

Webブラウザで動作するインターネット会議システムの構築

榎園裕崇 伊藤毅志

電気通信大学 情報工学科

E-Mail: {enokiz-h, ito}@phaeton.cs.uec.ac.jp

概要

WWWの出現によって、広く一般の人々がインターネットを利用するようになってきた。本稿では、一般ユーザが行うネット上での日常的な会議を支援するグループウェアについて議論する。ここでは、「旅行計画の立案」という議題を題材に、まず、参加者に議論進捗状況を認識させるための議論モデルについて考察する。そして、Webブラウザ上で実現したGUIを用いた会議支援を行うシステムについて説明する。

キーワード：会議支援システム、ブラウザ、議論モデル、GUI

Development of Discussion Support System on Web-Browser

Hirotaka ENOKIZONO, Takeshi ITO

Department of Computer Science,
The University of Electro-Communications
E-Mail: {enokiz-h, ito}@phaeton.cs.uec.ac.jp

Abstract

The appearance of the WWW technology has made many general users to use Internet. In this article, we explain the group-ware supporting daily network discussion. We discuss the argument model to recognize the argument process. On the basis of the results, we explain the prototype of the discussion support system, which show the graphical discussion space on WEB-browser.

Keyword : discussion support system, browser, argument model, GUI

1. はじめに

インターネットにおけるWWWの出現によって、広く一般の人々がコンピュータオープンネットワークを利用するようになってきた。現在これらの主な利用目的は、マルチメディア文章の発信と受信に限られているが、WWW技術はグループウェアの基盤技術としても利用できる（インターネット、X-TRAネット）ことから、今後はインターネット上で、日常的な議論や情報交換を支援するグループワークの研究が望まれるようになるだろう。

本研究では、「旅行計画の立案」という議題を題材にして、多人数における会議を支援するシステムの構築を行っている。本システムは文字ベースの非同期型会議システムで、ブラウザ上で動作するJavaアプリケーションと、サーバアプリケーションで構成されている。現在はシステムのプロトタイプで運用実験を行っている状態であるが、最終的にはインターネット上で公開し、広く一般の人々が利用できるシステムにする予定である。

本報告では、まず議論上の問題点について指摘する。そして、その問題点を解消するシステムの設計方針について述べる。最後に、現在実装を進めているシステムについて詳述し、今後の課題について考察する。

2. 議論進行上の問題点と支援

一般的に、会議の議論内容は多重構造を持つ。「旅行計画の立案」も複数の副議題からなり、図1のような階層木構造をなすものと考えられる。このような議題に関する議論は、しばしば迷走・停滞・中断する。

我々は、実際に「旅行計画の立案」を議題に会議の実験を行い、発話データの収集を行った[1]。収集したデータの分析から、議論を迷走・停滞・中断させる原因には、次のものがあることが分かってきた。

(1) 議題構造の認識不足

- 話題（何を話し合えばいいのか）
- 順序（どう話し合えばいいのか）
- 完結性（議論が不完全な話題はないか）

(2) 議論の進捗状況の認識不足

- インタラプト（話題を変え、再びもとの話題に戻ったときの、議論の進捗状況の把握ができるか）

(3) 世界知識の不足

- 情報（会議参加者が、議論を進める上で不可欠な情報を持っていないことがある）

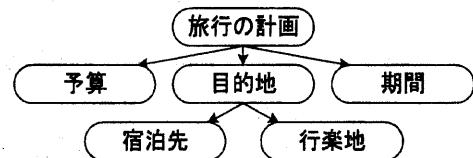


図1:議題「旅行計画の立案」の木構造図

議論進行上の問題点を解決するためには、議論の構造や進捗状況を会議参加者に明示し、会議に必要な情報を適宜提供することが望まれる。

本研究では、「議題の構造」と「議論の進捗状況」をコンピュータグラフィックにより視覚的にユーザーへ提供するシステムを構築する。さらに、観光地情報の検索機能によって、「世界知識」の情報も得られる会議支援システムの開発を目指す。

3. 議論進捗状況のユーザへの提供

システムは、議題別に「議論の状態図」を作成し、ユーザーへ提供する。これにより、ユーザーが議論の進捗状況を視覚的に把握することを期待できるだろう。議論の状態図は、議論空間を記述する「議論モデル」に基づいて作成するものとする。

3.1. 議論モデル

議論空間を記述するためのモデルは、今までにCSCWやグループウェアといった研究領域において様々な提案がなされている[2][3][4]。本研究では、QOCモデルに注目し、それに改良を加えた議論モデルを新たに提案する。

QOC(Questions, Options, & Criteria)は、ソフトウェアの設計過程における議論を記述することを目的に、XeroxのMacLeanらによって提案されているモデルである[5][6]。QOCは図2に示すように、課題を示すQuestionノード、解決案の候補を示すOptionノード、比較評価の基準となるCriteriaノードの3種のノードからなるシンプルなモデルである。

