

マルチモーダル対話システムに関する研究  
A Study on Multi-modal Conversation  
System

2003年3月

早稲田大学大学院 理工学研究科  
電気工学専攻 知覚情報システム研究

松坂 要佐

# 目次

第1章	序論	1
1.1	背景	1
1.2	マルチモーダル対話システムの概要とその類型	2
1.2.1	マルチモーダル対話システムの基本構成	2
1.2.2	対話する対象による類別	3
1.2.3	対話環境の実現方法による類別	6
1.2.4	本研究の立場と目的	7
1.3	マルチモーダル対話システムの実現における問題	9
1.3.1	マルチモーダル情報統合にかかわる神話と現状	9
1.3.2	対話チャネルの種類とその特性	10
1.3.3	対話のダイナミクスとモダリティ	11
1.3.4	本研究の立場と目的	13
1.4	本論文の構成	13
第2章	情報共有アーキテクチャ	18
2.1	はじめに	18
2.2	接続アーキテクチャの検討	19
2.2.1	従来研究	19
2.2.2	本研究の立場と目的	20
2.3	モジュール間接続アーキテクチャの検討	24
2.3.1	情報公開・メッセージ通知モデル	24
2.3.2	プロセッサ間の接続方法	24
2.3.3	処理の粒度の異なるモジュールの統合	26
2.4	アーキテクチャの実装	26
2.4.1	システム構成	26
2.4.2	オンボードバス	28
2.4.3	ネットワークサーバ・クライアント	29
2.4.4	バスブリッジ	31
2.4.5	タグのグループ購読	33
2.4.6	開発ツール	33
2.4.7	黒板のデータ構造と操作コマンド	33

2.5	システムの稼動状況	35
2.6	議論	39
2.7	まとめ	40
<b>第3章</b>	<b>基本ソフトウェアと開発環境</b>	<b>42</b>
3.1	はじめに	42
3.2	開発における問題点と開発環境への要件	43
3.3	機能モジュールの開発環境	44
3.3.1	バザールのシステム開発の実現	44
3.3.2	基本構成・基本的デザインプロセス	44
3.4	インタラクティブシステムの実現	46
3.4.1	自律性・応答性の実現	46
3.4.2	即応的なモジュール選択と動作のマッピング	47
3.5	システムアーキテクチャの実装	47
3.5.1	情報公開・メッセージ通知サーバ	47
3.5.2	タグの命名規則・グループ購読	49
3.5.3	モジュールの優先度管理機構	49
3.5.4	状況判断機構	50
3.5.5	動作のマッピング機構	50
3.6	システムの運用	51
3.6.1	システム構成	51
3.6.2	状況切替の動作例	53
3.6.3	システムの稼動状況	53
3.7	議論	56
3.8	まとめ	58
<b>第4章</b>	<b>グループ会話に参加する対話ロボットの構築</b>	<b>60</b>
4.1	はじめに	60
4.2	用語の定義：グループ会話と会話参加者の役	61
4.3	グループ会話の特徴	63
4.3.1	会話状況の曖昧性とその理解・明確化の必要性	63
4.3.2	発話の広報性と当事者間対話の理解に基づく発話の必要性	64
4.3.3	アウェアネスの不均衡性とその改善努力の必要性	65
4.4	システムの設計	65
4.4.1	実ロボットの身体表現を用いた会話状況の明確化	65
4.4.2	マルチモーダルな情報処理による会話状況の理解	67
4.4.3	当事者対話の理解に基づく従たる聴者の割込み発話	68
4.4.4	アウェアネスの改善	68
4.4.5	システムの振舞い	68

4.5	システムの概要	70
4.5.1	ハードウェア	70
4.5.2	ソフトウェアモジュール構成	70
4.6	システムの動作	73
4.7	議論	76
4.8	まとめ	79
第 5 章	総論	80
	参考文献	83
	関連する研究業績	90
	謝辞	95

