

Title	電子メールを用いたコミュニケーションに対する支援環境の設計と実現
Author(s)	山見, 太郎
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1467
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 修士

修 士 論 文

電子メールを用いたコミュニケーションに対する 支援環境の設計と実現

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

山見 太郎

平成十三年三月

修 士 論 文

電子メールを用いたコミュニケーションに対する 支援環境の設計と実現

指導教官 落水 浩一郎 教授

審査委員主査 落水 浩一郎 教授

審査委員 篠田 陽一 助教授

審査委員 島津 明 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

学生番号: 910125

山見 太郎

2001 年 2 月

要 旨

本稿では、電子メールを用いたコミュニケーションにおける会話のモデルである討議構造木モデルに基づく討議構造を表現するための XML ボキャブラリを設計し、討議の道筋として討議構造木を抽出する討議構造抽出エンジンと討議構造木の参照機能をもつメールクライアントからなる支援環境を設計し、実現する。

目次

第1章	はじめに	1
1.1	電子メールを用いたコミュニケーションの特徴	1
1.2	設計方針	2
1.3	構成	3
第2章	電子メールを用いたコミュニケーションにおける討議構造	4
2.1	電子メールを用いた会話のモデルの必要性	4
2.2	討議スレッド	4
2.3	討議構造木	6
第3章	XML による討議構造木の表現: UMML+Linkbase	13
3.1	討議構造データの交換方法	13
3.2	XLink: XML 文書のリンク機能に関する規約	14
3.3	UMML	16
3.3.1	要素: umml	16
3.3.2	要素: head	16
3.3.3	要素: body	18
3.4	Linkbase	19
3.4.1	要素: linkbase	19
3.4.2	要素: link	20
3.5	UMML+Linkbase によるマークアップ例	21
第4章	討議構造木の自動抽出: 討議構造木抽出エンジン	24
4.1	自動抽出の必要性	24
4.2	電子メールを用いたコミュニケーションにおける言語的特徴	24
4.2.1	電子メールボディ部の特徴	24

4.2.2	文章表現にあらわれる特徴	25
4.3	抽出アルゴリズム	27
4.3.1	概略	27
4.3.2	手順 1 . 手がかり語辞書とメンバーリストの読み込み	29
4.3.3	手順 2 . ボディ部から発話候補文章部を抽出	29
4.3.4	手順 3 . 発話候補文章部から発話を抽出	30
第 5 章	討議構造参照機能を有するメールクライアント: ICEMail++	44
5.1	構成	44
5.2	発話抽出結果の提示	44
5.3	討議構造木の呈示	44
第 6 章	おわりに	47
6.1	まとめ	47
6.2	今後の課題	47
6.2.1	運用実験による評価	47
6.2.2	参加者によるアノテーション	47
6.2.3	MIME マルチパートメッセージへの対応	48
謝辞		49
参考文献		50
本研究に関する発表論文		52
付 録 A	UMML.dtd	53
付 録 B	Linkbase.dtd	55

目 次

1.1	支援環境 (概略)	3
2.1	討議ストリームによる討議の構造	6
2.2	コントリビューションツリーによるサイドシーケンスの表現	9
2.3	討議構造木	11
2.4	討議構造木による討議構造	11
3.1	要素: umml	16
3.2	要素: header	16
3.3	要素: From	17
3.4	要素: Subject	17
3.5	要素: Date	17
3.6	要素: linkbaseRef	18
3.7	要素: body	18
3.8	要素: utterance	19
3.9	要素: quote	19
3.10	要素: linkbase	19
3.11	要素: link	20
3.12	要素: locator	20
3.13	要素: arc	20
3.14	Linkbase の例	21
3.15	討議構造木	21
3.16	UMML によるマークアップ例	22
3.17	メール B	23
4.1	手がかり語辞書の例	28

4.2	ボディ部と発話候補文章部の関係	30
4.3	手順 3.1.1	31
4.4	手順 3.4.1	32
4.5	手順 3.5.1	33
4.6	手順 3.5.2.1 , および , 手順 3.5.2.2 (前半)	35
4.7	手順 3.5.2.1 , および , 手順 3.5.2.2 (後半)	36
4.8	手順 3.5.2.3.B.B	38
4.9	手順 3.B	40
4.10	手順 3.D	41
4.11	手順 3.D.1 (前半)	42
4.12	手順 3.D.1 (後半)	43
5.1	ICEMail++	45
5.2	解析結果の呈示	45
5.3	討議構造木の呈示	46

表 目 次

4.1	言語的手がかりの一覧	25
4.2	返答発話の状態と手がかり語との関係	35

第 1 章

はじめに

1.1 電子メールを用いたコミュニケーションの特徴

地理的に分散する人々が共同で作業を進める場合，電子メールを利用した非同期コミュニケーションが良く利用される．電子メールを用いたコミュニケーションは，テキストを介したコミュニケーションである．そして，メッセージとして記述するテキストに決まったフォーマットはない．したがって，送信者の表現を制限することがなく，送信者は気軽にメッセージを記述することができる．また，電子メールを用いたコミュニケーションは，時間の面においても場所の面においても参加者を拘束しない，非同期コミュニケーションでもある．メッセージを送る側は送りたいときにメッセージを相手の元に届けることができ，受け取る側は受け取りたいときに受けとることができる．しかも，電子メールを利用するに当たっては，特別な装置やソフトウェアを必要としない．結果，電子メールはネットワーク上における基本的かつ重要なコミュニケーションツールの 1 つとして大きな役割を担っている．

しかし，長所だけではない．フォーマットのないテキストをやり取りすることによって成立するコミュニケーションであるため，計算機を介した 2 次利用が難しい．また，非同期なコミュニケーションであるため，沈黙などのような対面の場合などにおいて生じる手がかかりを元にした話者交代を行うことができない．そのかわりに，送信者は複数の話題を同一メール中に記述する傾向にある．そのため，複数の話題が並列に議論されやすく，議論が終結しているか否か，返答を要求されているか否かといった現在の状況を見失いやすい．また，テキストのみを介した非同期コミュニケーションであるため，聞き手が話し手の発話を理解しているかどうかを知るための手がかかりも少ないと言える．このため，対話者間の会話に関する共通理解が阻害され，認識の不一致が発生しやすい．

1.2 設計方針

本研究の目的は，1.1 節において述べた問題に対して，ある話題に対する討議の流れやいくつかの討議の流れの間の相互関係（討議の全体像）などの討議の構造を呈示するためのモデルとツールを開発することにある．電子メールを用いたコミュニケーションにおける長所を殺さずに短所を改良するため，以下の方針に従う支援環境を設計し，実現する．

方針 1．既存の環境（本稿に述べる支援環境が存在しない環境）との連帯を重視する．

電子メールを利用するすべての人が，本稿に述べる支援環境を利用できる環境にあるわけではない．STB(セットトップボックス)，PDA，携帯電話など制約の厳しい環境における電子メールの利用を考慮にいとると，本稿に述べる支援環境が利用できない状況は将来においてもありうると考えられる．したがって，既存の環境との連帯は無視できず，少なくとも，既存の環境からのメッセージ送受信を妨げてはならない．

方針 2．参加者に対し，既存の環境における負荷以上の負荷を与えない．

電子メールは，非同期型グループウェアに関する研究のベースとして古くから用いられている．それらの研究において開発された環境に用いられている手法の中には，決められたキーワードとフォーマットの指定のない値からなる半構造メッセージを導入する手法 [1] [2] [3] [4] や，参加者によって明示された発話意図を利用する手法 [5] [6] [7] [8] などがある．

半構造メッセージには以下の利点がある．

- 構造を解析する必要がない．
- 値にはフォーマットの指定がないので，例外的なことも記述できる．
- 伝票などといった，半構造に基づいた文書のやり取りをそのまま表現できる．
- 記入漏れのチェックが容易である．

しかしながら，この手法は討議における発話のような構造がまったく固まっていないメッセージにたいして利用することが難しいと思われる．また，発話意図を明示させる手法に関しては，発話意図の入力が参加者にとって心理的障壁が高いことなどから，発話意図を明示させるという考え方に対して批判的な意見もある [9] ．

したがって，設計・実現する支援環境においては，参加者に対して表現上の制限を加えることもしないし，発話意図のようなメタ情報の入力を強要することもしない．

方針 3．支援環境で取り扱う文章の最小単位を電子メール一通とせず，より細分化する．

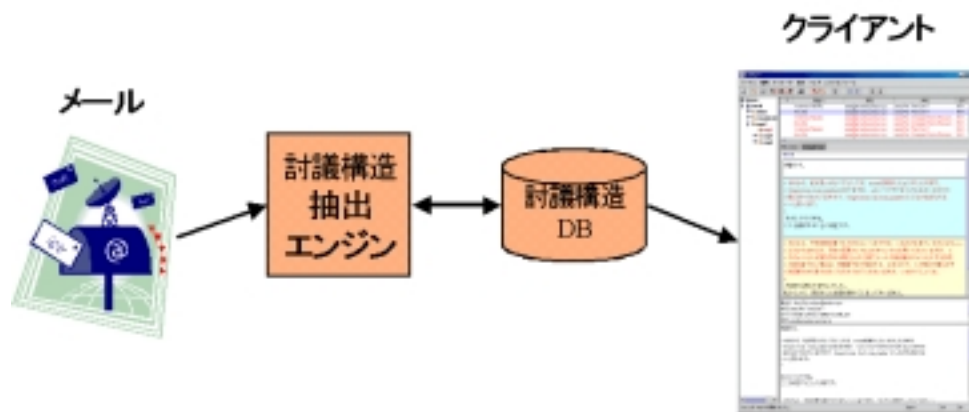


図 1.1: 支援環境 (概略)

これは、同一電子メール中に話題の異なる複数の文章を記述する傾向にあることをふまえるということを意味する。どこまで細分化し、どのように細分化するのかについては、次章にて述べる。

方針 4. 討議の道筋を容易に参照できるようにする。

過去の発話を話題毎に分類、整理することによって、討議の道筋を参照することができれば、例えば、話題の終結に関する参加者間の認識に不一致や同じ話題の議論の繰り返しを回避できるという効果があると予想される [10]。

1.3 構成

実現する支援環境の概略を図 1.1 に示す。コミュニケーションを構成する電子メールは、まず討議構造抽出エンジンに送られる。送られた電子メールは、討議構造抽出エンジンによって発話に分割され、これまでに構築した討議構造に追加される。コミュニケーションの参加者は、クライアントを通じて蓄積された討議構造である討議の道筋を参照することができる。

次章以降、まず、討議構造抽出エンジンにおいて抽出すべき討議構造自体について述べ (第 2 章)、環境に依存しない XML を用いた討議構造の表現方法について述べる (第 3 章)。次に、討議構造抽出エンジンがどのようにして討議構造の抽出を行うのかについて述べ (第 4 章)、クライアントのプロトタイプ実装である ICEMail++ について述べる (第 5 章)。

第 2 章

電子メールを用いたコミュニケーションにおける討議構造

2.1 電子メールを用いた会話のモデルの必要性

対面での会話と、電子メールを用いた会話では、いくつか異なる点が存在する。例えば、対面での会話の場合、話者交代ごとに 1 つの発話が生成されると捉えることができるが、電子メールを利用したコミュニケーションの場合、対面での話者交代とは異なるため、発話を話者交代によって定義することができない。したがって、電子メールを利用したコミュニケーションには、対面の会話の場合と異なる会話モデルを用いる必要がある。この章では、電子メールを利用したコミュニケーションにおける会話モデルとして、2 つのモデルを取り上げる。

2.2 討議スレッド

討議スレッドは、発話の接続ペアによる会話の典型的な連鎖の考え方を利用した、電子メールを用いた 1 対 1 の会話モデルである [11][12]。

会話は、会話が行われている時に生じる話し手と聞き手の役割の交代(話者交代)によって特徴づけることができる。会話分析の分野において、話者交代を支配する規則についての研究が行われている。発話の接続ペアは、そのうちの 1 つである。発話は、“提案”，“了承”，“質問”，“回答”，“挨拶”，“申し出”，“受容”といった役割を持っている。これらの発話が，“提案－了承”，“質問－回答”，“挨拶－挨拶”，“申し出－受容”といった典型的

な連鎖を構成する。発話の隣接ペアは、会話を構成するための基本単位であると言える。以下に、発話の隣接ペアの概念をまとめる。

- 発話の隣接ペアは、次のような2つの発話の連続のことである。
 - － 隣り合っている。
 - － それぞれの発話は、異なった話者によるものである。
 - － 第1部分と第2部分という順序がある。
 - － 第1部分の役割が決まった役割をもつ第2部分を要求する。例えば、第1部分の役割が“質問”ならば、第2部分として“回答”の役割を持つ発話を要求する。
- ある隣接ペアの第1部分の後で現在の話者が話しをやめたなら、その時点で次の話者が第2部分を発ししなければならない。