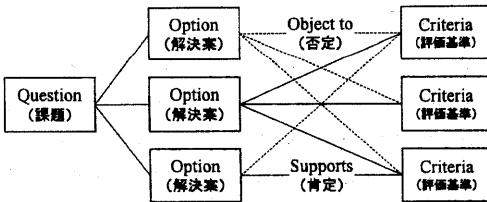


図 2:QOCモデル

QOCは、「議論のプロセスよりも議論の構造を記述するのに適している」「比較評価の基準を明示的に示せる」「他のモデルに比べてシンプルである」といった特長を持っている[7]。これらの特長から、QOCは本研究で使用する議論への適用が考えられるが、以下の点で不十分なモデルである。

- ① 比較評価の基準は各解決案の持つ様々な情報を明示的に表現できない。
- ② 各提案事項に対する会議参加者それぞれの評価を表現できない。また、単に賛成と反対だけではなく、きめ細かな評価を表現できない。
- ③ 提案事項を絞り込むための条件を明示的に表現できない。

以上のような問題点を解消するために、我々は、QOCに次のような改良を加えた（図3）。まず評価基準を示すものとして、Criteriaノードの変わりに、解決案の持つ様々な情報を表現するPropertyノードを準備する。PropertyノードはCriteriaノードとは異なり、単一のOptionノードとのみリンクし、リンクは評価を伴わない。評価はOptionノードが、会議参加者それぞれの評価値を保持する。評価値は5段階で、それぞれ「賛成」「容認」「評価保留」「難色」「反対」である。さらに、解決案を絞り込むための条件を示すConditionノードを加える。Conditionノードは、その条件に適合するPropertyノードとリンクする。

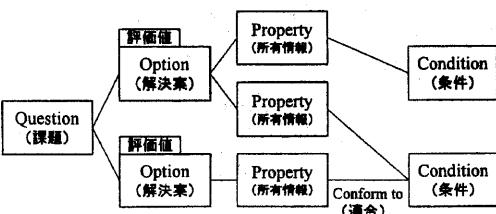


図 3:本システムのモデル

3.2. 基本操作

ノードとリンクの追加や、評価値の変更といった操作は、会議内での発話意図に対応している。本モデルにおける基本操作を、操作の対象となるノード別に表1に示す。

対象ノード	操作
Question （課題）	(a) Optionノードの付加 (b) Conditionノードの付加
Option （解決案）	(c) Propertyノードの付加 (d) Optionの評価値変更
Property （所有情報）	(e) Propertyに基づくOptionの評価値変更
Condition （条件）	(f) Propertyへのリンク付加

表1:モデルにおける基本操作

(d)と(e)はどちらも、対象となるOptionの評価値を変更する操作であるが、評価の理由となるPropertyノードを伴っているかという点で異なる。また、この基本操作の中には、「議題提起」を示すQuestionノードを付加する操作が含まれていない。本モデルでは、議題は最初から固定で存在するものとしている。

3.3. 議論状態図の記述例

議論の状態図を作成した例を図4に示す。これは、A,B,Cの3名で箱根温泉旅行の計画を話し合った場合の発話データから、「宿泊先の選定」部分を抜粋したものである。

発話データには、各発言によってモデルにどのような操作が加わるかを、表1中の基本操作(a)-(f)で示した。例えば、発話データ中に*で示した参加者Bによる発言は、以下の3つの操作に対応している。

- ① 課題「宿泊先(Q)」に対して、解決案「小桶園(O)」を付加 一基本操作(a)
- ② 解決案「小桶園(O)」に対して、所有情報「ケーキがおいしい(P)」を付加 一基本操作(c)
- ③ 所有情報「ケーキがおいしい(P)」は条件「食べ物重視(C)」に適合するので、この二つをリンク 一基本操作(f)

これから分かるように、1回の発言は複数の発話意図を持つことがあり、この場合のモデルにおける操作は、基本操作を複数組み合わせたものになる。

発話データ	
A: やっぱり今度は食べ物重視でいきたいな	(b)
B: 露天風呂も条件だ	(b)
A: 日は山のホテルがいい、食べ物がうまいから	(a) (c) (e) (f)
B: 小桶園のケーキもおいしいんだよねー*	(a) (c) (f)
A: 小桶園は高いよ	(c) (e)
C: 一泊1万5千円くらいが妥当かな?	(b)
B: 小桶園の安いやつなら1万3千円くらいだったよ	(c) (f)
A: 塔ノ沢の旅館がちょうど1万5千円だ	(a) (c) (f)