# 第1章 序論

## 1.1 背景

近年，高精度の音声認識技術の実用化や，画像処理技術の発展，計算機の性能向上を背景としたそれら技術の応用範囲の拡大により，受付応答システムや対話ロボットに代表されるマルチモーダル対話システムに関する研究が盛んになってきている．マルチモーダル対話システムは，従来の音声のみを用いた対話システムと比較して，音声情報・視覚情報など複数の情報交換チャネルを扱うことのできる入出力インターフェースを持ち，ユーザフレンドリーなシステムの実現に貢献する．また，従来用いられなかった視覚情報のチャネルを用いることで，音声情報の交換のみでは実現不可能であった様々な形態の対話に対処することが可能となる．

現在，マルチモーダル対話システムの開発が試みられているが，これらシステムの開発には，現状においては，以下のような問題がある．従来の音声対話システムと比較してシステムが複雑化し，それが開発効率の悪化やシステムの不安定性の原因となる．複数のチャネルにまたがる情報の統合的な利用や，複雑化する対話タスクに対処する必要がある，開発されたシステムを統合・制御する枠組みが必要となる．従来用いられなかった視覚情報という情報交換チャネルが対話のいかなる局面において役に立つか分析・整理されていない．本研究においては，これら問題を解決し，実用的なマルチモーダル対話システムを実現するために，それに必要な機能の実現とシステムとしての統合を容易化するハードウェアアーキテクチャを検討し，複数のチャネルとタスクにまたがる情報の統合の容易化と即時的な行動の発現を実現するシステムアーキテクチャを検討するとともに，応用としてのグループ会話システムを実現する．

## 1.2 マルチモーダル対話システムの概要とその類型

### 1.2.1 マルチモーダル対話システムの基本構成

対話システムとは、人と情報のやり取りを行い、そのやり取りの中で問題を解決する役割を持った計算機システムのことを指す。一般的な計算機システムと比較した場合、対話システムは、人を相手にするための入出力インターフェースを持つ、人とのやり取りの中で問題解決するための戦略を持つ、などの特徴がある。

近年利用されている多くの計算機システムは、入出力インターフェースとしてキーボードとマウス、ディスプレイをもち、人間と対話的にやり取りするアプリケーションプログラムを搭載しており、広い意味では対話システムといえることができる。しかしながら、これらの広い意味での対話システムを人間が使うことを考えた場合、キーボードをたたいたりマウスをクリックするなど、通常人間同士で行うやり取りとは異なる方法で、システムとやり取りしなければならない。本稿では、上記のような広い意味での対話システムに対して、より人間に近いインターフェースをもち、人にとって自然な感覚での対話の実現を目指したシステムのことを指して、特に対話システムと呼ぶことにする。

対話システムを構成する各要素を図 1.1 に示す。対話システムは、入力インターフェース、内部システム、出力インターフェースからなる。入出力インターフェースは複数あってもよく、各々の入出力対の間をつなぐ伝送路をチャンネルと呼ぶ。