しかしながら、先に述べたように、対面での会話の場合では話者交代ごとに1つの発話が生成されると捉えることができるが、電子メールを利用したコミュニケーションの場合、非同期型のコミュニケーションであるため、発話を話者交代によって定義することができない。したがって、電子メールを用いたコミュニケーションにおける発話を定義する必要がある。同一メール内に複数の話題が存在しうることが、電子メールを利用したコミュニケーションの特徴の一つである。そこで、発話を、同一メール内で同じ話題に言及している文章と定義する。ただし、発話を構成する文章間には、同一話題という整合性(コヒーレンス)が維持されるものとする。発話は、以下の三種類に分類することができるものとする。

- 相手からの返答を求めず、議論の進行に直接の影響を与えない発話(以降、宣言的な発話)
- 相手からの返答を要求する発話(以降、返答を要求する発話)
- 相手からの返答を求めず、話題を終結する発話(以降、話題を終結する発話)

電子メールを利用したコミュニケーションの場合は、対面の会話とは異なり、隣接した発話同士による発話ペアの構築は意味を持たない。そこで、発話ペアを構築する発話を隣接した発話同士に限定しないこととする。その結果、発話ペアは次のような関係を持った発話によって構成される。

- 会話の典型的な連鎖となる発話。
- ある発話とその発話に対する挿入連鎖の開始となる発話。
- ある発話とその発話に基づく発話。

以上の関係から、発話ペアの先行発話は返答を要求する発話であり、後続発話は返答を要

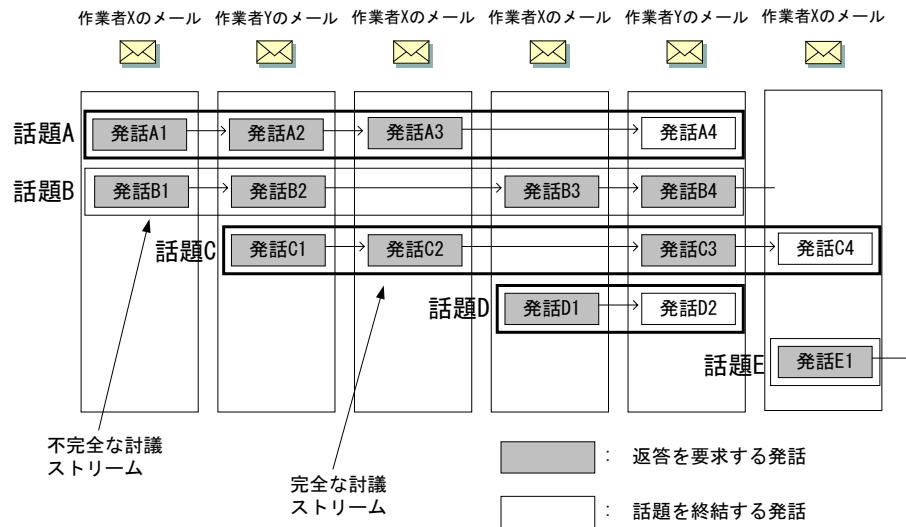


図 2.1: 討議ストリームによる討議の構造

求する発話か話題を終結する発話かのどちらかであると言える．そこで，発話ペアを基本とした，同一話題に言及している 1 つ以上の返答を要求する発話と話題を終結する発話による連鎖を，討議ストリームと定義する．討議ストリームの最初の発話は，返答を要求する発話である．また，返答を要求する発話に続く発話は，返答を要求する発話か，話題を終結する発話である．討議ストリームが話題を終結する発話で終結している場合，そのストリームを完全な討議ストリームと呼ぶ．反対に，話題を終結する発話で終結していない討議ストリームを，不完全な討議ストリームと呼ぶ．

以上の定義によって構成される討議の構造の例を図 2.1 に示す．図 2.1 では，メールを縦長の長方形で，メールに含まれる発話を小さい長方形で表現している．また討議ストリームを，水平のラインでリンクされている発話の列で表現している．

2.3 討議構造木

討議構造木は，対面による 2 者の会話モデルであるコントリビューションツリー [13] を拡張した，電子メールを用いた複数話者の会話モデルである [14][15] ．

コントリビューションツリーとは，Clark により提案された対面による 2 者の会話モデルである．コントリビューションとは，話し手と聞き手による会話のユニットのことであ

る．コントリビューション (C) は，プレゼンテーションフェーズ (Pr) とアクセプタンスフェーズ (Ac) から形成される．会話参加者がコントリビューションを形成するためには，話し手の発話の意図を聞き手が理解しているかどうかの事実が会話参加者の共通認識として話し手が獲得する必要がある．

Pr 話し手が，聞き手に対し，自分の意思を伝達するための発話 u を行うフェーズ．

Ac 聞き手が，話し手の発話 u を理解したか否かを示すために，話し手が理解できると思われる行動 e で相手に示し，発話 u を受け入れるフェーズ．

コントリビューションツリーとは，会話参加者によるコントリビューションがツリー状に構築されたものである．このツリーには，以下の特徴がある．

- すべての発話は，ツリーを構成するコントリビューションに属するプレゼンテーションフェーズに対応づけることができる．
- ほとんどのコントリビューションは，同一レベルのコントリビューションの開始によって，完結する．
- 会話参加者 X による最初の発話内容が会話参加者 Y に理解されている場合， Y による発話はアクセプタンスフェーズであるとともに，次のコントリビューションのプレゼンテーションフェーズとなる．

討議構造木とコントリビューションツリーでは以下の点が異なる．

発話の定義

先にも述べたが，電子メールを利用した会話には，対面での会話と同じ発話の定義を用いることができない．代わりに，討議構造木では，発話の定義として，討議スレッドにおける発話の定義を拡張したものをを用いる．

発話 同一メール内で同じ話題に言及している文章．話題を構成する文章間には，同一話題という整合性 (コヒーレンス) が維持されるものとする．ただし，以下の場合においては，前半部分と後半部分に分割する．

パターン 1 後半部分において前半部分の詳細に言及しているもの．

パターン 2 前半部分に述べている条件に後半部分の内容が依存するもの．

フェーズの定義

コントリビューションツリーに基づく複数人による対面の会話のモデルに，Novic によるコントリビューショングラフがある [16]．討議構造木では，コントリビューショング

ラフにおけるプライマリエビデンスとセカンダリエビデンスの概念を取り入れる．プライマリエビデンスとセカンダリエビデンスの定義を示す．

プライマリエビデンス 聞き手のうちの一人 Y_i が自分こそ話し手 X の意図する聞き手だと信じるときに Y_i が示す行動 e ．言い換えるならば， u を X と Y_i が互いに理解したということを示す根拠を X が求めていると Y_i が信じている時に， Y_i が示す行動 e ．

セカンダリエビデンス 聞き手のうちの一人 Y_j が起こす行動 e' ．ただし，プライマリエビデンスと異なり， Y_j が X の意図する聞き手 Y_i でない場合，もしくは， X はプライマリエビデンスや共通理解を求めているないと Y_j が判断した場合の行動を指す．

プライマリエビデンスとセカンダリエビデンスの概念を取り入れるため，コントリビューションツリーにおけるプレゼンテーションフェーズとアクセプタンスフェーズを再定義し，新しいフェーズ，インダイレクトアクセプタンスフェーズ (*InAc*) を導入する．討議構造木における各フェーズの定義は次の通りとする．なお，定義中， Y_i は，話し手 X 以外の聞き手とし， n はその聞き手の人数とする¹．

Pr 話し手 X が聞き手の一部 $\mathbb{Y}$ を対象にその話題を継続するか否かの意思を伝達するために発話 u を行うフェーズ．この時， X は次の仮定をおいている．

$\mathbb{Y}$ のうちの誰か Y_i がプライマリエビデンス e を示したならば， u を発したときの X の意思を Y_i が理解していると X が信じることができる．

Ac Y_i が発話 u を受理したことを， u で示された X の意思を理解したことを意味するプライマリエビデンス e によって示すフェーズ．この時， Y_i は次の仮定をおいている．

プライマリエビデンス e を示したならば，「 u を発したときの X の意思を Y_i が理解していると X が信じる」と Y_i が信じることができる．

InAc *Ac* と異なり，プライマリエビデンス e ではなく，セカンダリエビデンス e' によって示すフェーズ．

¹したがって， $1 \leq i \leq n$ である．

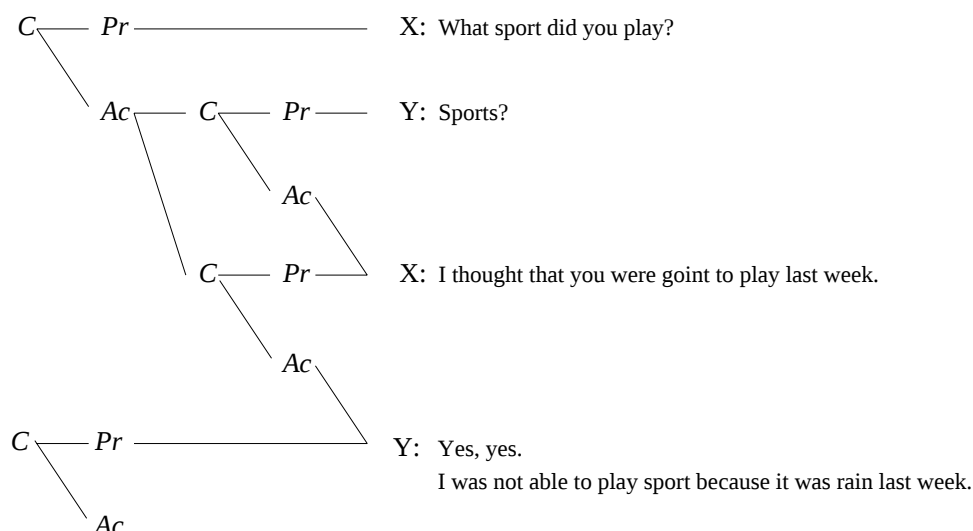


図 2.2: コントリビューションツリーによるサイドシーケンスの表現

サフィックスの有無

隣接ペアにおいて、会話の相手が第一部分を受け入れることができなかった場合、通常、その相手は第一部分を受け入れるための修復シーケンスを導入する。この状況をコントリビューションツリーにて表現したものが図 2.2 である。図 2.2 において、参加者 Y は、参加者 X が “What sport did you play?” という発話によって何に言及しているのか明確でないと思ったため、修復シーケンスを導入している。X と Y は 2 ターンにわたって、何に言及しているのかを明確にしている。この会話のプロセスをサイドシーケンスと呼ぶ。

このサイドシーケンスを含む、電子メールを利用したコミュニケーションにおいて生じる複数の討議の流れを表現するため、討議構造木では C , Pr , Ac および $InAc$ にサフィックスを導入している。サフィックスが $x.y.z...$ であるならば、 $C_{x.y.z...}$ と表記する。サフィックスの第 1 ラベル (x) をサブジェクト番号と呼ぶ。サブジェクト番号とは、サブジェクト毎に割り当てた番号のことで、導入された順番でナンバリングする。第 2 ラベル以降 ($y,z,...$) は、そのラベルより上位のサフィックスの下でコントリビューションが導入された順番を示す。さらに、第 3 ラベル以降については、次のルールに従う。

- ある発話が修復シーケンスを導入したならば、その発話の Pr に、新しいラベルを追加する。
- ある発話が修復シーケンスを終了したならば、その発話の Pr から、この修復シーケンスを導入したときに追加したラベルを除去する。

属性の有無

ソフトウェア工学関連に関するシンポジウム開催のための事務局用メーリングリスト，および，プログラム委員会用のメーリングリストで行われた討議を分析した結果，電子メールを用いた複数話者によるコミュニケーションにおける話者交代に関して，以下に挙げる特徴があった [17] ．

- 次に返答すべき参加者を明示的に示唆することがある．
- 返答を要求されていない参加者が返答を行うことがある．
- 1 つの発話に関して複数の参加者が返答を行い，討議の流れが分岐する場合がある．

これらの複数話者による電子メールを用いたコミュニケーションの特徴を明確に反映するため， C ， Pr ， Ac ， $InAc$ に対して以下に挙げる属性を導入する．

C (トピック番号)

トピック番号 導入された話題毎に割り当てた番号．サブジェクト番号と区別するため，番号の前に“ T ”を付加する (例: $T5$) ．

$Pr(WhoPr, WhomPr, PrS)$

$WhoPr$ 話し手 X を示す．

$WhomPr$ 話し手 X が意図する聞き手集合 $\mathbb{Y}$ を示す．例えば，参加者 A ， B ， C が対象ならば， $A/B/C$ と表記する．

PrS Pr に属する発話の状態を示す．

R 返答を要求しており，話題は継続している．

F 返答を要求しておらず，話題は終結している．

$Ac(WhoAc)$ および $InAc(WhoInAc)$

$WhoAc$ および $WhoInAc$ Ac または $InAc$ に属する話し手を示す．言い換えるならば， Pr に対して何らかの根拠を示した聞き手 Y_i を示す．

以上の定義による討議構造の例を図 2.3 と図 2.4 に示す．図 2.4 は図 2.3 と電子メールとの関係を図示したものである．縦長の長方形が電子メールを，小さい長方形が発話を，二つの発話を結ぶ線がコントリビューションを表現している．