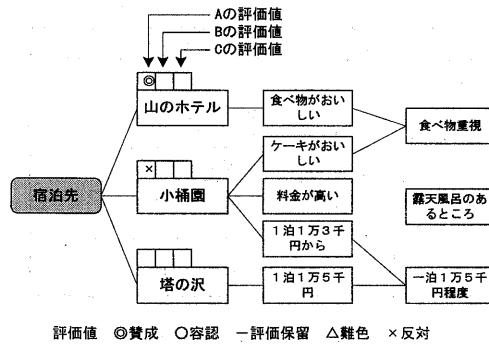


図 4: モデルに基づいた議論の状態図

4. システム

「議論の状態図」をコンピュータグラフィックによってユーザーへ提供する、GUIベースの非同期型会議システムを構築した。

4.1. 概要

本システムはクライアントサーバ型のシステムで、Java/HORB[8]を用いて構築した。サーバはスタンダードアロンのアプリケーションだが、クライアントはJavaアプレットで、Javaが動作するWWWブラウザ上で動作する。

Javaアプレットが実行されると、ユーザ認証を行うダイアログが表示され、ユーザは「ユーザ名」「パスワード」「会議室名」を入力する。入力が完了するとクライアントはサーバへの問い合わせを行う。ユーザ名とパスワードの組み合わせに不正が無く、会議の参加者であることが認められれば、目的の会議室への入室が許可される。

サーバの機能は、大きく「ユーザ管理」と「会議室の管理」に分けられる。サーバ上には複数の会議室を設置することができ、ユーザは複数の会議を掛け持ちすることができる。

システムの実行画面を図5に示す。これは、3人でスキー場の選定を行った場合の例である。画面左が議論の状態図で、右が発話履歴である。発話履歴は、参加者の発言を順に並べたもので、従来の電子掲示板のように、議論の流れを時系列的に閲覧できる。

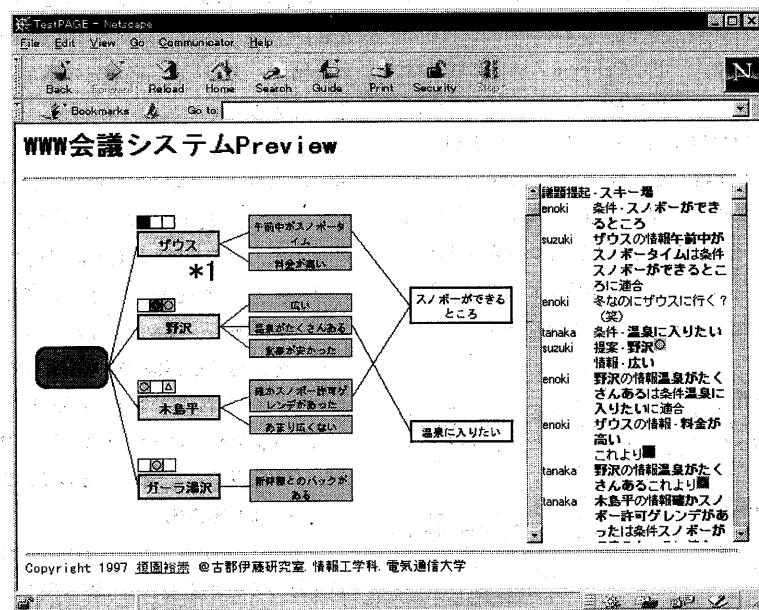


図 5: 実行画面

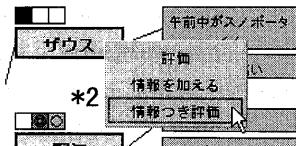


図 6: ポップアップメニュー

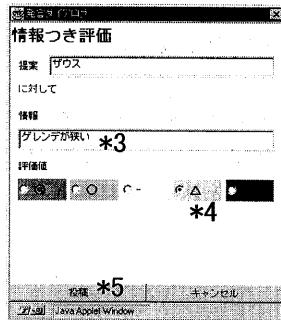


図 7: 発言ダイアログ

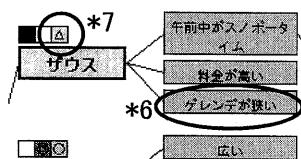


図 8: 発言後の更新された状態図

4.2. 発言方法

ユーザは、議論の状態図に基づいて、GUIによって発言を行い会議を進行させる。発言の手順を以下に示す。

- ① 発言の対象となるノードを一つ選択し、マウスカーソルでクリックする。
- ② 「発言の種類」がポップアップメニューで表示されるので、発言の意図に適合するものを一つ選択する。
- ③ 「発言ダイアログボックス」が表示されるので、必要事項を記入し「投稿ボタン」をクリックする。例えば、図5において3番目のユーザが、「スキー場の候補としてあがっている『ザウス』は、ゲレンデが狭いから少し嫌だ(難色)」という意味の発言をする例を考えてみよう。