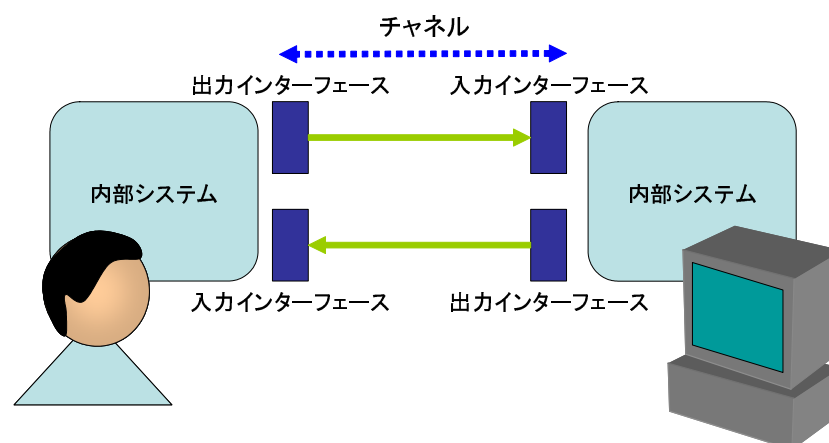


図 1.1: 対話システムの構成要素

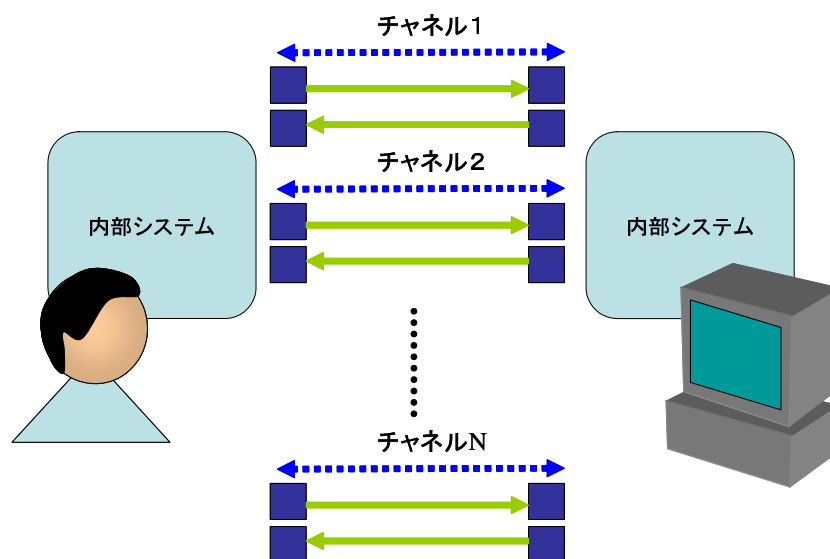


図 1.2: マルチモーダル対話システム

人間にとってもっとも自然な対話の方法は、音声言語を用いた対話である。それゆえに、音声対話システムの実現は、古くから熱心に研究されてきた。音声対話システムは、音声言語のチャンネルにおける情報交換を扱うために、入力インターフェースとして音声認識器、出力インターフェースとして音声合成器を持つ。

人間同士の対話においては、音声言語以外にも、さまざまなチャンネルを用いて相互に情報交換している。対話システムの研究においても、音声言語以外のチャンネルを用いて情報交換をするために、複数のチャンネルを扱うことができる入出力インターフェースを備えた対話システムが検討されている。これらのシステムは、音声対話システムに対して、マルチモーダル対話システムと呼ばれる (図 1.2)。

### 1.2.2 対話する対象による類別

マルチモーダル対話システムは、想定するユーザとの対話する対象により、以下のように類別することができる。

- 道具との対話
- 環境との対話
  - － 環境化

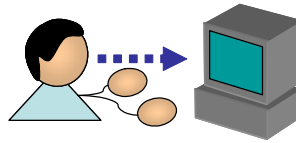
- － 一体化
- 擬人化された対象との対話
  - － 部分擬人化
  - － 完全擬人化

上記類別を対話の自然性と円滑性の観点から分析する．ここで，対話の自然性とは，人と人の対話にどれだけ近い形態での対話を実現できるかの程度を指す．また，対話の円滑性とは，目的とする情報の伝達において，伝達を達成するために，どれだけのやり取りを必要とするかの程度を指す．自然性の高い対話は，人と人との対話に近い対話であるから，円滑性もある程度の高さであることが期待できる．円滑性の高い対話は，少ないやり取りで情報を伝達できる対話であるが，その実現は人と人との対話に習う必要はなく，必ずしも自然性も高くなるとはいえない．

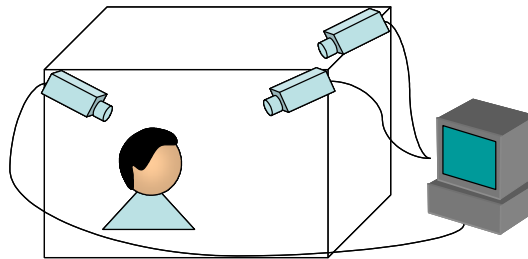
道具との対話とは，ユーザがシステムを使うに当たり，システムを道具と意識して行う対話である．この類別の対話においては，ユーザ側からシステムにあわせて対話を行うことが許されるため，対話の自然性はさして重要な要件ではなくなる．システムのマルチモーダル化により，一度のやり取りで伝達される情報を増加させることで，対話の円滑性が向上することが期待される．マウスやキーボードによる対話がこれに含まれる．