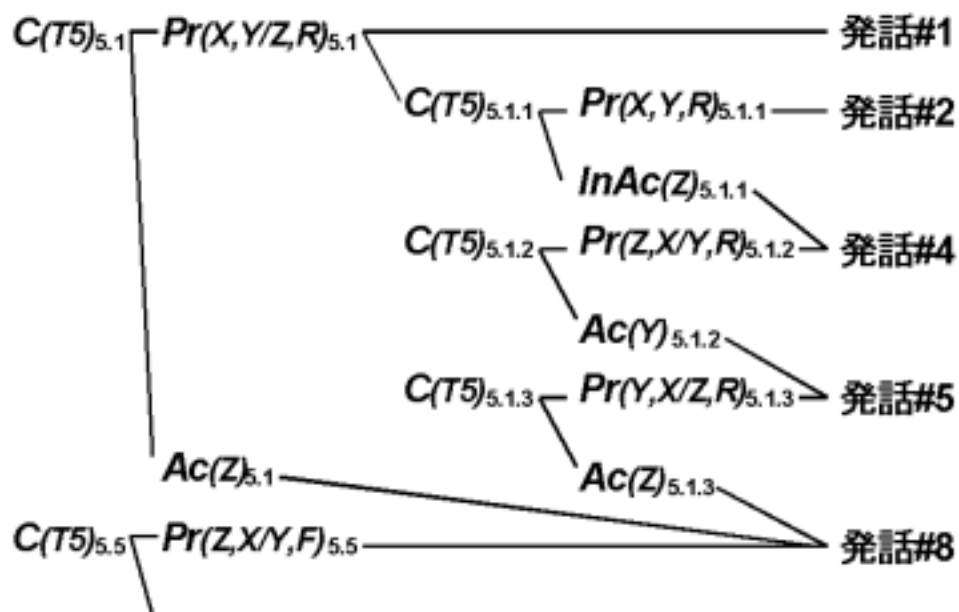


図 2.3: 討議構造木

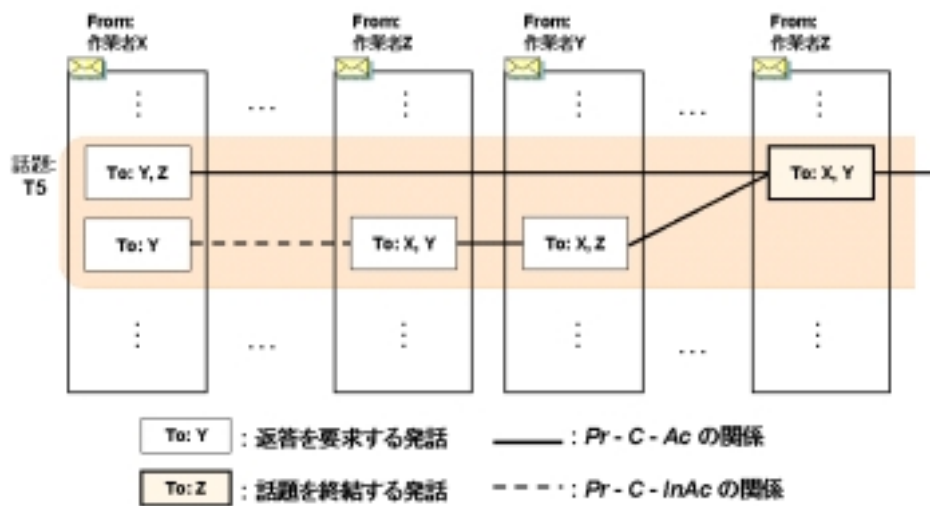


図 2.4: 討議構造木による討議構造

これらを支援環境における討議構造モデルとすれば，1.2 節で述べた方針 3．支援環境で取り扱う文章の最小単位を電子メール一通とせずより細分化するを満たすことになり，討議スレッド，もしくは，討議構造木を方針 4．討議の道筋を容易に参照することができるにおける討議の道筋とすることができる．しかし，討議スレッドは 1 対 1 の会話のモデルであるため，適用範囲が狭い．一方，討議構造木は，複数人による会話のモデルであるため，適用範囲も広い．また，討議スレッドと比べ，討議構造木は，発話の定義の拡張やプライマリエビデンスとセカンダリエビデンスの概念の導入，サフィックスによるサイドシーケンスの表現の導入などにより，討議をより正確に捉えていると言える．したがって，本稿において述べる支援環境においては，討議構造木を討議構造として採用する．

次章では，討議構造木に基づく討議構造を電子メールにマークアップするための XML ボキャブラリである UMML+Linkbase について述べる．

第 3 章

XML による討議構造木の表現: UMML+Linkbase

3.1 討議構造データの交換方法

抽出エンジンとクライアントの設計と実装を行うにあたり、それらのあいだのデータ交換をどう実現するかを考える必要がある。その方法には大きく次の 2 つが考えられる。

方法 1 何らかのデータベース管理システムを介する方法

方法 2 データベース管理システムを介さず、いくつかのファイルを介する方法

一般に、何らかのデータベース管理システムを介する場合、整合性の維持やトランザクション管理、障害回復やスケーラビリティの確保などの面におけるメリットが期待できる。しかし、通信プロトコルなど基本的に通信相手を限ることがおおく、データの可視性に乏しい。方法 1 におけるデメリットは、1.2 節で述べた方針 1。既存の環境との連帯を重視するにそぐわない。また、現時点ではデータベース管理システムを導入しなければならないほど厳密な管理を必要としていない。したがって、方法 2 を選択する。

次に、そのファイルのフォーマットを決定する必要がある。選択肢としては次のものが考えられる。

方法 2.1 独自に定義する方法

方法 2.2 XML に基づく方法

方法 2.1 は設計を行う側にとって自由度の高い方法であるが、支援環境に関わるすべての要素が、そのフォーマットを理解できなければならない方法である。この制約は、方法 1 と同様、1.2 節で述べた方針 1。既存の環境との連帯を重視するにそぐわない。方法 2.2 で

あれば、人間が読むこともでき¹、ソフトウェアからの利用も容易²である。また、テキストに対してメタ情報を付加する方法として自然である。したがって、方法 2.2 を選択する。

3.2 XLink: XML 文書のリンク機能に関する規約

UMML+Linkbase では、XML 文書のリンク機能に関する規約である、XLink[18] を利用している。XLink により、XML 文書において次のことが可能となる。

- 2 つ以上のリソースの間のリンク関係を示す
- リンクとメタデータを関連づける
- リンクされたリソースとは別の場所で、そのリンクを表現できる

まず、本節以降用いる XLink 関連の用語を定義する。

リソース サービスや情報のユニット。ファイルや画像、文書やプログラム、検索結果など。

リンク リソースやリソースを示すポインタの間の明示された関係。

リンク要素 XLink 規約を満たす要素で、リンクの存在を明示する要素。XLink は XML 文書のリンク機能を規定するための名前空間と属性と制約を与えるものであり、具体的な要素を提供するわけではない。具体的には、次の 2 つを満たすものをリンク要素という。

1. XLink 名前空間に属する属性 type を持ち、その値が、“simple”、“extended”、“locator”、“arc”、“resource”、“title”、または“none”のうちのどれかである。
2. 属性 type によって選択した XLink 要素の種類によって決まる適用制約を順守している。

巡回 どのような目的においても、リンクをたどること。

スタートリソース 巡回の始点となるリソース。

エンドリソース 巡回の終点となるリソース。

アーク 2 つのリソース間の巡回方法に関するルール。巡回の方向や適用可能な振る舞いに関する情報などを含む。

ローカルリソース リンクに参加している XML 要素のうち、そのリソースが、

¹テキストベースであるため直接読むこともでき、汎用の XML ビューアを介して読むこともできる。

²パーサなどの実装が数多く存在する。

そのリンク要素の親，またはリンク要素である自分自身である場合，そのリソースをローカルリソースと呼ぶ．

リモートリソース たとえそのリンクと同じ XML 文書にあったとしても，URI によって指し示されているリソースを，リモートリソースと呼ぶ．

アウトバウンド あるアークがローカルリソースをスタートリソースとして持ち，リモートリソースをエンドリソースとして持つとき，「このアークはアウトバウンドアークである」という．HTML における A タグが示すアークも，アウトバウンドアークある．

インバウンド あるアークがリモートリソースをスタートリソースとして持ち，ローカルリソースをエンドリソースとして持つとき，「このアークはインバウンドアークである」という．

サードパーティ あるアークのスタートリソースとエンドリソースがどちらもリモートリソースであるとき，「このアークはサードパーティアークである」という．

リンクベース インバウンドリンクとサードパーティリンクのコレクションを含む文書．リンクデータベースとも呼ぶ．

XLink は，拡張リンクと単純リンク，2つの種類のリンクを提供する．拡張リンクは，任意の数のリソースが参加するインバウンドアークやサードパーティアークのような，XLink において定義するすべての機能を提供する．結果として，リモートリソースを示す要素やローカルリソースを含む要素，巡回のルールを定める要素にリンク自体のタイトルなど，その構造は複雑なものとなりうる．一方，単純リンクは，HTML の A タグのような一般的なリンク，すなわち2つの参加要素からなるアウトバウンドリンクを示すための略記方法を提供する．また，XLink は，アプリケーションに対し適切である可能性のあるリンクベースを指示するための機構をもつ．これにより，リンクベースを用いて多数の関連するリンクを集めて管理しても問題が生じない．

次節以降では，討議構造を表現するための XML ボキャブラリである UMML と Linkbase の詳細を述べる．これらの XML ボキャブラリとその役割は次の通りである．

UMML 各発話に属するサフィックスと属性を表現するための XML ボキャブラリ．他の電子メールに含まれる発話との関係は表現しない．

Linkbase コントリビューションによる異なる電子メールに含まれる発話間の関係を表現するための XML ボキャブラリ．各発話に関する詳細な情報は表現しない．

3.3 UMML

UMML(Utterances-in-Mail Markup Language) は，XML 仕様に準拠したマークアップ言語で，討議構造木中における発話に関する情報(サフィックス，属性)を電子メールに付加するための言語である．1つのUMML文書に対し，1つの電子メールが対応するように設計した．DTD全体は付録Aに示す．UMML文書に現れる要素と属性について，順に説明する．

3.3.1 要素: umml

UMMLのルート要素である．UMML自体の名前空間をデフォルトの名前空間として宣言し，UMMLにて用いるXLinkの名前空間をプリフィックスxlinkの名前空間として宣言する．この要素は，電子メールのヘッダ部に相当する要素headerと，ボディ部に相当する要素bodyを，この順番で，1つずつ，子要素として持つ．

```
1 <!ELEMENT umml (header, body)>
2 <!ATTLIST umml xmlns CDATA #FIXED "http://ochimizu-
  www.jaist.ac.jp/mailStructure">
3 <!ATTLIST umml xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink">
4
```

図 3.1: 要素: umml

3.3.2 要素: head

この要素は，電子メールのヘッダ部分をマークアップするための要素である．子要素として，討議構造木中にも現れる情報をマークアップするための要素をもつ．

```
5 <!ELEMENT header (#PCDATA | From | Subject | Date | linkbaseRef)*>
6
```

図 3.2: 要素: header

要素: From

討議構造木において *WhoPr* に相当する要素．電子メール中の From フィールドの値を要素内容とする．

```
7 <!ELEMENT From (#PCDATA)>
```

```
8
```

図 3.3: 要素: From

要素: Subject

討議構造木におけるサフィックス中のサブジェクト番号に相当する要素．電子メール中の Subject フィールドの値を要素内容とし，サブジェクト番号を属性として保持する．

```
9 <!ELEMENT Subject (#PCDATA)>
```

```
10 <!ATTLIST Subject num CDATA #REQUIRED>
```

```
11
```

図 3.4: 要素: Subject

要素: Date

メッセージが記述された時刻と，マークアップを行った時刻を保持する要素．メッセージが記述された時刻を要素内容として保持し，マークアップを行った時刻を属性として保持する．この情報は，クライアントによるマークアップ内容の更新の検出に用いる．

```
12 <!ELEMENT Date (#PCDATA)>
```

```
13 <!ATTLIST Date marked CDATA #REQUIRED>
```

```
14
```

図 3.5: 要素: Date

要素: linkbaseRef

この電子メール中に含まれる発話と他の発話との関係を記述したLinkbase 文書を示す，単純リンクである．要素内容はこのUMML 文書のファイル名または URL とし，属性で，Linkbase 文書へのリンクを表現する．名前空間が示す通り，属性はすべて XLink 由来のものである．属性xlink:arcrole に指定された値，<http://www.w3.org/1999/xlink/properties/>により，この要素は，第 3.2 節にて述べた，有効なリンクベース (Linkbase 文書) を示すためのリンクとして機能する．

```

15 <!ELEMENT linkbaseRef (#PCDATA)>
16 <!--ATTLIST linkbaseRef xlink:role CDATA #IMPLIED-->
17 <!--ATTLIST linkbaseRef xlink:arcrole CDATA #FIXED
    "http://www.w3.org/1999/xlink/properties/linkbase"-->
18 <!--ATTLIST linkbaseRef xlink:title CDATA #IMPLIED-->
19 <!--ATTLIST linkbaseRef xlink:show (new|replace|embed|other|none)
    "new"-->
20 <!--ATTLIST linkbaseRef xlink:actuate (onLoad|onRequest|other|none)
    "onLoad"-->
21 <!--ATTLIST linkbaseRef xlink:type (simple) "simple"-->
22 <!--ATTLIST linkbaseRef xlink:href CDATA #IMPLIED-->
23

