

論証ダイアグラムを用いた動的議論支援ツールの提案

Dynamic Argumentation Support Tool Using Argument Diagram

桂 裕樹 岡田 将吾 新田 克己
Yuki Katsura Shogo Okada Katsumi Nitta

東京工業大学大学院総合理工学研究科

Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Japan

In this paper, we propose an argumentation support tool which displays justified arguments to users. Until now, though many visualization techniques of argumentation have been proposed, we cannot obtain justified arguments. We develop an argumentation support tool for visualization with a function of logical analysis. When the user inputs an utterance, the server converts it to a logical expression, and then converts it to an argumentation framework (AF). Furthermore, it calculates semantics of AF, displays the argumentation process in two graphs status in real time. As a result, this system gives the users several helpful information to take the next move.

1. はじめに

議論を分析評価するために、議論中の発言をルールや論理式とみなし、発言間の関係をダイアグラムなどを用いて議論の論理構造を視覚化する研究がこれまでも行われてきた [Toulmin 58]。この目的の1つは議論の対立点はどこか、議論による結論（正当化された主張）は何か、などを求めることであるが、議論が複雑になると、ダイアグラムが複雑になり、視覚的に結論を求めることは難しい。

一方で、議論の結論を求める手法として、議論構造を抽象的な論証の攻撃関係として表現し、その意味論を定義した Dung の議論フレームワーク AF [Dung 95] や、それを拡張した、価値に基づく議論フレームワーク VAF [Bench-Capon 03] の研究がある。AF は論証をベースとするため複雑な構造の視覚化に適している。

また、ルールで構成された論証木を AF に変換する手法 Aspic+ [Prakken 11] も提案されており、ルールによる議論を論証間の関係として抽象的に表現することも可能となった。

もし、現実の議論で意味論計算を行うことができれば、議論のどこを攻めればよいかといった戦略や自分の発言した意見が議論にどのような影響を持つか事前に知ることができ、議論はよりよいものになる。

しかし、これらの手法を実際に進行中の議論に適用しようとした PIRIKA [Katsura 14] では人がリアルタイムに論理式を入力し、論証木の作成を行うことは非常に困難であると分かった。

だが、Toulmin のように発言を論理式ベースで表現すれば、ダイアグラムで表示でき、発言の論理構造を正確に記述できるため、ツールとしての入力には Toulmin レベルで行うことが望ましい。一方で、議論が複雑になってくるとダイアグラムが巨大になり、議論全体の構造が見えにくくなるため、ダイアグラムが視覚的にわかりやすい、抽象的な AF で議論全体の構造を示すほうがよいと考えられる。

そこで、われわれは、ダイアグラムを利用してテキストベースで発言を入力し、その抽象的な論証構造を表示し、それに基

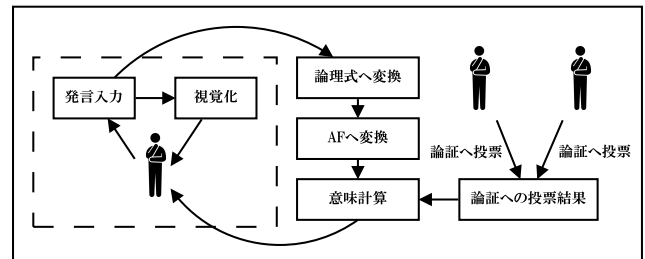


図 1: 本ツールが提供する機能

づいて議論を進行するための助言を与えることを目的とした議論支援ツールを開発する。

本ツールは「発言のダイアグラム」と「その議論フレームワークへの変換および意味論の計算」の2つのインターフェースを提供し、意味論の計算から議論戦略に関する材料を提供する。この様子を図 1 に示す。

従来研究ではユーザの発言をなんらかの形で視覚化し支援するのみであったが本研究ではリアルタイムに意味論を計算して、結果をユーザにフィードバックすることで支援を行う。

2. 関連研究

論証の可視化で最も広く使われているのは Toulmin モデル [Toulmin 58] である。ディベートの授業でも用いられおり、日常的な論証を分析・評価できる図式モデルである。このモデルは意見を「データ」、「論拠」、「裏付け」、「限定子」、「論駁」、「結論」の6つの要素に分けて考える。これにより一つの論証より正確に記述できる。