環境との対話とは，システムがユーザを観察する形態で行われる対話である．ユーザの発話や動作を入力として，その行為を観察することによって情報を検出するシステムを指す．この類別の対話においては，ユーザにシステムを対話する対象として意識させる場合と，意識させないことを理想とする場合の2つがある．前者においては，上記した道具との対話と同じく，対話の自然性はさして重要ではなく，円滑性を向上させることが重要な要件となる．後者においては，システムの存在を意識しないで行ったユーザの自然な発話や行動から情報を検出することが必要とされるため，対話の自然性の向上も重要な要件となる．システムの実現方法としては，インテリジェントルームなどのようなシステムを環境化するアプローチや，ウェアラブルコンピュータのようなシステムとユーザが一体化するアプローチがある．





(a) 道具との対話



(b) 環境との対話



(c) 擬人化された対象との対話

図 1.3: マルチモーダル対話システムの対話する対象による類別

擬人化された対象との対話とは，ユーザの発話や動作を入力とするだけでなく，システム側からもユーザにとって自然な形で発話や動作を出力する，双方向の表現による対話を指す．人間であるユーザにとって自然に認識できる発話や動作は，人間のする行為に近い動作であると考えられるため，対話システム側の表現にも人を模擬した表現が用いられることが多い．人の表現の重要な要素のみを抽出して表現する部分擬人化のアプローチと，完全な人の表現を実現しようとする完全擬人化のアプローチがある．

人にとっての対話の自然性の観点から見ると，道具との対話・環境との対話よりも，擬人化された対象との対話のアプローチの方が自然性が高い．なぜなら道具との対話・環境との対話は，人が通常体験しない形での対話だからである．道具は，人に対して人のするそれとは異なった方法で語りかける．また，環境は，人が対話の相手として捕らえるには大きすぎる対象であり，多くの場合対話者に違

和感を与える．それらの対話環境におかれた対話者は，新しい対話のあり方を訓練を通して獲得する必要がある．それに対して，本研究の対象でもある擬人化のアプローチを用いた対象との対話には，対話の高い自然性を実現する可能性がある．ただし，真の意味での対話の自然性を実現するためには，多くの検討が必要である．それら検討の必要性については，[1.3.1節](#)で議論する．

### 1.2.3 対話環境の実現方法による類別

対話環境の実現方法として，下記の3種類のアプローチがある．コンピュータ上に仮想世界を構築し，人をその世界に没入させる仮想化のアプローチ．コンピュータ上に仮想世界を構築し，ディスプレイを通じて人にその世界を提示する仮想オブジェクト化のアプローチ．実世界上に実体を持ったオブジェクトを構成する実世界志向のアプローチ．

仮想化のアプローチは，人を取り囲む世界全体を操作できるメリットがあるが，その一方で仮想世界への没入感を与える装置を装備させる必要があるため，人に対して負担が大きいデメリットがある．

仮想オブジェクト化のアプローチは，オブジェクトの自由な操作と人に対する負担の少なさの両方のメリットを持つが，その一方で2次元上のディスプレイによる提示には3次元空間上での表現力が足りないというデメリットがある．

実世界志向のアプローチは，人に対する負担が少なくかつ高い表現力を持ったシステムを実現できるメリットがあるが，その一方でオブジェクトを部材とアクチュエータによって物理的に構成・制御する必要があるため，対象の複雑な操作を実現しにくいというデメリットがある．

マルチモーダル対話システムにおいては，人に対する情報を提示する出力インターフェースとともに，人の表現する情報を取得するための入力インターフェースを実現する必要がある．仮想化のアプローチにおいては，人に対して仮想世界に没入させる装置と同時に，挙動を検出するセンサーを取り付けることが多い．仮想オブジェクト化や実世界志向のアプローチにおいては，対話の自然性を重視する観点から，画像処理などの非接触の方法で人の挙動を検知することが求められる．