```

図 3.6: 要素: linkbaseRef

3.3.3 要素: body

電子メールのボディ部をマークアップするための要素である。子要素として、ボディ部に現れる発話をマークアップするための要素utteranceをもつ。

```

24 <!ELEMENT body (#PCDATA | utterance)*>
25

```

図 3.7: 要素: body

要素: utterance

討議構造木中において1つの発話に相当する要素。発話自体を要素内容とし、発話に関する各種属性(発話の種類、各サフィックス、WhomPr)を属性として持つ。さらに、第4章にて述べる討議構造抽出エンジンにおいて、発話の切出の原因となった手がかり(手がかり語、フラグ)についても、属性として保持することができる。子要素として、この発話と隣接する引用部を示す要素quoteを持つことができる。

要素: quote 引用部を発話と区別するために導入した、引用部を要素内容とする要素である。

実際にマークアップを行った例を図 3.16 に示す。

```

26 <![ELEMENT utterance (#PCDATA | quote)*>
27 <![ATTLIST utterance type CDATA #IMPLIED>
28 <![ATTLIST utterance Ac CDATA #IMPLIED>
29 <![ATTLIST utterance Ac2 CDATA #IMPLIED>
30 <![ATTLIST utterance Pr ID #IMPLIED>
31 <![ATTLIST utterance whomPr CDATA #IMPLIED>
32 <![ATTLIST utterance clues CDATA #IMPLIED>
33 <![ATTLIST utterance flags CDATA #IMPLIED>
34

```

図 3.8: 要素: utterance

```

35 <![ELEMENT quote (#PCDATA)]>

```

図 3.9: 要素: quote

3.4 Linkbase

異なるメールに含まれる発話間の接続関係は、リンクベースに格納したサードパーティリンクとして表現する。ただし、第 3.2 節に述べたように、XLink は XML 文書のリンク機能を規定するための名前空間と属性と制約を与えるものであり、具体的なボキャブラリを定義する必要がある。Linkbase は、その UMML に対するサードパーティリンクを集めたリンクベースを記述するための XML ボキャブラリである。DTD 全体は付録 B に示す。Linkbase 文書に現れる要素と属性について、順に説明する。

3.4.1 要素: linkbase

Linkbase 文書のルート要素である。Linkbase 自体の名前空間をデフォルトの名前空間として宣言し、XLink の名前空間をプリフィックス xlink の名前空間として宣言する。子要素として、発話間の関係を保持する要素 link 要素を 0 個以上、Ac をもつ一つの発話につき一つの link 要素を保持する。

```

1 <![ELEMENT linkbase (link*)>
2 <![ATTLIST linkbase xmlns CDATA #FIXED "http://ochimizu-
  www.jaist.ac.jp/mailStructure/properties/linkbase">
3 <![ATTLIST linkbase xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink">
4

```

図 3.10: 要素: linkbase

3.4.2 要素: link

Ac をもつ発話の接続関係を拡張リンクとして記述するための要素である。

```
5 <!ELEMENT link (locator+, arc+)>
6 <!ATTLIST link xlink:role CDATA #IMPLIED>
7 <!ATTLIST link xlink:title CDATA #IMPLIED>
8 <!ATTLIST link xlink:type (extended) "extended">
9
```

図 3.11: 要素: link

要素: locator

発話がどのUMML 文書に含まれているかを，リモートリソースとして示す。

```
10 <!ELEMENT locator EMPTY>
11 <!ATTLIST locator xlink:role CDATA #IMPLIED>
12 <!ATTLIST locator xlink:title CDATA #IMPLIED>
13 <!ATTLIST locator xlink:label NMTOKEN #IMPLIED>
14 <!ATTLIST locator xlink:type (locator) "locator">
15 <!ATTLIST locator xlink:href CDATA #REQUIRED>
16
```

図 3.12: 要素: locator

要素: arc

サードパーティークの形式で，発話と発話を結び付ける。

```
17 <!ELEMENT arc EMPTY>
18 <!ATTLIST arc xlink:arcrole CDATA #IMPLIED>
19 <!ATTLIST arc xlink:title CDATA #IMPLIED>
20 <!ATTLIST arc xlink:show (new|replace|embed|other|none) "new">
21 <!ATTLIST arc xlink:actuate (onLoad|onRequest|other|none) "onLoad">
22 <!ATTLIST arc xlink:type (arc) "arc">
23 <!ATTLIST arc xlink:from NMTOKEN #IMPLIED>
24 <!ATTLIST arc xlink:to NMTOKEN #IMPLIED>
```

図 3.13: 要素: arc

```

1 <?xml version='1.0'?>
2 <!DOCTYPE linkbase SYSTEM "Linkbase.dtd">
3 <linkbase>
4 <link xlink:type="extended">
5 <locator xlink:type="locator" xlink:label="C1.3.0.1-Ac1.3.0.1-C1.3.0.1.4.0.1-Pr1.3.0.1.4.0.1" xlink:
   href="4.xml#C1.3.0.1-Ac1.3.0.1-C1.3.0.1.4.0.1-Pr1.3.0.1.4.0.1" xlink: title ="4.xml#C1.3.0.1-Ac1
   .3.0.1-C1.3.0.1.4.0.1-Pr1.3.0.1.4.0.1"/>
6 <locator xlink:type="locator" xlink:label="C1.3.0.1-Pr1.3.0.1" xlink:href="3.xml#C1.3.0.1-Pr1.3.0.1
   " xlink: title ="3.xml#C1.3.0.1-Pr1.3.0.1"/>
7 <arc xlink:type="arc" xlink:from="C1.3.0.1-Pr1.3.0.1" xlink:to="C1.3.0.1-Ac1.3.0.1-C1.3.0.1.4.0.1-
   Pr1.3.0.1.4.0.1" xlink:show="new" xlink:actuate="onRequest" xlink: title ="Ac_Pr"/>
8 </link>

```

(中略)

```

422 </linkbase>

```

図 3.14: Linkbase の例

実際のLinkbase 文書の例を図 3.14 に示す .

3.5 UMML+Linkbase によるマークアップ例

図 3.16 , 図 3.17 に示す 2 つのUMML 文書と , 図 3.14 に示す 1 つのLinkbase 文書は , 図 3.15 に示す討議構造木を表現したものである .

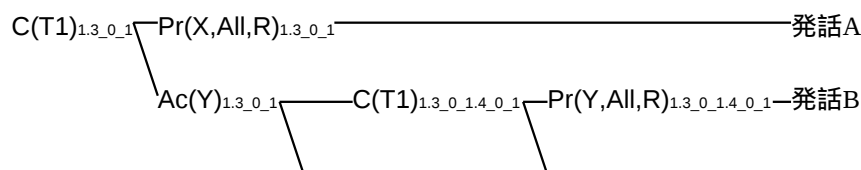


図 3.15: 討議構造木

次章では , 討議を構成する電子メール群から討議構造を自動抽出し , UMML+Linkbase 文書群を生成するシステム , 討議構造抽出エンジンについて述べる .

```
1 <?xml version='1.0' encoding='ISO-2022-JP'?>
2 <!DOCTYPE umml SYSTEM "UMML.dtd">
3 <umml>
4 <header>
5 Return-Path: exp-admin@nuts.sample.sp
6 X-UIDL: /]D&quot;(!_8]&quot;!&?&];]!!mwj!!
7 Return-Path: &lt;exp-admin@nuts.sample.sp&gt;
```

(中略)

```
12 Date: <Date marked="Mon, 20 Nov 2000 14:52:04 +0900">Mon, 20 Nov
    2000 14:52:10 +0900</Date>
13 Posted: Mon, 20 Nov 2000 14:51:34 +0900 (JST)
14 From: <From>hanako@sample.sp (Hanako)</From>
15 Reply-To: exp@nuts.sample.sp
16 Subject: <Subject num="1">[exp:00007] Re: read-finish</Subject>
17 To: exp@nuts.sample.sp (exp ML)
```

(中略)

```
24 X-ML-Info: If you have a question, send a mail with the body &quot;#
    help&quot; (without quotes) to the address exp-ctl@nuts.sample.sp
25 X-UMML: <linkbaseRef xlink:href="linkbase.xml">3.xml</linkbaseRef>
26 Precedence: bulk
27 Lines: 9
28 Content-Type: text
29 Status: U
30 </header>
31 <body>
32 <utterance type="0" whomPr="">
33 花子です。
34 </utterance>
35 <utterance type="1" Pr="C1.3_0_1-Pr1.3_0_1" whomPr=""
    flags="C1.3_0_1-Pr1.3_0_1=0,">
36 今、読み終わりました。
37 足りない内容の把握のため 15:00 までかけてもう一読みします。
38 その間、お二人には、「mail、メール、メール」といった語の統一と、
39 メールによるサービスが実行される前提条件のチェックを、
40 もう一度お願いします。
41 </utterance>
42 --
43 Hanako &lt;hanako@sample.sp&gt;
44 </body>
45 </umml>
```

図 3.16: UMML によるマークアップ例

```
1 <?xml version='1.0' encoding='ISO-2022-JP'?>
2 <!DOCTYPE umml SYSTEM "UMML.dtd">
3 <umml>
4 <header>
5 Return-Path: exp-admin@nuts.sample.sp
6 X-UIDL: =V+#!WDS!!l@Q&quot;!\+*!!
```

(中略)

```
11 Received: from localhost (HELO nuts.sample.sp) (127.0.0.1) by local-
12 host with SMTP; 20 Nov 2000 06:00:03 -0000
13 Date: <Date marked="Mon, 20 Nov 2000 14:59:56 +0900">Mon, 20 Nov
14 2000 15:00:02 +0900</Date>
15 Posted: Mon, 20 Nov 2000 14:59:40 +0900
16 From: <From>Jiro &lt;jiro@sample.sp>;</From>
17 Reply-To: exp@nuts.sample.sp
18 Subject: <Subject num="1">[exp:00008] Re: read-finish </Subject>
19 To: exp@nuts.sample.sp (exp ML)
```

(中略)

```
23 X-ML-Info: If you have a question, send a mail with the body &quot;#
24 help&quot; (without quotes) to the address exp-ctl@nuts.sample.sp
25 X-UMML: <linkbaseRef xlink:href="linkbase.xml">4.xml</linkbaseRef>
26 Mime-Version: 1.0 (generated by tm-edit 7.106)
27 Precedence: bulk
28 Lines: 8
29 Content-Type: text/plain; charset=ISO-2022-JP
30 Status: U
31 </header>
32 <body>
33 <utterance Pr="C1.3_0_1-Ac1.3_0_1-C1.3_0_1.4_0_1-Pr1.3_0_1.4_0_1"
34 type="1" clues="ね&lt; 助詞, 終助詞&gt; 。&lt; 記号, 句点&gt; [Clue6]"
35 Ac="C1.3_0_1-Ac1.3_0_1" flags="C1.3_0_1-Pr1.3_0_1=0,C1.3_0_1.4_0_1-
36 Pr1.3_0_1.4_0_1=3," whomPr="hanako@sample.sp">
37 <quote>
38 &gt; その間、お二人には、「mail、メール、メール」といった語の統一と、
39 &gt; メールによるサービスが実行される前提条件のチェックを、
40 &gt; もう一度お願いします。
41 </quote>
42 よく似てるのですが、「format、フォーマット」もどちらかに統一する
43 必要がありますね。
44 </utterance>
45 </body>
46 </umml>
```

図 3.17: メールB

第 4 章

討議構造木の自動抽出: 討議構造木抽出エンジン

4.1 自動抽出の必要性

方針 2 . 参加者に対し , 既存の環境における負荷上の負荷を与えないにおいて述べたように , 実現する支援環境においては , 参加者に対して表現上の制限を加えることもしないし , メタ情報の入力も強要しない . したがって , 討議構造木を自動抽出するシステムである討議構造木抽出エンジンは , 実現する支援環境において必要不可欠な構成要素である . この章では , その自動抽出エンジンにおいて , どのような手がかりを元に , どのようにして討議構造を抽出しているのか説明する .