議論を最も客観的に表現・計算するのに適しているのは、Dung の手法であるが、実際の議論では個人の価値観が伴う。Bench-Capon は価値観を議論フレームワークに導入するために抽象議論フレームワークを拡張した価値に基づく議論フレームワーク [Bench-Capon 03] を提案した。この手法はノードにそれぞれ価値を割り振り、その大小関係をあらかじめ定義しておくことで、「客観的受理」と「あるユーザでの主観的受理」を定義した。

論証をより正確に記述するため、Prakken は論理式から議論フレームワークへ変換する Aspic+ [Prakken 11] を定義した。

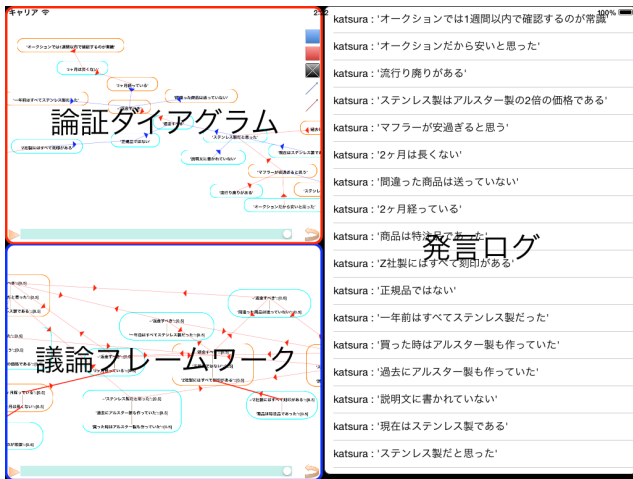


図 2: 画面例

Aspic+は与えられた知識およびルールの集合から論証木を作成して、論証木同士の攻撃関係をみることで議論フレームワークへ変換を行った。

以上までの手法はいずれも終了した議論を分析するため・保存するために用いられ、実際の議論中には用いることができない。

この問題を部分的に解決したのが PIRIKA [Katsura 14] である。PIRIKA では知識表現言語 EALP (Extended Annotated Logic Programming) を用い、Prolog 風にプログラミングすることで知識と攻撃関係を表現し、論証木を組み立て、議論の結果を計算することができた。実際の議論にも適用することができたが、EALP で記述することは完全である反面、論証単位で意見を述べるのがユーザにとって容易ではなかった。

3. ツールの概要

使用する機器は入力を直感的・高速にするため、タブレット型コンピュータの iPad*1 を用いる。対象となる議題は意見を互いに出し合い、ある議題が正しいか正しくないかを客観的に求めたいときに適する。たとえば、あるプロジェクトを進めるか否かを決めるような会議であったり、掲示板で行われるような議論である。

3.1 画面の外観

実際の画面を図 2 に示す。左上が論証ダイアグラム、左下が論証フレームワーク、右が発言ログを表す。

3.1.1 論証ダイアグラム

論証ダイアグラムは発言の論理関係をグラフ構造で表示した図である。この画面は議論の時系列情報を保存しているほか、ユーザの意見入力を受け付ける画面である。ユーザはこの画面に自身の意見と関係を同時に入力することで議論に参加する。

定義 1. 論証ダイアグラム

論証ダイアグラムは組, $AD = A, attacks, supports$ で与えられる。A は命題の集合, attacks および supports は AR 上の二項関係, すなわち, $attacks \subseteq A \times A$, $supports \subseteq A \times A$ である。