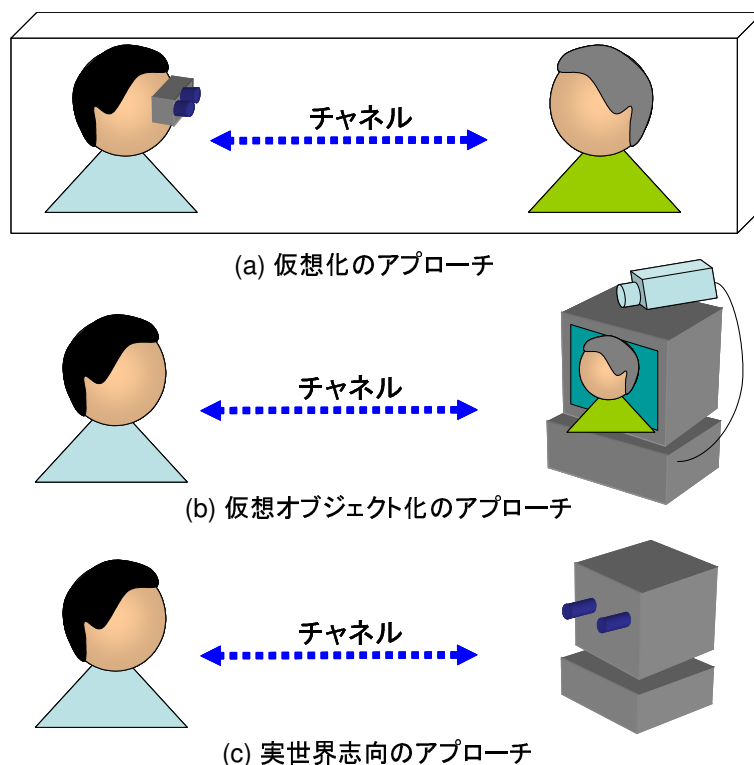


図 1.4: 対話環境の実現方法による類別

#### 1.2.4 本研究の立場と目的

本研究では，マルチモーダル対話システムの構築にあたって，擬人化された対話の対象を実世界に実現するアプローチを採る．擬人化・実世界志向の対話システムとして，人間の形状・機能を模擬したロボットであるヒューマノイドロボット [Kat91] を研究の対象とする．

本研究で用いたヒューマノイドロボット ROBITA (Real-world Oriented BImodal Talking Agent: 図 1.5) は，多くのヒューマノイドロボットに備わっている 2 足歩行機能を省略した上半身型のロボットであり，人間との対話インターフェースの研究のために開発された研究プラットフォームである．

本研究は，この研究プラットフォーム上に，音声・音響・画像処理などの複数メディアを扱うことのできる入出力インターフェースを実装し，人間との自然性の高い対話を実現することのできる，マルチモーダル対話システムを実現することを目的とする．