4.2 電子メールを用いたコミュニケーションにおける言語的特徴

4.2.1 電子メールボディ部の特徴

あるメールに対する返答を含むメールのメッセージ部では , 引用文が利用されているという特徴がある . 引用文とは , 各行の先頭に “》” , “>” といった引用符を伴ってあらわれる文章であり , 以前の議論であらわれた文章を引用している文章のことである . 引用文は , 引用文の直後にあらわれる発話の先行発話を示すことが多い . また , メッセージの最初は , 自分の名前や身分を示す宣言的な発話であることが多く , メッセージの最後は , 一般に “署名” と呼ばれるものであることが多い . 署名は , 自分の名前や身分や連絡先などをまとめたもので , 討議と直接の関係はない . 署名が存在する場合は , 署名の直前に

示唆する特徴	略号	例
話題の並列展開	Clue#1	列挙します，挙げます
話題の転換	Clue#2	ところで，～については
接続語による結束性(詳細の関係)	Clue#3	具体的には，例えば
接続語による結束性(前提の関係)	Clue#4	それで，だから
指示語による結束性	Clue#5	この場合，これは
返答の要求	Clue#6	か，しましょう，?
同意・了承	Clue#7	了解です，確に
対となる表現	Clue#8	～と思いませんか－思います

表 4.1: 言語的手がかりの一覧

“---”，“-=-=-=”といった記号だけからなる行(セパレータ)を伴うことが多い。

4.2.2 文章表現にあらわれる特徴

空行や改段落による区切り

電子メールを利用したコミュニケーションでは，対面での対話と異なり，会話の一時停止などを利用した話題転換の指示ができない．その代わりに，文章中に意図的に空行や改段落を挿入することによって話題転換の指示を行なっているものと考えられる．したがって，空行や改段落で区切られた文章が発話を構成する基本単位となる．しかし，発話の最低単位は一文であり，一文中に空行や改段落があっても一文が発話であることが多い．

新たな話題の並列展開を示唆する表現

相手からの返答を求めない発話の中には，新たに議論される複数の話題を明示的に示しているものがある．たとえば，「挙げておきます」のような表現は，続く空行や段落で分割される文章を，それぞれ異なった話題の最初の発話とするための手がかりになることが予想される．そこで，新たな話題の並列展開を示唆する表現に着目する．その表現例を表 4.1 中の Clue#1 に示す．

話題の転換を示唆する表現

会話における話題の転換では，それまでの話題を終始，あるいは停止し，新たな議論の開始を示唆する表現が用いられることがある [19][20]．同一メール中の隣接した発話間においても，発話の先頭文中に，話題転換を示唆する文副詞，接続詞があることがある．また，明示的に新たな話題の開始を示唆することによって，話題の転換を示唆することもある．これらの例を，表 4.1 中の Clue#2 に示す．

結束性を示唆する表現

文章の長さや読みやすさを考慮して，空行や改段落を用いて一つの話題の文章を分割することがある．その結果，メール中の空行や改段落によって分割された文章同士でも，同一の話題に言及している場合がある．そのような文章中には，後続の文章中に先行する文章との間の結束性を示唆する指示語や接続語が現れる．結束性とは，文や発話の間の「言語的なつながり」である [21]．結束性を示唆する接続語は「具体的には」のように後続文章が先行文章の詳細に当たることを示唆する接続語群と「だから」のように先行文章が後続文章の前提に当たることを示唆する接続語群の 2 つのカテゴリに分類できる．それぞれの例を表 4.1 中の Clue#3，Clue#4，Clue#5 に示す．

返答の要求を示唆する表現

返答を要求する発話には，相手への質問や提案などを示す語，つまり相手への返答の要求を示唆する表現が含まれていることが多い．表 4.1 中の Clue#6 にその表現例を示す．

同意・了承を示唆する表現

話題を終結する発話の場合，文章の先頭文の文頭あるいは文末に，相手の発話を理解し，同意を示す表現が含まれていたり，先頭文の文頭に感嘆を示す語が含まれる傾向にある．その上，先頭文以外の文末に，返答の要求を示唆する表現 (Clue#6) が含まれない傾向にある．表 4.1 中の Clue#7 に同意・了承を示唆する表現例を示す．

対を構成する表現

発話ペアを構成する先行発話の最後の文末と後続発話の先頭文には、相手への問いかけと、それに対応した同意を示す表現の対が存在することがある。これらの対は、発話ペアを見つけるための手がかりになるだけでなく、後続発話が話題を終結する発話であるかを決定する手がかりにもなると思われる。その表現例を表 4.1 中の Clue#8 に示す。

4.3 抽出アルゴリズム

4.3.1 概略

本アルゴリズムでは、以下の情報を用いて、指定されたメールのボディ部から発話を抽出し、指定されたメールの現れる直前の討議構造と結び付ける。

- 入力
 - 複数人による討議に現れる言語的特徴
(構造的手がかり/言語的手がかり)
 - 討議参加者を示す語彙
 - 指定されたメールが現れる直前の討議構造を表現する UMML+Linkbase 文書
- 出力
 - 指定されたメールを含む討議構造を表現する UMML+Linkbase 文書

抽出手順は大きく次の 3 つからなる。

- 手順 1 手がかり語辞書とメンバーリストの読み込み
- 手順 2 ボディ部から発話候補文章部を抽出
- 手順 3 発話候補文章部から発話を抽出

本アルゴリズム中で用いる情報の格納形式等について触れた後、各手順の詳細について、順に説明する。

手がかり語辞書

複数人による討議に現れる言語的手がかりをまとめたものである。手がかり語辞書は以下の形式に沿ったテキストファイルとし、抽出エンジンと同じディレクトリに配置する。

- 手がかり語の示唆する内容毎に 1 つのファイルとする。
- ファイル名は Clue?_dic.txt とする。?には 1 から 7 の数字が入る。
Clue1_dic.txt には、clue1「話題の並列転換を示唆する手がかり語」

```

1 ましょう [。 . ]$
2 でしょう [。 . ]$
3 \?$
4 ?$
5
6 か<助詞, 副助詞 / 並立助詞 / 終助詞> [\S]+?<記号 [\S]+? $
7 か<助詞, 副助詞 / 並立助詞 / 終助詞> な<助詞, 終助詞> [\S]+?<記号 [\S]+? $
8 よ<助詞, 終助詞> [\S]+?<記号 [\S]+? $
9 ね<助詞, 終助詞> [\S]+?<記号 [\S]+? $
10 けど<助詞, 接続助詞> [。 ??]+?<記号 [\S]+? $

```

図 4.1: 手がかり語辞書の例

を格納する .

- 各手がかり語辞書は次のフォーマットからなる .

1 つの手がかり語につき 1 行とし , 空行をセパレータとして , 前半に「形態素情報を必要としない手がかり語」, 後半に「形態素情報が必要な手がかり語」を集める .

- 形態素情報が必要な手がかり語については , 次のフォーマットにおける茶筌の解析結果を与える .

フォーマット '%M<%P, >_ '

手がかり語例 "挙げます"

→ "挙げる<動詞, 自立>_ます<助動詞>"

メンバーリスト

討議参加者を示す語彙をまとめたものである . メンバーリストは以下の形式に沿ったテキストファイルとし , 討議を構成するメール群を格納したディレクトリと同じディレクトリに配置する .

- 1 つのメールフォルダにつき 1 つのファイルとする .
- ファイル名は members とする .
- メンバーリストは次のフォーマットからなる .

CSV 形式 , 参加者 1 人につき 1 行 . 先頭に参加者のメールアドレス . それ以降に , その参加者を示す語 . 語数は問わない .

4.3.2 手順 1．手がかり語辞書とメンバーリストの読み込み

手順 1.1．手がかり語辞書の読み込み

手がかり語辞書を元に、手順 3 で用いる評価関数を生成する。この評価関数は、引数に発話候補文章および発話を取り、もし、その文章中に手がかり語があれば、その語とその語が属するカテゴリを示す文字列を返す。無ければ空文字列を返す。

手順 1.2．メンバーリストの読み込み

メンバーリストを元に、手順 3 で用いる評価関数を生成する。この評価関数は、引数に文字列を取り、もし、その文字列中にメンバーを示す語があれば、その文章中に現れたメンバーのリストを返す。無ければ、空のリストを返す。

4.3.3 手順 2．ボディ部から発話候補文章部を抽出

空行や改段落などを利用し、ボディ部を引用部とシグネチャ部と発話候補文章部にわけける。

引用部 先行する文章を引用している部分。引用記号を伴う。

シグネチャ部 送信者に関連する情報が記述されている部分。発話候補文章部と区別できるように、セパレータを伴う。

発話候補文章部 発話を構成する文章からなる。(言い換えるならば、段落または箇条書きの各箇条からなる)

具体的な分割手順は次の通りである。対象行とは処理を行う対象である行のことを示し、前の対象行とは、現在の対象行が対象行となったときの対象行を示し、次の行とは、対象行の直後の行を示す。

手順 2.1 ボディ部の最後から数行を対象に、セパレータの探索をおこなう。探索は上からボディ部末に向かって行い、セパレータが見つかり次第、そのセパレータからボディ部の最後までを発話の候補からはずす。探索が終わり次第、対象行をボディ部の先頭行に変更する。

手順 2.2 対象行が引用符を伴っているならば、引用符を伴わない行まで読み飛ばす。対象行をその引用符を伴わない行に変更する。

手順 2.3 対象行が空行なら，空行でない行まで読み飛ばす．その空行でない行を対象行として，手順 1-2 に戻る．

手順 2.4 対象行が改段落であり，かつ前の対象行が空行または引用文でなければ，対象行を新たな文章の始まりとする．次の行を対象行として，手順 1-2 に戻る．

手順 2.5 対象行は発話の候補となる文章を構成する行である．もし，前の対象行が引用文でも空行でもなければ，対象行は前の対象行を含む文章を構成する文章である．そうでなければ，新たな文章を構成する行である．次の行を対象行として，手順 1-2 に戻る．

4.3.4 手順 3．発話候補文章部から発話を抽出

引用部以前に存在する発話候補文章部を文章集合 A，引用部以降に存在する発話候補文章部を文章集合 B_i と呼ぶ．さらに，文章集合 A に含まれる文章を，導入順に，文章 A-1，

1	文章集合 A
2	引用部
3	文章集合 B1
4	引用部
5	文章集合 B2
6	引用部
7	:

図 4.2: ボディ部と発話候補文章部の関係

文章 A-2, ..., 文章 A-n と表現し，文章集合 B1 に含まれる文章を，導入順に，文章 B1-1，文章 B1-2, ..., 文章 B1-n と表現する．これらの文章から発話を抽出し，討議構造木を構築する．

手順 3.1．文章集合 A からの発話仮抽出

もし，ボディ部が引用部から始まっているのであればこの手順は省略し，手順 3.2 に進む．

手順 3.1.1．宣言的な発話の仮抽出 宣言的な発話を検出し，仮抽出する．指定された文章が宣言的な発話かどうかを判定するための手順については手順 3.A として別に記す．

```

1  n=1;
2  while (A-nが存在する and n <= 2) {
3      if (A-nが宣言的な発話である(手順3.A)) {
4          A-nを宣言的な発話として仮抽出;
5          n++;
6      } else {
7          break;
8      }
9  }
10 if (B1-1が存在する and A-2が存在しない) {
11     A-1を宣言的な発話として仮抽出;
12     n=2;
13 }
14 A-nを対象文章として、手順3.1.2へ

```

図 4.3: 手順 3.1.1

手順 3.1.2 . 複数文章からの発話仮抽出 手順 3.1.1 で指定した文章を対象文章として手順 3.B を行う .

手順 3.2 . 文章集合 **Bi** からの発話仮抽出

Bi-1 を対象文章として手順 3.B を行う .

手順 3.3 . サブジェクト番号と発話者の確定

討議構造木に現れるサブジェクト番号と発話者 (*WhoPr*) を確定する .

手順 3.3.1 . サブジェクト番号の確定 同じサブジェクトを持つメールは同じサブジェクト番号を持つ . ここでいう「同じサブジェクト」とは、メーリングリスト固有の文字列や返信識別子や余計な空白を取り除いた上で同じ文字列となるサブジェクトとする .

まず、フィールドは折り返されていることがあるので、それを除去する . 次に、MIME エンコードしてあるならばデコードしたのち、空白を除去する、さらに、メーリングリストを識別ための文字列 (ex. [rise-ml]) を除去したのち、返信識別子 (ex. Re:) を除去する . 過去に同一サブジェクトのメールがあるならば、サブジェクト番号は既に決定しているのでその番号を与える . そうでなければ、そのサブジェクトを示す新しい番号を与える、

手順 3.3.2 . 発話者の確定 処理対象メールのヘッダ部中に含まれる From: フィールドとメンバーリストを利用して、発話者を確定する .

手順 3.4 . 文章集合 A からの発話抽出

手順 3.1 と同様、ボディ部が引用部から始まっているのであれば、この手順は省略し、手順 3.5 に進む .

手順 3.4.1 . 宣言的な発話の抽出 手順 3.1.1 での抽出結果を用いて、宣言的な発話を検出し、抽出する .

```
1  n=1;
2  while (A-n が存在する and
3      A-n は宣言的な発話である (cf. 手順 3.1.1) and
4      n < 2) {
5      A-n に現れるメンバーのリストを取得する (手順 3.A);
6      取得したメンバーのリストを tempWhomPr と呼ぶ .
7      if (tempWhomPr が空リストでなければ) {
8          whomPr を確定する (手順 3.C);
9          A-n を宣言的な発話として抽出;
10         n++;
11     } else {
12         break;
13     }
14 }
15 if (B1-1 が存在する and A-2 が存在しない) {
16     whomPr を確定する (手順 3.C);
17     A-1 を宣言的な発話として抽出;
18     n = 2;
19 }
20 A-n を対象文章として、手順 3.4.2 へ
```

図 4.4: 手順 3.4.1

手順 3.4.2 . 複数文章からの発話抽出 (文章集合 A) 指定された対象文章から引用文の直前の発話候補文章までを対象に、発話の抽出を行う .