A は画面上のノードに対応し, attack および support はノード間の 2 種類のリンクを表す。

*1 iPad は Apple Inc. の登録商標

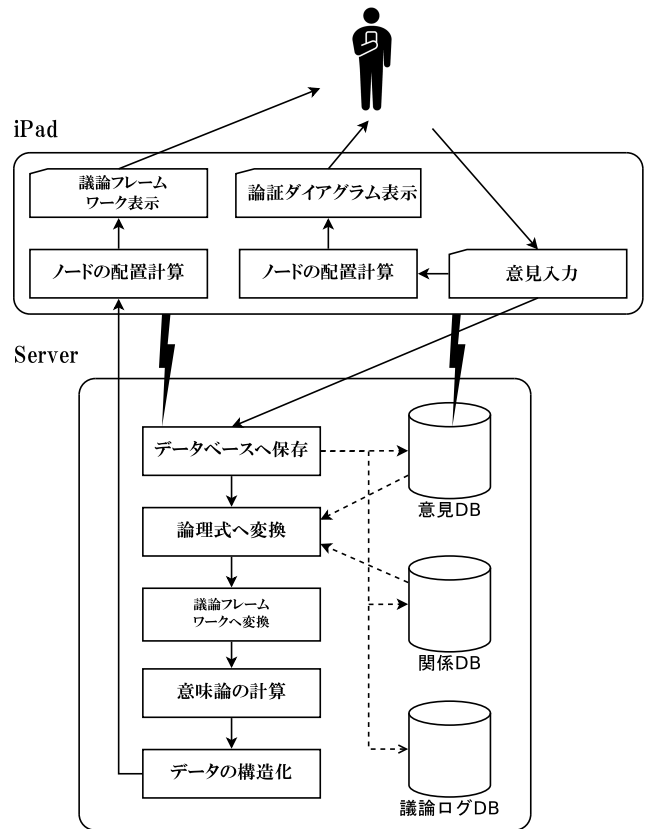


図 3: システムの動作図

3.1.2 議論フレームワーク

議論フレームワークは今まで入力されたすべての発言が総合的大小関係を考慮して論証木にまとめ、その論証木同士の関係を表示したものである。議論全体を見渡すのに適している。

定義 2. 価値に基づく議論フレームワーク [Bench-Capon 03]
価値に基づく議論フレームワークは組, $VAF = AR, attacks, V, val, valpref$ で与えられる。AR および attacks は抽象議論フレームワークと同様である。V は価値の空でない集合であり, val は AR の要素と V の要素を関連付ける関数である。valpref は $V \times V$ 上の選好関係である。

V は 0~1 までの連続値であり, valpref は数字の大小で決まるものとする。デフォルトでは各ノードに 0.5 が与えられている。

また、本ツールの論証ダイアグラムの表示では、結果を色分けし、どの意見が現時点で正しいのかを表示する。

3.1.3 発言ログ

発言ログは、誰がいつ意見を述べたかを時系列順に表示したり、システムからのメッセージを表示したりするものである。

3.2 システムの動作手順

システムの動作手順を図 3 に示す。ユーザは iPad を用いて自分の発言を、他の意見との関係とともに入力する。すると、入力された情報は二つに分岐する。1 つは即座に論証ダイアグラムへと反映される。もう 1 つはサーバへと送られ、論理式へ変換、議論フレームワークへ変換し、意味論を計算されて、iPad へ返却され、VAF へ反映される。

3.2.1 意見入力

論証ダイアグラムへの意見入力は次の3つの要素からなる。

1. ノード作成
2. ノードに命題を入力
3. 他のノードへリンクを張る

これらを行う様子を図4および図5に示す。図4では右側のパネルから青色の四角角を選び、ドラッグしている様子を示す。これによりノード作成および他のノードへリンクを張る操作が行われる。これは、ドラッグする際、関係の種類を指定し、ドロップしたときに重なっているノードを調べることで関係を追加するノードを指定する。ノードの種類とその意味は表1に示す。

表1: ノードの色と意味

色	意味
青	あるノードの意見をサポートしたいときに用いる。
赤	あるノードの意見に反対意見を出したいときに用いる。
黒	ノードを削除したいときに用いる。

図4の操作を行うとポップアップが発生し、新しい入力を行う画面が発生する。この画面にキーボードで命題を入力することで、ノードに命題を入力する。図5では一連の動作を行った結果、ノードが追加された様子を示す。画面左に指定した関係でノードが追加されたことが分かる。

3.2.2 論証ダイアグラムから論理式へ変換

入力された意見は関係とともに一旦、データベースへ保存される。したがって、論証ダイアグラムから論理式へ変換する際は、データベースから今まで出たすべての意見と関係を読み出し、次のルールにしたがって、表2の形式に変換される。

表2: 各集合と意味

集合	意味
R_s	反論できないルール集合
R_d	反論できるルール集合
K_p	前提の集合
K_n	原理の集合

論証ダイアグラムを組, $AD = A, attacks, supports$ とすると, R_d および K_p は以下のようにして得られる。

- $\forall a, b \in A, (b, a) \in supports$
 $\{a \Leftarrow b\} \in R_d \quad \{b\} \in K_p$
- $\forall a, b \in A, (b, a) \in attacks$
 $\{\neg a \Leftarrow b\} \in R_d \quad \{b\} \in K_p$

3.2.3 論理式から議論フレームワークへの変換

Aspic+の変換にしたがい変換する。例を例1に示す。

例1. 以下のルールセットが与えられたときの議論フレームワークへの変換は図6のようになる。まず, R_d のルールにしたがって, 論証木を作成する。例えば A_1 のトップルール, s_1 を調べる。 p は $p \in K_n$ であり, それ以上調べる必要はない。