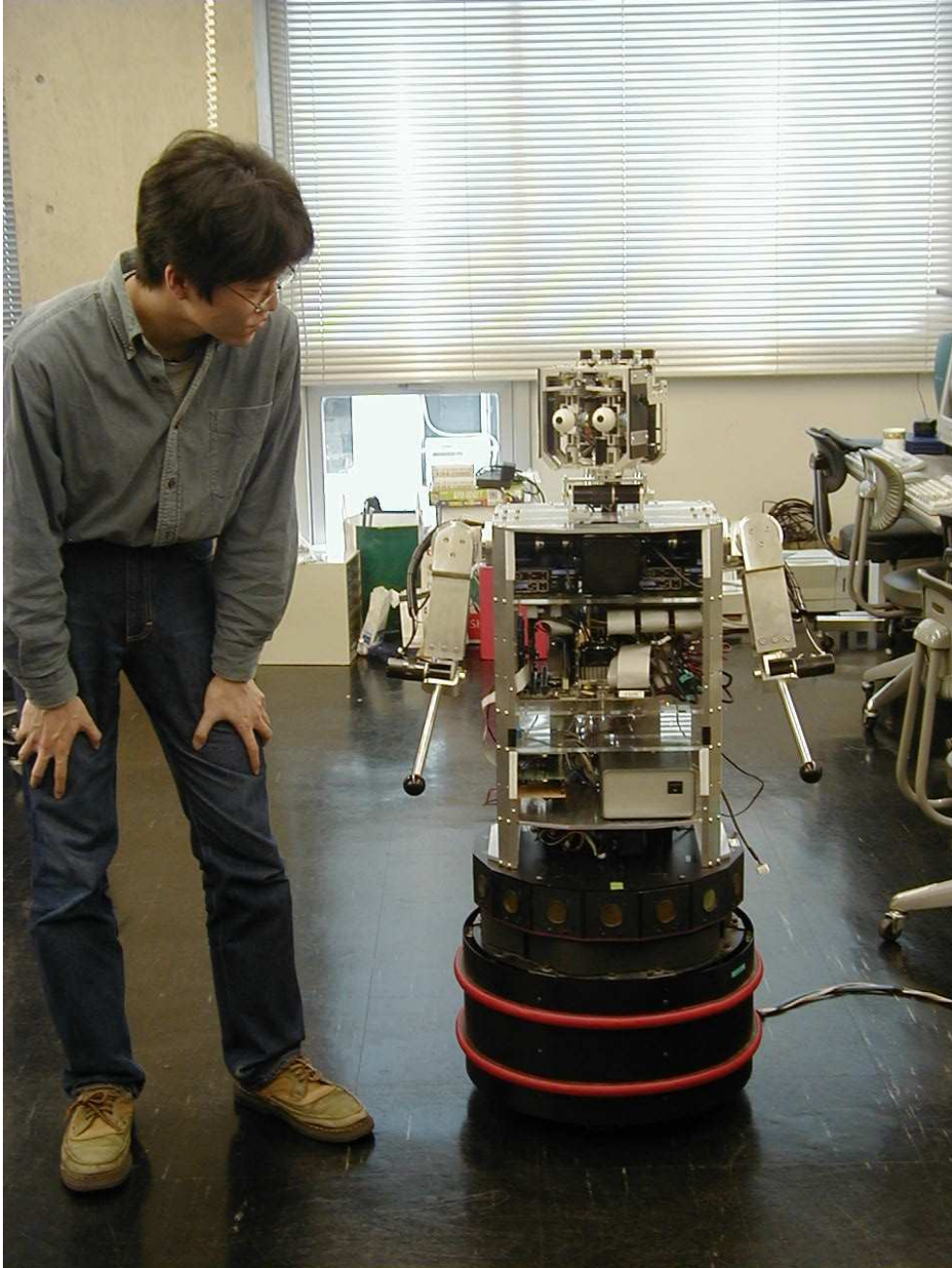


図 1.5: ROBITA(Real-world Oriented BImodal Talking Agent)

ヒューマノイドロボット研究は、実用システムの開発という側面のほかに、人間研究としての側面も持っている。人型のロボットを作ろうという試みは、人の形状を模擬するだけでなく、人の機能も模擬することを必要とする。人型のロボットのコミュニケーション機能の実現を目的とする本研究は、人間のコミュニケーション機能の解明に関する研究としての側面も持つ。

## 1.3 マルチモーダル対話システムの実現における問題

### 1.3.1 マルチモーダル情報統合にかかわる神話と現状

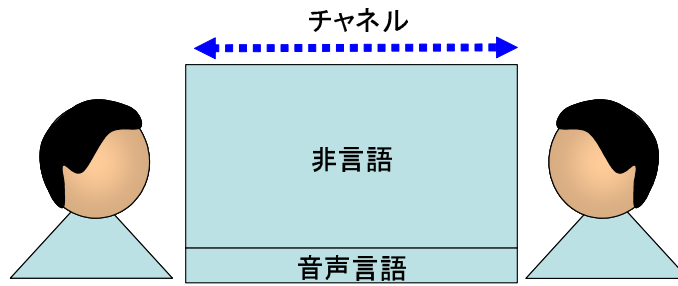
人間同士のコミュニケーションにおいては、音声言語チャンネル以外にも、さまざまなチャンネルを通して情報交換が行われている。音声言語チャンネル以外のチャンネルによる情報交換は、比率にして 90 パーセントを超えともいわれ [\[Meh68\]](#)、それらデータがマルチモーダル対話システムに対する期待を高いものとしている。