まず、Optionノード「ザウス」(*1)をマウスでクリックする。ポップアップメニューが表示されるので

(図6)、「情報つき評価」を選択する(*2)。次に、「情報つき評価」を行うための「発言ダイアログボックス」が表示されるので(図7)、「ザウス」に対する情報として「ゲレンデが狭い」を入力し(*3)、評価の「△難色」を選択する(*4)。「投稿ボタン」(*5)をクリックすると、議論の状態図が更新される(図8)。Propertyノード「ゲレンデが狭い」が付加され(*6)、「ザウス」に対する3番目のユーザの評価値が変更される(*7)。

4.3. 発言の種類

ポップアップメニューで選択できる発言の種類は、選択したノードによって異なる。ノードと発言の種類の対応を表2に示す。表内の太文字ゴシックで表記したキーワードは、実際にシステムのユーザインタフェースで使用しているものである。

発言の種類のそれぞれは、実際の会議における1回の発言のメタファとなるように、表1で示した基本操作を複数組み合わせた複合操作になっている。これを表2では、①-⑨のそれぞれに対して、表1の基本操作(a)-(f)で示した。

対象ノード	発言の種類
Question 議題	① 提案 (a)+(d)
	② 情報つき提案 (a)+(c)+(e)
	③ 条件を加える (b)
Option 提案	④ 評価 (d)
	⑤ 情報を加える (c)
	⑥ 情報つき評価 (c)+(e)
Property 情報	⑦ これをもとに評価 (e)
	⑧ 既存の情報を指摘 (f)
Condition 条件	⑨ 新規の情報を指摘 ((a)+(c)+(f))
ノード外の余白	⑩ その他の会話

表2: ユーザインタフェースとして準備した発言の種類

表2の①-⑨のいずれかを用いれば、会議内の全ての発言をカバーすることができると考えている。例えば、図4で示した発話データの3番目の発話などは、②+⑧もしくは⑨+⑦の組み合わせで発言することができる。

この基本操作の数や種類についての妥当性は、実際本システムを使わせてみて、検証を行っていく必要があるだろう。複合操作の組み合わせを増やせば、それだけ一度多くの発言方法をカバーできるが、発言

の種類をいたずらに増やせば、ユーザは自分の発言したい内容を多くの発言種類の中から探さなくてはならず、ユーザに負担をかける結果となる。

さらに、本研究で提案している議論モデルでは表現できない発言も存在する。これらの発言を行うために、⑩の「その他の会話」を準備した。この方法で発言すると、議論の状態図は変化せずに、発話履歴に発言内容がそのまま記録される。

5. おわりに

本稿で紹介したシステムのプロトタイプは、現在研究室内の学生と、インターネット上の有志によって運用実験を行っている。今までにシステムを試用してもらった人達からは、「Webブラウザから利用できるので、空いた時間に簡単に会議を進めることができ」「会議の内容を覚えていなくても、状態図から即座に議論の進行状況を把握できる」といった意見が得られ、本システムの有用性を確認することができた。

今後は、さらに多くの人達に運用実験に参加してもらい、システムの検証を行う予定である。

参考文献

- [1] 榎園裕崇, 伊藤毅志, 古郡廷治: “文字ベースのネットワーク会議を支援するシステムの構築”, 情報処理学会 第53回全国大会論文集-4 pp85-86 (1996).
- [2] Conklin, J., et al.: “gIBIS : A Hypertext Tool for Exploratory Policy Discussion”, Proc. Of CSCW pp140-152 (1988).
- [3] 黒須正明, 鹿志村香: “対話プロセスの記述モデル(EDCモデル)の提案”, 情報処理学会 マルチメディア通信と分散処理研究報告63-4 グループウェア研究報告5-4 pp25-31 (1994).
- [4] 炭野重雄, et al.: “議論推移モデルを用いた知的遠隔会議支援に関する検討”: 情報処理学会 グループウェア研究報告11-3 pp13-18 (1995).
- [5] MacLean, A., Young, R., Bellotti, V. & Moran, T.: “Questions, options, and criteria : Elements of design space analysis”, Human-Computer Interaction, 6(3&4), pp.201-250 (1991).
- [6] MacLean, A., Bellotti, V., & Shum, S.: “Developing the Design Space with Design Space Analysis”, In Computers, communication and usability: Design issues, research and methods for integrated services. Amsterdam: Elsevier North Holland Series in Telecommunication pp.197-219 (1993).
- [7] 桑名栄二: “ソフトウェア履歴利用の研究動向”, 電子情報通信学会誌 Vol.77 No.5 pp.531-538 (1994).
- [8] 平野聰: “新しいネットワークプログラミングパラダイムHORB”, 電総研つくばソフトウェアシンポジウム予稿集 pp87-90 (1996).