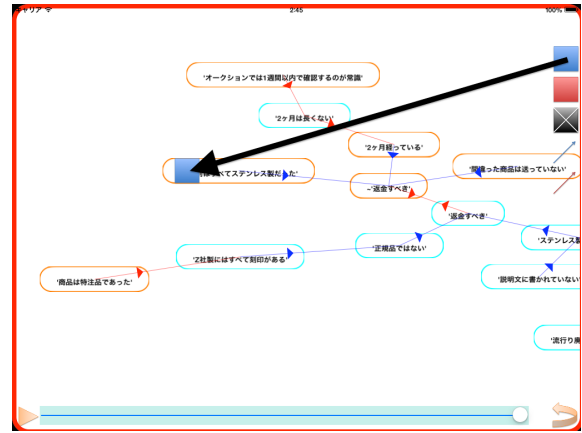


図4: ドラッグで意見を指定

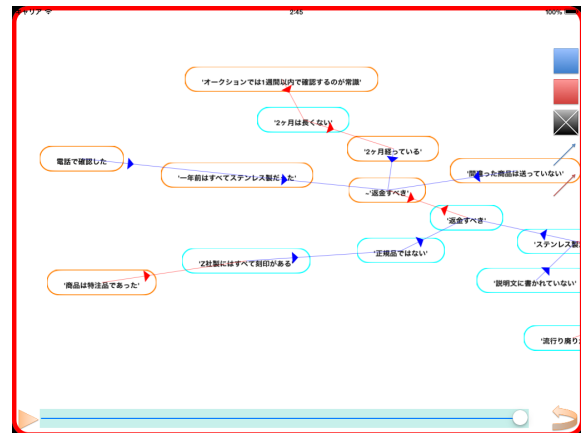


図5: 図の更新

次に q は $Conc(d_1) = q$ であるので, d_1 を調べる。 p は先ほどと同様に $p \in K_n$ であり, それ以上調べる必要はない。結果, 論証 A_1 は図のようになる。同様に $A_1 \sim A_4$ まで作成し, 各論証木の否定記号に着目して攻撃関係を作成する。最後に論証と攻撃関係のみを書き出せば図6右のようになり, 議論フレームワークとなる。

$$\begin{aligned}
 R_s &= \{s_1, s_2\} & R_d &= \{d_1, d_2, d_3, d_4, d_5, d_6\} \\
 K_p &= \{s, u, x\} & K_n &= \{p\} \\
 d_1 : p \Rightarrow q & & d_4 : u \Rightarrow v & & s_1 : p, q \rightarrow r \\
 d_2 : s \Rightarrow t & & d_5 : v, x \Rightarrow \neg t & & s_2 : v \rightarrow \neg s \\
 d_3 : t \Rightarrow \neg d_1 & & d_6 : s \Rightarrow \neg p & &
 \end{aligned}$$

3.2.4 意味論の計算

与えられた VAF について選好拡大 (Preferred Extension) を計算することにより求める。図5の入力を行った結果, 得られた VAF を図7に示す。薄く青い背景となっているノードが現在の時点で正当化される可能性のある論証である。

VAF では各ノード間の攻撃関係はそのまますべて意味論の計算に用いられるわけではなく, 議論に参加しているユーザ全体の価値観によって採用される攻撃関係とそうでないものが存在する。