しかしながら、現状のマルチモーダル対話システムにおいて扱うことのできている情報交換チャンネルはわずかであり、ユーザと対話システムの間には、大きな不均衡性が存在する (図 1.6)。ユーザはシステムからの出力をその入力チャンネルすべてを通して認識するのに対して、システム側の入力インターフェースは限られており、ユーザからの出力はシステムの限定されたチャンネルを通してしか受理されない。ユーザと対話システムの間の不均衡性は、しばしば下記の 2 点において問題になる。

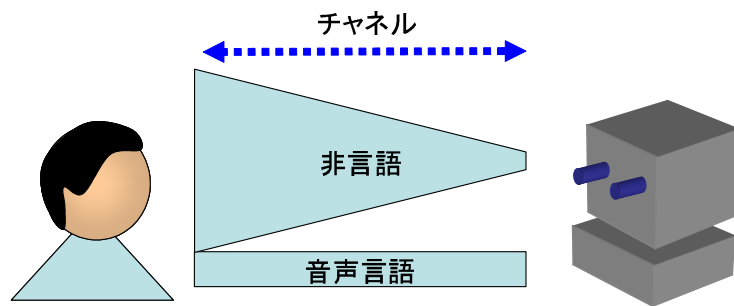
第一に、対話の円滑性 (定義については、[1.2.2 節参照](#)) が妨げられる問題がある。ユーザとシステムの間には狭いチャンネルしか存在しない。その狭いチャンネルのみを利用して対話を進行させなければならないため、高い対話の円滑性を実現することは難しい。

第二に、対話の自然性 ([1.2.2 節参照](#)) が妨げられる問題がある。人と人の対話においては通常利用されるチャンネルのいくつかが存在しないことによって、本来ユーザの自然な行動呈示から受理されるべき情報が検知されなくなる。そのような場合、ユーザは、本人にとっては不自然な代替のチャンネルを利用して情報を再提示する必要がある。

マルチモーダル対話システムは、上記の 2 点の問題の改善に貢献する。利用で



(a) 人間同士の対話におけるチャンネルの利用



(b) 人間・対話システム間におけるチャンネルの利用の不均衡性

図 1.6: 対話におけるチャンネルの利用とその不均衡性

きるチャンネル数が増えることによって，対話の円滑性の向上を期待することができる．また，人間同士の対話において本来使われる表現を受理できるチャンネルを持つことによって，対話の自然性の向上が期待できる．

現状においてチャンネルの利用の不均衡性は非常に大きなものである．その不均衡性を解消するために，求められる入出力インターフェースのすべてを実現することは，現状の技術では難しい．マルチモーダル対話システムの実現を考えた場合，対話の円滑性・自然性を効率よく向上させるチャンネルを十分に検討し，それを扱うことのできるインターフェースを増設することが望まれる．

