *ACM Comput. Surv.*, vol.32, pp.337–383, 2000.
- [2] C. Reed and G. Rowe, "Araucaria: Software for argument analysis, diagramming and representation," *Int. J. AI Tools*, vol.14, no.3-4, pp.961–980, 2004.
- [3] C.I. Chesñevar and A.G. Maguitman, "Arguenet: An argument-based recommender system for solving web search queries," *Proc. 2nd International IEEE Conference on Intelligent Systems*, pp.282–287, IEEE Press, 2004.
- [4] 木下, 川村, 菅原, "モバイルエージェントによるスケジュール調整に関する研究," 第5回IEEE広島支部学生シンポジウム論文集, pp.205–206, 2003.
- [5] 伊藤, 新谷, "好みに基づく分散atmsを用いたグループスケジュール管理システムについて," 第34回人工知能基礎論研究会, pp.75–80, 1998.
- [6] 伊藤, 新谷, "エージェント間の説得に基づく議事スケジュールリングについて," 1997信学ソ大(通信), pp.120–125, 1997.
- [7] N.R. Jennings and A.J. Jackson, "Agent based meeting scheduling: A design and implementation," *IEE Electron. Lett.*, vol.31, no.5, pp.350–352, 1995.
- [8] L. Garrido and K. Sycara, "Multi-agent meeting scheduling: Preliminary experiment results," *Proc. Second International Conf. on Multi-Agent System*

ICMAS, pp.95-102, 1996.

- [9] T. Haynes, S. Sen, N. Aroa, and R. Nadella, "An automated meeting scheduling system that utilizes user preferences," Proc. First International Conf. on Autonomous Agents, pp.208-315, 1997.
- [10] T. Takahashi and H. Sawamura, "A logic of multiple-valued argumentation," Proc. Third International Joint Conference on Autonomous Agents and Multi Agent Systems, pp.800-807, ACM, 2004.
- [11] T. Isogai, T. Fukumoto, and H. Sawamura, "Integrated argumentation environment for arguing agents," Demonstration Track of Intt. Conf. on WI'06, 2006.
- [12] A.W. Bollen, "Relevant logic programming," J. Automated Reasoning 7, pp.563-585, 1991.
- [13] T. Fukumoto and H. Sawamura, "Argumentation based learning," Third International Workshop on Argumentation in Multi-Agent Systems, pp.59-76, Hakodate, Japan, 2006.

(平成 19 年 7 月 11 日受付, 11 月 10 日再受付)



福本 太郎

2003 新潟大・工・福祉人間工学卒。2005 同大学院自然科学研究科情報・計算機工学専攻修士課程了。現在、同大学院自然科学研究科情報理工学専攻博士課程に在学中。エージェントテクノロジー、機械学習に興味をもつ。



澤村 一 (正員)

1978 北海道大学大学院工学研究科情報工学専攻単位取得退学。博士(工学)1980~1996 (株)富士通国際情報社会科学研究所, (株)富士通研究所で室長, 主管研究員など歴任。オーストラリア国立大学, ニュージーランドヴィクトリア大学の Visiting Fellow を歴任。1996~現在, 新潟大学工学部情報工学科准教授。計算論理学, ソフトウェア基礎論, 人工知能分野に興味をもつ。情報処理学会, ソフトウェア科学会, 日本科学哲学会各会員。