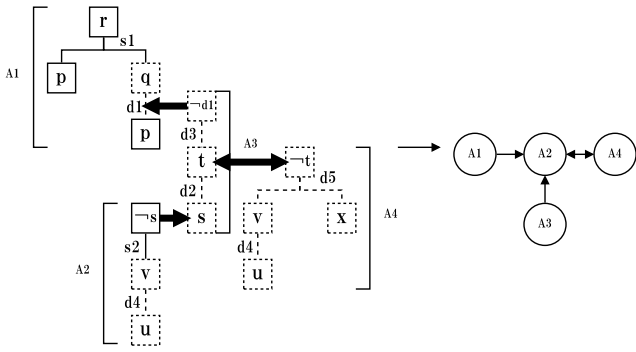


図 6: 論理式から議論フレームワークへの変換例

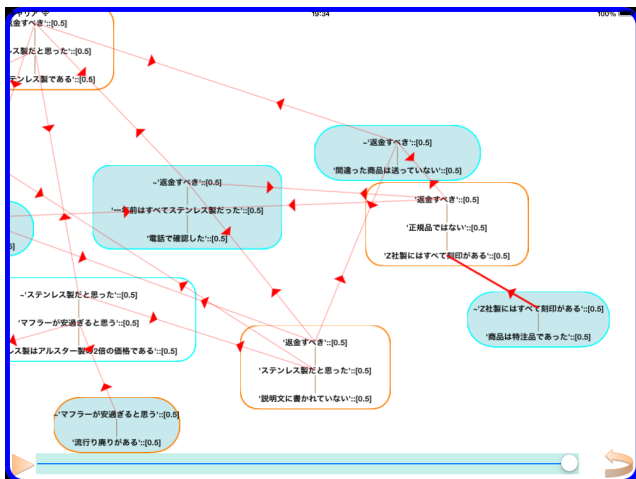


図 7: 議論フレームワークの例

3.3 機能

本ツールの代表的な機能を紹介します。大きく分けて3つの機能があり、攻撃するとよいポイントを助言する助言機能、参加者全体の価値観を反映するための投票機能、そしてユーザが議論を行いやすくするためのインターフェース機能である。

- どの意見を攻撃・サポートすればよいかを表示する。図7の例でいえば、青になっているノードが正当化されており、これらの論証木に攻撃する意見を出せばよいことが分かる。また、線が太くなっているノード同士は優劣関係がはっきりとしており、負ける原因を作っている関係である。この部分を解消することも有効であると分かる。また、図4では赤と青で賛成側・反対側が色分けされている。論証ダイアグラムでは議論の時系列を保存しており、常に中心から外側へ伸びるグラフとなっている。したがって、最も外側にあるノードは攻撃もサポートされていない根拠のない弱い意見となり、この部分への意見は常に有効な意見となる。
- ノードに価値をつけることにより、議論を行っているユーザ全体の価値観を議論結果に反映することができる。1つの論証に対して、1人1回ノードを評価することにより、論証の価値が上下し、議論結果が変化することがある。この機能により人間が価値観に基づいて公平に判断した結果に近づけることができる。
- ノードの中に論証木を表示する。図7に示すように、論証木内の意見に対して攻撃関係をつなげることにより議

論全体を見渡すときに可読性が上昇するほか、サブ論証によるノードの数を減らすことができる。

- 特定のノードを注視することができる。議論が進むにつれて VAF は複雑になり、攻撃関係も絡まっていく。本ツールでは論証をタップすることでそのノードが攻撃している、または攻撃されている関係のみを表示し、議論を理解しやすくしている。
- 自分の発言を議論に出す前に確認してみることができる。議論が複雑になるにつれて、1つの発言が議論の結果に大きく影響する場合が多々ある。そこで本ツールでは、意見を出す前に、意見を出した後の議論フレームワークを仮に計算させ、あらかじめ結果を取得することができる。これにより、自身の意見で議論を不利にすることはなく、戦略を立てやすくなる。

4. おわりに

本研究では、現在まで数理議論学で研究されてきた議論の表現・計算方法を実際の議論へ適用するための検討を行った。その結果、実際の議論で論理式を入力するのは現実的ではなく、論理式とユーザの直感との間を埋める入力方式が必要であると分かり、論証ダイアグラムを提案した。これにより、ユーザは意見を直感的に入力できるほか、議論フレームワークでは表現できなかった議論の時系列を図形式で保存できるようになり、議論の可読性を増すことができた。

今後はするか、しないかという0または1の議論だけではなく、合意形成を行うような議論など、様々な議論を扱えるようにするほか、Web上での実装を行い、論理学の教育面でも使用可能なツールに発展させていきたい。

参考文献

[Toulmin 58] Stephen Edelson Toulmin: The use of Argument, pp.97-103(1958)

[Dung 95] Dung, P. M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games, Artificial Intelligence, Vol. 77, pp. 321 - 357 (1995)

[Bench-Capon 03] T. J. M. Bench-Capon: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation, 13(3):429 - 448, 2003.

[Prakken 11] Prakken, H.: An overview of formal models of argumentation and their application in philosophy. Studies in logic. Vol4, no 1: 65-86, 2011.

[Katsura 14] Katsura Y., Sawamura H., Hagiwara T. and Riche J.: Asynchronous Argumentation with Pervasive Personal Communication Tools, in The International Conference on Agents and Artificial Intelligence (ICAART2014), pp105-114(2014)