具体的な手順は、手順 3.D である . 手順 3.D に与える条件は次の通り .

- 一つ前に抽出した発話は、返答を要求する発話でも話題を終結する発話でもない .
したがって、一つ前に抽出した発話の *Pr* は「ない (空文字列)」

- *WhomPr* は、一つ前に抽出した発話の *WhomPr* .
- 次に抽出する発話は新しい話題 . [フラグは 0 .]
- 返答発話ではない .

手順 3.5 . 文章集合 **Bi** からの発話抽出

手順 3.5.1 . 先行発話の確定 引用行を手がかりとして先行発話を確定 , 先行発話の *Pr* と *WhoPr* , *whomPr* , フラグを取得する . 引用部が複数の行からなる場合 , 一番下の行から順に手がかりとして用いる . (下の例の場合 , 5,4,3,2,1 の順)

>引用行 1
>引用行 2
>引用行 3
>引用行 4
>引用行 5

手がかりとして用いる引用行を手がかり行と呼ぶ .

```

1  if (手がかり行が存在しない) {
2      return  検索失敗;
3  }
4  手がかり行をもちいて引用元を検索 .
5  while (引用元候補が見つかった)
6      if (引用元が発話である) {
7          return  引用元の Pr , whoPr , whomPr , フラグ;
8      }
9      引用元 (行) を手がかりに引用元を検索 .
10 }
11 手がかり行を 1 行繰り上げて , はじめからやり直し .

```

図 4.5: 手順 3.5.1

手順 3.5.1' . 先行発話の確定に失敗した場合 手順 3.D によって発話の抽出を行う . ただし , 前の文章集合からの発話抽出において最後に抽出した発話の種類によって与える条件が異なる .

最後に抽出した発話が , 宣言的な発話以外の発話なら , その発話の *Pr* と同じ高さ (深さ) から発話を抽出する . 手順 3.D に与える条件は次の通り .

- これから抽出する発話の Pr の元となる Pr は，最後に抽出した発話の Pr .
- $WhomPr$ は，最後に抽出した発話の $WhomPr$.
- これからは抽出する発話の Pr の木の高さは最後に抽出した発話の Pr と同じ . [フラグは 2 .]
- 返答発話ではない .

前の文章集合がない，または，最後に抽出した発話が，宣言的な発話なら，新しい話題の導入として発話の抽出を始める .

手順 3.D に与える条件は次の通り .

- 一つ前に抽出した発話は，返答を要求する発話でも話題を終結する発話でもない .
したがって，一つ前に抽出した発話の Pr は「ない(空文字列)」
- $WhomPr$ は，一つ前に抽出した発話の $WhomPr$.
- 次に抽出する発話は新しい話題 . [フラグは 0 .]
- 返答発話ではない .

手順 3.D による抽出が終わったならば，次の文章集合 $Bi+1$ を対象として手順 3.5 に移る .

手順 3.5.2 . 後続発話の仮抽出・一部属性確定 手順 3.5.3(手順 3.D) で付加できない属性 (Ac および先行発話のフラグの継承，話題の終結・未終結属性) の決定・保存と，手順 3.5.3(手順 3.D) では計算できない後続発話としての Pr の決定を行う .

手順 3.5.2.1 . Ac の確定，および，手順 3.5.2.2 . 後続発話を構成する文章の確定と話題の終結・未終結属性の保存 手順 3.5.1 によって特定した先行発話の各種情報と，手順 3.3 で特定したサブジェクト番号と発話者を利用して，後続発話の Ac を確定する . もし，後続発話の Ac が $InAc$ でなく Ac であるならば，引き続き，後続発話を構成する文章を確定し，話題の終結を示唆しているかそうでないかを判定，発話に付随する属性として保存する (図 4.6，図 4.7) . ここでは，先行発話の Pr ， $WhoPr$ ， $WhomPr$ ，フラグを，それぞれ， $prePr$ ， $preWhoPr$ ， $preWhomPr$ ， pre フラグと呼び，手順 3.3 で特定したサブジェクト番号を sn ，発話者を $WhoPr$ と呼ぶ .

手順 3.5.2.3 . 仮 Pr 確定 返答発話の Pr に付随するサフィックスのラベルの数は，その返答発話が修復シーケンスの導入であれば増え，修復シーケンスの導入でなく，かつ，話題の終結を示唆する内容であれば減る可能性がある . この観点から， $Bi-1$ を含む発話候補文章を，次の通り 3 つの種類に分類する (表 4.2) . それを求める手順は手順 3.5.3 に含まれな

```

1  if (preWhomPr に whoPr が含まれない) {
2      prePr の末端の 'Pr' を 'InAc' に書き換えたものを後続発話の Ac とする;
3      // InAc である発話が, その話題を終結を Pr することはない.
4      // デフォルトである「返答要求」.
5  } else {
6      prePr の末端の 'Pr' を 'Ac' に書き換えたものを後続発話の Ac とする;
7      // Ac はである発話は, その話題の終結を Pr する可能性があるので,
8      // 調べる.
9      Bi-1 を対象文章とする;
10     while (1) {
11         手順 3.D.1(注);
12         if (まだ発話が確定しない) {
13             next;
14         }
15         // この時点で, 文章 Bi-1 から始まる発話が仮確定している.
16         // 以降, 仮発話と呼ぶ.
17
18         if (Bi の中において, 仮発話以降に発話候補文章が存在しない) {
19             if (対象発話は話題の終結を示唆する発話である) {
20                 対象発話が話題の終結を示唆する発話であることを保存;
21             }
22             last;
23         }
24
25         // 以降, 仮発話の直後にある発話候補文章を続く文章と呼ぶ.
26         if (続く文章中に Clue2 が存在する) {
27             if (仮発話は話題の終結を示唆する発話である) {
28                 仮発話が話題の終結を示唆する発話であることを保存;
29             }
30             last;
31         }

```

図 4.6: 手順 3.5.2.1, および, 手順 3.5.2.2 (前半)

	Clue6 あり	Clue6 なし
Clue7 あり	話題未終結/変化無し	話題終結/減る?
Clue7 なし	話題未終結/増える	話題未終結/変化無し

表 4.2: 返答発話の状態と手がかり語との関係

```

33     if (続く文章中に Clue3 が存在する) {
34         if (仮発話は話題の終結を示唆する発話である) {
35             仮発話と続く文章を連結して 1 つの発話候補文章とする;
36             連結した文章を対象文章とする;
37             next;
38         }
39         last;
40     }
41
42     if (続く文章中に Clue4 または Clue5 が存在する) {
43         if (仮発話は話題の終結を示唆する発話である
44             または、続く文章中には Clue6 が存在しない) {
45             仮発話と続く文章を連結して 1 つの発話候補文章とする;
46             連結した文章を対象文章とする;
47             next;
48         }
49         last;
50     }
51
52     if (対象発話は話題の終結を示唆する発話である) {
53         対象発話が話題の終結を示唆する発話であることを保存;
54         last;
55     }
56
57     if (対象発話中に Clue6 が存在する) {
58         last;
59     }
60
61 }
62 }

```

図 4.7: 手順 3.5.2.1 , および , 手順 3.5.2.2 (後半)

い．次の手順で仮 Pr を確定する (返答発話の Pr の深さを確定する) ．

話題未終結・ラベルの数が増える 手順 3.5.2.3.A

話題終結・ラベルの数は減る？ 手順 3.5.2.3.B

話題未終結・ラベルの数は変わらない 手順 3.5.2.3.C

手順 3.5.2.3.A．ラベルの数が増える この時，これから抽出する発話は，修復シーケンスの導入である．手順 3.5.2.1 で求めた Ac からサフィックスを取り出し，木につける枝 ($-C-Pr$) をつくり， Ac とつなげたものを，仮 Pr とし，この返答発話の Pr のサフィックスに新しいラベルを追加する．

手順 3.5.2.3.B．ラベルの数は減る？ この発話は修復シーケンスの導入ではなく，かつラベルの数を減らす可能性がある．しかし，この発話が手順 3.E により話題の終結を示唆する発話「ではない」と判定されたならば，木の深さは変わらない．この場合は手順 3.5.2.3.C に従う．あとは先行発話の Pr の決定理由によりさらに手順をわけると．

パターン 1(提案・詳細提案など)の「詳細」である 手順 3.5.2.3.B.A

パターン 2(前提・提案など)の「提案」である可能性がある 手順 3.5.2.3.B.B

修復シーケンスの導入である 手順 3.5.2.3.B.C

どれにも該当しない 手順 3.5.2.3.C

手順 3.5.2.3.B.A．パターン 1(提案・詳細提案など)の「詳細」にたいして話題の終結を示唆する発話が行われた場合 ラベルの数は減る．この時，後続発話は直接の先行発話(詳細提案)と，その提案と対応する発話(提案)を受理する．この発話の Pr の木の高さは，(提案)と同じ高さになる．

手順 3.5.2.3.B.B．パターン 2(前提・提案など)の「提案」である可能性がある場合 この状態にはさらに3つの場合が考えられる．

- 前提 (以降，パターン 2.1)
- (前提に対する) 提案 (以降，パターン 2.2)
- 先行発話に対し木の高さが変化しない (以降，パターン 2.3)

これらは，次の手順によって分類できる．パターン 2.1 とパターン 2.3 に対しては，話題の終結を示唆する発話が行われたとしても，木は浅くならない．以降手順 3.5.2.3.C と同等の手順に従う．パターン 2.2 に対して話題の終結を示唆する発話が行われた場合のみ，

```

1 対象となる先行発話の Pr の末端の Pr を -Pr sn.cn1...cni-1.cni とする
2
3  if (cni-1 == cni) {
4      パターン 2.1
5  } else {
6      if (-Pr sn.cn1...cni-1.cni-1 という Pr を持つ発話が存在する) {
7          パターン 2.2
8      } else {
9          パターン 2.3
10     }
11 }

```

図 4.8: 手順 3.5.2.3.B.B

木は浅くなる．この時，後続発話は直接の先行発話 (提案) と，その提案と対応する発話 (前提提案) を受理する．この発話の *Pr* の木の高さは (提案) や (前提提案) よりも一つ浅くなる．

手順 3.5.2.3.B.C．修復シーケンスに対して話題の終結を示唆する発話が行われた場合 先行発話を受理し，木が浅くなる．この発話の *Pr* の木の高さは，先行発話よりも一つ浅くなる．

手順 3.5.2.3.C．木の深さはそのまま この発話は話題の終結を示唆する発話であるが，木を浅くする可能性はない．手順 3.5.1 で求めた先行発話の *Pr* を仮 *Pr* とする．

手順 3.5.2.4．*Ac* およびフラグの保存 手順 3.5.2.1 で確定した *Ac* と，先行発話の持つフラグの履歴を *Bi-1* から始まる発話の属性として保存する．

手順 3.5.3．複数文章からの発話抽出 (文章集合 **Bi**) *Bi-1* から引用文の直前の発話候補文章までを対象に，発話の抽出を行う．

具体的な手順は，手順 3.D である．手順 3.D に与える条件は次の通り．

- 一つ前に抽出した発話，すなわち，これから抽出する発話の *Pr* の元となる発話の *Pr* は「手順 3.5.2 で確定した仮 *Pr*」．
- *WhomPr* は，一つ前に抽出した発話の *WhomPr* と先行発話の *WhomPr* の和から，発話者を取り除いたもの．
- 返答発話である．

- 手順に与える意図は，手順 3.5.2 に依存．次のどちらか．
 - 「最初の発話は返答発話で，修復シーケンスの導入ではない」[フラグは 2.]
 - 「最初の発話は返答発話で，修復シーケンスの導入である」[フラグは 3.]

手順 3.A．宣言的な発話の判定，または，返答を求める相手の抽出

手順 1.2 で生成した評価関数を利用して，指定された発話候補文章の先頭文，あるいは最終文中に，メンバーを示す語が含まれているかを調べ，もしいるならば，それらの文章中に現れたメンバーのリストを返す．

宣言的な発話の判定と，返答を求める相手の抽出には多少違いがある．

宣言的発話の判定では，先頭文のみを考慮し最終文を無視する．さらに，指定された発話が 2 行以上ならば，無条件で空のリストを返す．

返答を求める相手の抽出では，先頭文も最終文も考慮に入れる．行数による制約はない．

手順 3.B．発話の仮抽出

指定された対象文章から引用文の直前の発話候補文章までを対象に，発話の仮抽出を行う．手順は図 4.9 の通り．

手順 3.C．*WhomPr* の確定

手順 3.A(返答を求める相手の抽出)によって求めたメンバーリストから，発話者を除き，現在の *WhomPr* とマージする．

手順 3.D．複数文章からの発話抽出

話題の並列展開(列挙)のみを対象とした発話抽出．それ以外の発話の抽出に関しては，手順 3.D.1 に記述する．

手順 3.D.1．対象文章と次の文章は同じ発話に属する文章か

同じ発話に属する文章ならば，連結した文章をそのまま返す．異なる発話に属する文章ならば，対象行を含む発話を抽出して，各種属性をセット，次の発話候補を示すインデックスを返す．この手順は手順 3.5.2.2 でも用いるが，その場合は，必要に応じた発話候補

```

1 while (対象文章が存在する) {
2     if (対象文章中に Clue1 が存在する) {
3         対象文章を宣言的な発話として仮抽出;
4         for (対象文章の次の文章から
5             引用文の直前の発話候補文章まで) {
6             返答を要求する発話として仮抽出;
7         }
8         発話仮抽出終了;
9     }
10
11 LOOP:
12     if (対象文章が引用文の直前の発話候補文章) {
13         対象文章を返答を要求する発話として仮抽出;
14         発話仮抽出終了;
15     }
16     対象文章の次の文章を単に次の文章と呼ぶ;
17     if (以下のどれかに当てはまる
18         ・ 次の文章中に Clue2 が存在する
19         ・ 次の文章中に Clue3 が存在する
20         ・ 次の文章中に Clue4 が存在する
21         ・ 次の文章中に Clue5 が存在する
22         ・ 対象文章中に Clue6 が存在する
23         ・ 対象文章中に Clue7 が存在する
24     ) {
25         対象文章を発話として仮抽出;
26         次の文章を対象文章とする;
27     } else {
28         対象文章と次の文章を連結;
29         連結した文章を対象文章とする;
30         goto LOOP;
31     }
32 }

```

図 4.9: 手順 3.B

```

1 while (対象文章が存在する) {
2     if (対象文章が宣言的な発話である
3         // 対象文章中に Clue1 が存在する
4         ) {
5         対象文章を宣言的な発話として抽出;
6         for (対象文章の次の発話候補文章から
7             引用文の直前の発話候補文章まで) {
8             フラグに合わせ木の高さを調節する (手順 3.D.A);
9             返答を求める相手の抽出 (手順 3.A);
10            whomPr の確定 (手順 3.C);
11            返答を要求する発話として抽出;
12            フラグを「2: 木の高さは変化しない」とする;
13        }
14        発話仮抽出終了;
15    }
16
17    手順 3.D.1;
18    対象発話は返答発話ではない;
19 }

```

図 4.10: 手順 3.D

の連結のみ行い，各種属性の確定は行わない．

手順 3.D.A フラグに合わせた木の高さの調整 このアルゴリズムにおいては，一つ前に抽出した発話がそのあと抽出する発話の *Pr* の高さ (深さ) に影響を与える．その調整のための手順であり，一つ前に抽出した発話の *Pr* と，フラグを元に，これから抽出する発話の *Pr* を決定する．

もし，フラグが「0:話題の転換」なら，この発話は，一つ前に抽出した発話の話題と関係ないので，木を一番短くする．

「1:詳細化 (ex. 提案/詳細提案)」なら，この発話は，一つ前に抽出した発話に対する詳細であるので，一つ前に抽出した発話に対して一段木を深くする．

それ以外の場合は，一つ前に抽出した話題の継続などで，一つ前に抽出した話題に対して木の高さを変えない．

手順 3.E．話題の終結を示唆する *Pr* の判定

指定された発話が話題の終結を示唆する発話かどうかを判定する．話題の終結を示唆する発話には，発話者以外の参加者を示す語を含まず，Clue7 を含み，かつ，Clue6 を含

```

1  if (続く文章が存在しない) {
2      // 対象文章は発話である .
3
4      if (対象文章中に Clue6 が存在する) {
5          対象文章中に存在する Clue6 と対象文章を結び付ける;
6      } else if (抽出しようとしている発話は返答発話である
7          and 対象文章中に Clue7 が存在する) {
8          対象文章中に存在する Clue7 と対象文章を結び付ける;
9      }
10     手順 3.D.A;
11     対象文章を発話として抽出する;
12 }
13
14 // 続く文章が存在する場合
15 if (続く文章中に Clue2 が存在する) {
16     続く文章が属する発話は、対象文章とは異なる話題である; // [フラグは 0]
17     続く文章中に存在する Clue2 と続く文章文章を結び付ける;
18     手順 3.D.A;
19     対象文章を発話として抽出する;
20 }
21
22 if (続く文章中に Clue3 が存在する) {
23     続く文章が属する発話は、対象文章の「詳細」である; // [フラグは 1]
24     続く文章中に存在する Clue3 と続く文章文章を結び付ける;
25     手順 3.D.A;
26     対象文章を発話として抽出する;
27 }
28
29 if (続く文章中に Clue4 または Clue5 が存在する) {
30     if (続く文章中に Clue6 が存在する) {
31         対象文章は、続く文章が属する発話の「前提」である; // [フラグは 2]
32         続く文章中に存在する Clue3 と続く文章文章を結び付ける;
33         続く文章中に存在する Clue4 と続く文章文章を結び付ける;
34         続く文章中に存在する Clue5 と続く文章文章を結び付ける;
35         手順 3.D.A;
36         さらに木を 1 段深くする; // [手順 3.D.A を利用する]
37         対象文章を発話として抽出する;
38     }
39
40     対象文章と続く文章を接続して次の対象文章とする;
41 }

```

図 4.11: 手順 3.D.1 (前半)

```
43 if (対象文章中に Clue6 が存在する) {
44     続く文章が属する発話は、対象文章とは異なる話題である; // [フラグは 0]
45     対象文章中に存在する Clue6 と対象文章を結び付ける;
46     手順 3.D.A;
47     対象文章を発話として抽出する;
48 }
49
50 if (抽出しようとしている発話は返答発話である
51     and 対象文章中に Clue7 が存在する) {
52     続く文章が属する発話は、対象文章とは異なる話題である; // [フラグは 0]
53     対象文章中に存在する Clue7 と対象文章を結び付ける;
54     手順 3.D.A;
55     対象文章を発話として抽出する;
56 }
57
58 発話の抽出 (各種属性の確定);
59
60 対象文章と続く文章を接続して次の対象文章とする;
```

図 4.12: 手順 3.D.1 (後半)

まないという特徴を利用している．指定された発話が話題の終結を示唆する発話ならば，その発話中に含まれる Clue7 を返す．そうでなければ，空文字列を返す．

次章では，この抽出エンジンによって抽出した討議構造をもとに，方針 4．討議の道筋を容易に参照できるようにするためのクライアントの実装例，ICEMail++を紹介する．

第 5 章

討議構造参照機能を有する メールクライアント: ICEMail++

5.1 構成

抽出エンジンが生成したUMML ファイルとLinkbase を討議構造として利用，討議の道筋を呈示するソフトウェアのプロトタイプとして，ICEMail++(図 5.1) を開発した．これは，Java で記述されたメールクライアント ICEMail [22] に，討議構造呈示部を追加したものである．呈示部は以下に挙げる呈示法を実現しており，随時切り替えることができる．

5.2 発話抽出結果の提示

ユーザーが選択した電子メールから抽出した各発話を表形式で呈示する (図 5.2)．メール中の各発話を，導入順にしたがってセルに格納し，発話の種類をセルの背景色で区別できるようにした．また，引用・被引用部分を，引用部分から被引用部分へ，被引用部分から引用部分へとたどれるハイパーリンクとした．これにより，討議構造木中の発話を順にたどることを可能にしている．

5.3 討議構造木の呈示

ユーザーが選択したメールから抽出した各発話，および，それらの属する討議構造木を呈示する (図 5.3)．1 つの発話に 1 つのフレームをあてている．発話抽出結果の呈示と同

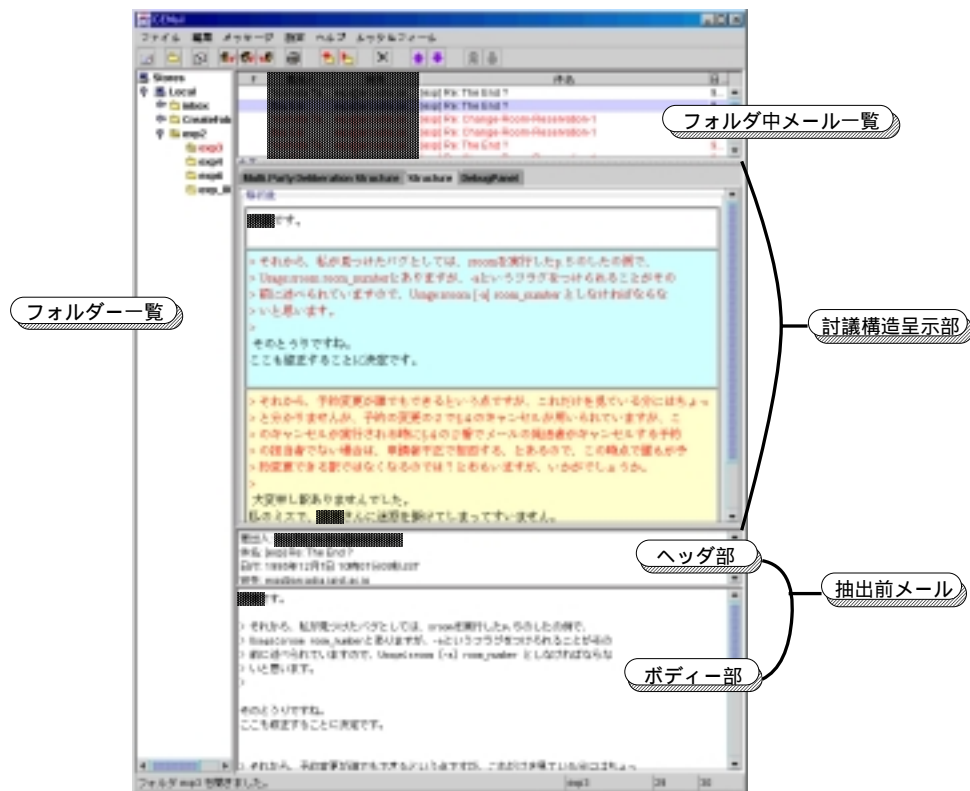


図 5.1: ICEMail++

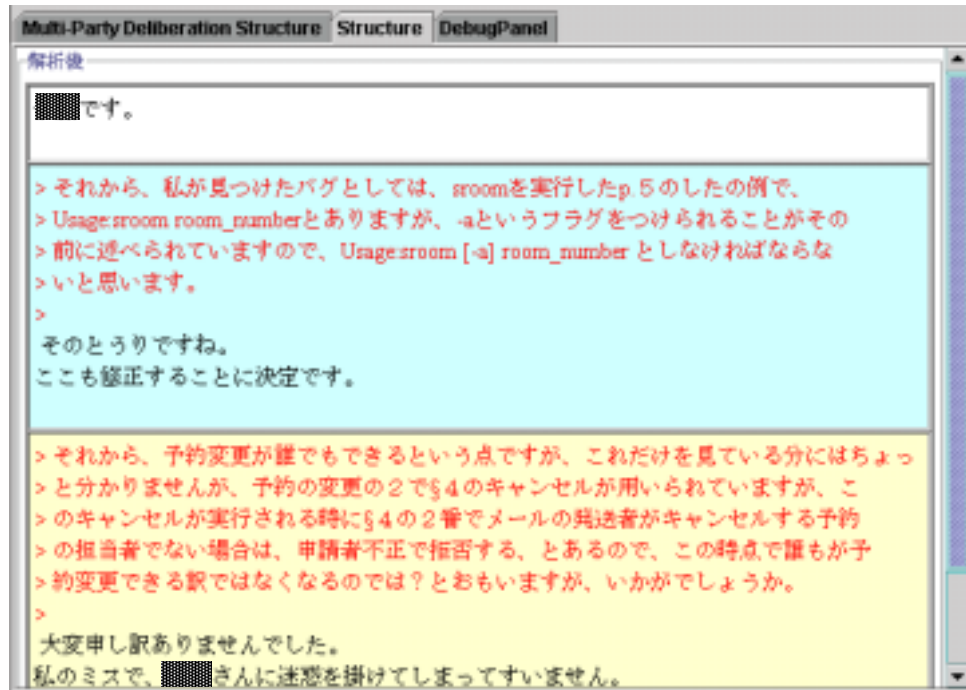


図 5.2: 解析結果の呈示

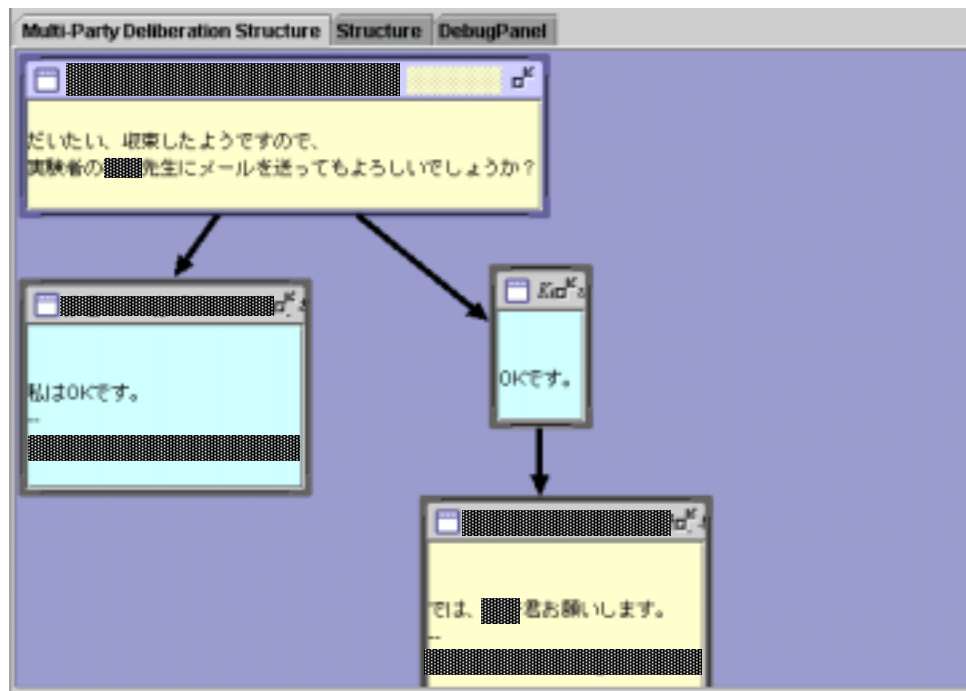


図 5.3: 討議構造木の呈示

様，フレーム内の背景色は発話の種類を示している．プレゼンテーションフェーズとしての発話からアクセプタンスフェーズとしての発話への矢印でコントリビューションを表現する．また，発話を示すフレームを選択することによって，その発話の属するメールを参照することができる．

第 6 章

おわりに

6.1 まとめ

電子メールを用いたコミュニケーションにおいて生じる問題に対する解の 1 つとして，電子メールを用いたコミュニケーションにおける会話のモデルである討議構造木モデルに基づく討議構造を表現するための XML ボキャブラリを設計し，討議の道筋として討議構造木を抽出する討議構造抽出エンジンと討議構造木の参照機能をもつメールクライアントからなる支援環境を設計し，実現した．

6.2 今後の課題

6.2.1 運用実験による評価

今回実現した支援環境は，一見の良さを目指したものではない．したがって，短期間の使用に基づく評価ではなく，実際にこの支援環境を運営することを通して，電子メールを用いるコミュニケーションにおける新たな知見を得る必要があると考える．

6.2.2 参加者によるアノテーション

今回実現した支援環境では，方針 2．参加者に対し，既存の環境における負荷以上の負荷を与えないに対する解として，参加者による明示的なアノテーションを許さず，代わりに言語的特徴を用いる方法を用いた．しかし，適度なアノテーションの許容は，抽出エンジンによる討議構造抽出精度の向上に貢献する可能性があるのは確かである．参加者によ

るアノテーションを許すことが円滑な討議に有効であるかどうか明らかにし、有効であるならば、その方法を確立する必要がある。

6.2.3 MIME マルチパートメッセージへの対応

MIME(Multipurpose Internet Mail Extensions) は、単純なテキストメッセージのやり取りである電子メールシステムを拡張して、任意の非テキストのファイルを添付できるようにするための仕組みである。ここ数年の間に、一気に普及した。今回実現した支援環境では、MIME マルチパートメッセージを特別扱いせず、添付されたファイルが無視する。しかし、添付されたファイルが、その議論のコンテキストに対して何らかの影響を与えることも考えられる。議論に対する添付されたファイルによる影響について考察し、もし影響を与えることがあるのであれば、それをふまえたシステムに改良する必要がある。

謝辞

終始変わらぬ御指導を賜りました落水浩一郎教授に心から深く感謝申し上げます。

村越広享助手には，本研究を進めるにあたり，随所で貴重なご意見を賜りました。深く感謝の意を表します。

また，研究の節目節目において，適切な助言を下さいました島津明教授，篠田陽一助教授に深く感謝致します。

最後に，落水篠田研究室所属諸氏の日頃の討論と助言，励ましに感謝致します。

参考文献

- [1] T. W. Malone, K. R. Grant, K. Lai, R. Rao, and D. Rosenblitt. Semi-structured messages are surprisingly useful for computer-supported coordination. In *Proceedings of the ACM 1986 Conference on Computer Supported Cooperative Work (CSCW'86)*, pp. 102–114. ACM, 1986.
- [2] Kum-Yew Lai and Thomas W. Malone. Object Lens: A “Spreadsheet” for Cooperative Work. In *CSCW 88 Proceedings*, pp. 115–124, 1988.
- [3] T. W. Malone, K. Lai, and C. Fry. Experiments with oval: A radically tailorble tool for cooperative work. In *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW'92)*, pp. 289–297. ACM, 1992.
- [4] T. W. Malone, K. Lai, and C. Fry. Experiments with oval: A radically tailorble tool for cooperative work. *ACM Transactions on Information Systems*, Vol. 13, No. 2, pp. 177–205, 1995.
- [5] F. Flores, M. Graves, B. Hartfield, and T. Winograd. Computer systems and the design of organizational interaction. *ACM Transactions on Information Systems*, Vol. 6, No. 2, pp. 153–172, 1988.
- [6] T. Winograd and F. Flores. *Understanding Computers and Cognition – A New Foundation for Design*. Addison-Wesley, 1987. [平賀譲 (訳), コンピュータと認知を理解する: 人工知能の限界と新しい設計理念, 産業図書, 1989].
- [7] S. M. Kaplan, A. M. Carroll, and K. J. MacGregor. Supporting collaborative process with conversationbuilder. In *Proceedings of ACM Conference on Organizational Computing Systems (COOCS'91)*, pp. 69–79. ACM, 1991.
- [8] S. M. Kaplan, W. J. Tolone, D. P. Bogia, and C. Bignoli. Flexible, active support for collaborative work with conversationbuilder. In *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW'92)*, pp. 378–385. ACM, 1992.
- [9] L. Bannon (Ed.). Commentaries and a response in the suchman-winograd debate. *Computer*

- Supported Cooperative Work*, Vol. 3, pp. 29–95, 1995.
- [10] Koichiro Ochimizu, Hiroyuki Murakoshi, Kazuhiro Fujieda, and Mitsunori Fujita. Sharing instability of a distributed cooperative work. In *International Symposium on Principles of Software Evolution(ISPSE 2000)*, pp. 33–42, Nov. 2000.
 - [11] Hiroyuki Murakoshi, Akira Shimazu, and Koichiro Ochimizu. Construction of Deliberation Structure in Email Communication. In *Pacific Association for Computational Linguistics(PACLING'99)*, pp. 16–28, Aug. 1999.
 - [12] Hiroyuki Murakoshi, Akira Shimazu, and Koichiro Ochimizu. A Method of Computing Measure for Evaluating Conversational Coherency in Email Communication. In *International Workshop on Collaboration and Mobile Computing(IWCMC'99)*, pp. 66–73. IEEE Computer Society Press, Sep. 1999.
 - [13] Herbert H. Clark and Edward F. Schaefer. Contributing to Discourse. *Cognitive Science*, Vol. 13, No. 2, pp. 259–294, 1989.
 - [14] Hiroyuki Murakoshi, Akira Shimazu, and Koichiro Ochimizu. Construction of deliberation structure in e-mail communication. *International Journal of Computational Intelligence*, Vol. 16, No. 4, pp. 570–577, 2000.
 - [15] 村越広享, 島津明, 落水浩一郎. メーリングリストを利用した共同作業における討議構造の自動構築法. コンピュータソフトウェア, 2001(accepted).
 - [16] D. G. Novick, L. Walton, and K. Ward. Contribution graphs in multiparty discourse. In *Proceedings of International Symposium on Spoken Dialogue*, pp. 53–56, 1996.
 - [17] 村越広享, 島津明, 落水浩一郎. メーリングリストを利用した共同作業における討議構造モデルの提案. 信学技報 SS2000-3, Vol. 100, No. 63, pp. 17–24, May 2000. ソフトウェアサイエンス.
 - [18] XML Linking Language (XLink) Version 1.0, Jul. 2000. <http://www.w3.org/TR/2000/CR-xlink-20000703/>.
 - [19] Rachel Reichman. Conversational Coherency. *Cognitive Science*, Vol. 2, No. 4, pp. 283–327, 1978.
 - [20] 泉子・K・メイナード. 会話分析. くろしお出版, 1993.
 - [21] M. A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, 1976.
 - [22] Tim Enders, Jeff Gay, and Y. Miyadate. ICEMail. <http://www.icemail.org/>.

本研究に関する発表論文

- [1] 山見太郎, 村越広享, 島津明, 落水浩一郎. 電子メールを利用したコミュニケーションにおける討議スレッド自動抽出法の実装と評価. 情報処理学会研究報告 2000-NL-137, Vol. 137, No. 10, pp. 69–76, June 2000.
- [2] 山見太郎, 村越広享, 島津明, 落水浩一郎. 討議スレッド参照機能を有するメールクライアント. 平成 12 年度電気関係学会北陸支部連合大会, Sep. 2000.
- [3] 山見太郎, 村越広享, 島津明, 落水浩一郎. ICEMail++: 討議構造参照機能を有するメールクライアント. インタラクション 2001, 東京, March 2001. 情報処理学会.
- [4] 山見太郎, 村越広享, 島津明, 落水浩一郎. 電子メールを用いたコミュニケーションを対象とした討議構造自動抽出エンジンの開発. 言語処理学会第 7 回年次大会 発表論文集, March. 2001.

付 録 A

UMML.dtd

```
1 <!ELEMENT umml (header, body)>
2 <!ATTLIST umml xmlns CDATA #FIXED "http://ochimizu-www.jaist.ac.jp/mailStructure">
3 <!ATTLIST umml xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink">
4
5 <!ELEMENT header (#PCDATA | From | Subject | Date | linkbaseRef)*>
6
7 <!ELEMENT From (#PCDATA)>
8
9 <!ELEMENT Subject (#PCDATA)>
10 <!ATTLIST Subject num CDATA #REQUIRED>
11
12 <!ELEMENT Date (#PCDATA)>
13 <!ATTLIST Date marked CDATA #REQUIRED>
14
15 <!ELEMENT linkbaseRef (#PCDATA)>
16 <!ATTLIST linkbaseRef xlink:role CDATA #IMPLIED>
17 <!ATTLIST linkbaseRef xlink:arcrole CDATA #FIXED "http://www.w3.org/1999/xlink/properties/linkbase">
18 <!ATTLIST linkbaseRef xlink:title CDATA #IMPLIED>
19 <!ATTLIST linkbaseRef xlink:show (new|replace|embed|other|none) "new">
20 <!ATTLIST linkbaseRef xlink:actuate (onLoad|onRequest|other|none) "onLoad">
21 <!ATTLIST linkbaseRef xlink:type (simple) "simple">
22 <!ATTLIST linkbaseRef xlink:href CDATA #IMPLIED>
23
24 <!ELEMENT body (#PCDATA | utterance)*>
25
26 <!ELEMENT utterance (#PCDATA | quote)*>
27 <!ATTLIST utterance type CDATA #IMPLIED>
28 <!ATTLIST utterance Ac CDATA #IMPLIED>
29 <!ATTLIST utterance Ac2 CDATA #IMPLIED>
```

30 <!ATTLIST utterance Pr ID #IMPLIED>
31 <!ATTLIST utterance whomPr CDATA #IMPLIED>
32 <!ATTLIST utterance clues CDATA #IMPLIED>
33 <!ATTLIST utterance flags CDATA #IMPLIED>
34
35 <!ELEMENT quote (#PCDATA)>

付 録 B

Linkbase.dtd

```
1 <!ELEMENT linkbase (link*)>
2 <!--!ATTLIST linkbase xmlns CDATA #FIXED "http://ochimizu-www.jaist.ac.jp/mailStructure/ properties /
   linkbase"-->
3 <!--!ATTLIST linkbase xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink"-->
4
5 <!ELEMENT link (locator+, arc+)>
6 <!--!ATTLIST link xlink: role CDATA #IMPLIED>
7 <!--!ATTLIST link xlink: title CDATA #IMPLIED>
8 <!--!ATTLIST link xlink:type ( extended) "extended"-->
9
10 <!ELEMENT locator EMPTY>
11 <!--!ATTLIST locator xlink: role CDATA #IMPLIED>
12 <!--!ATTLIST locator xlink: title CDATA #IMPLIED>
13 <!--!ATTLIST locator xlink: label NMTOKEN #IMPLIED>
14 <!--!ATTLIST locator xlink: type ( locator ) " locator "-->
15 <!--!ATTLIST locator xlink: href CDATA #REQUIRED>
16
17 <!ELEMENT arc EMPTY>
18 <!--!ATTLIST arc xlink: arcrole CDATA #IMPLIED>
19 <!--!ATTLIST arc xlink: title CDATA #IMPLIED>
20 <!--!ATTLIST arc xlink:show (new|replace|embed|other|none) "new">
21 <!--!ATTLIST arc xlink: actuate ( onLoad|onRequest|other|none) "onLoad">
22 <!--!ATTLIST arc xlink:type ( arc) " arc"-->
23 <!--!ATTLIST arc xlink:from NMTOKEN #IMPLIED>
24 <!--!ATTLIST arc xlink:to NMTOKEN #IMPLIED>
```