### 1.3.2 対話チャンネルの種類とその特性

人間同士のコミュニケーションにおいて利用される代表的な対話チャンネルには表 1.1 のような種類がある．

音声言語とは，音声として提示された言語的な情報を指す．周辺言語とは，音声から言語的な情報を除いたすべての情報を指す．これには，声の強さ・速さ・高



表 1.1: 代表的な対話チャンネル

音声言語	
周辺言語	韻律, 非言語 (感動子, 相槌)
身体動作	顔表情, 頭部動作 (うなずき, かしげ, 首振り) 視線, 指差し・ジェスチャ

さからなる韻律情報や、言語としての体をなしていない非言語音声などが当てはまる。身体動作とは、主として頭部や腕部の動きによって相手に提示される情報である。

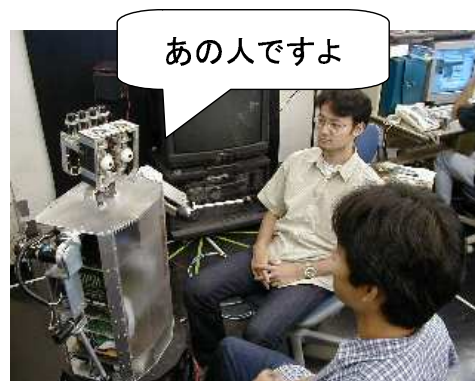
音声というメディアを通した情報の提示と、身体動作という映像をメディアとした情報の提示を比較すると、そこには表 1.2 のような性質の差異がある。音声メディアは、ある一点で発せられた後には、空間上に一様に拡散していく、時間軸上で音声区間同士が重なるとその音声は著しく聞き取りにくくなってしまうため、ターンテイクすることでチャンネルを時間分割して利用する必要がある。それに対して、映像メディアは、ある一点で表現された後、空間上に方向依存性を持って伝播される (図 1.7)。映像メディアは、3 次元的な情報の 2 次元空間上への射影として受理され、それがどのように受理されるかは 2 点間の関係で決まる。空間的に分割されて表現されるため、時間軸上で表現が重なったとしても一方の表現が他方の表現によって妨害されることはない。これらメディアによる特質の差は、マルチモーダル対話システムの実現を考える上で重要な検討事項となる。

表 1.2: 伝送メディアによる性質の差異

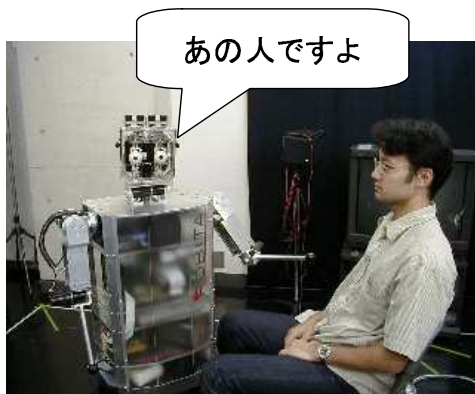
	ターンテイク	広がり
音声	必要 (時間分割)	一様拡散
映像	不要 (空間分割)	方向依存

### 1.3.3 対話のダイナミクスとモダリティ

人間同士の対話においては、参加者はチャンネルを通じて意図を伝達しあい、相互に調整しあいながら対話を進めている。人間の認知のプロセスは、複数の認知



(a) 対話の前景(手前の話者を話者1、奥の話者を話者2とする)



(b) 話者1から見た対話の風景



(c) 話者2から見た対話の風景

図 1.7: 方向依存性を持って伝達される映像メディアの性質



モジュールが並行して動作し、それが状況に応じて切り替わっていくダイナミクスを持っている。そのダイナミクスは、対話においても成り立ち、人間同士の対話は、その状況によって複数の意図・調整の機構が切り替わっていくダイナミックなものである。

対話におけるモダリティ(チャンネル)の利用においては、複数チャンネルを同時に利用する場合もあれば、複数ある中から単一のチャンネルを選択して利用する場合もある。チャンネルの利用もまた、時と状況に応じて切り替わっていくダイナミックなものである。これらダイナミクスの実現は、マルチモーダル対話システムを実現する上で、重要な要素となる。

#### 1.3.4 本研究の立場と目的

マルチモーダル対話システムを情報処理システムとしての側面から見ると、それは従来システムに比較して、新たに大量の情報を扱う必要があるシステムである。映像情報などは特に巨大な情報量を持つため、それらを効率的に扱っていく枠組みを検討する必要がある。

マルチモーダル対話システムを情報統合システムとしての側面から見ると、それは、音声・音響・映像・身体制御などの多岐に渡る情報を統合する複雑な統合システムであると同時に、入力に対して即時的な応答を返すことができる柔軟性も兼ね備える必要がある。

マルチモーダル対話システムを対話システムとしての側面から見ると、それは対話の円滑性と自然性に貢献する新たなチャンネルを備えた対話システムである。

本論文は、これらマルチモーダル情報統合を実現する上での諸技術を包括的に扱う。本論文においては、擬人化された実世界志向の対話システムとしてのヒューマノイドロボットを対象として、マルチモーダル対話システム実現にかかわる技術の検討、実装を行う。

### 1.4 本論文の構成

本論文は、1章導入、2章マルチモーダル対話システム構築のためのハードウェアアーキテクチャの設計と実装、3章機能のスムーズな開発と即時的な行動選択を

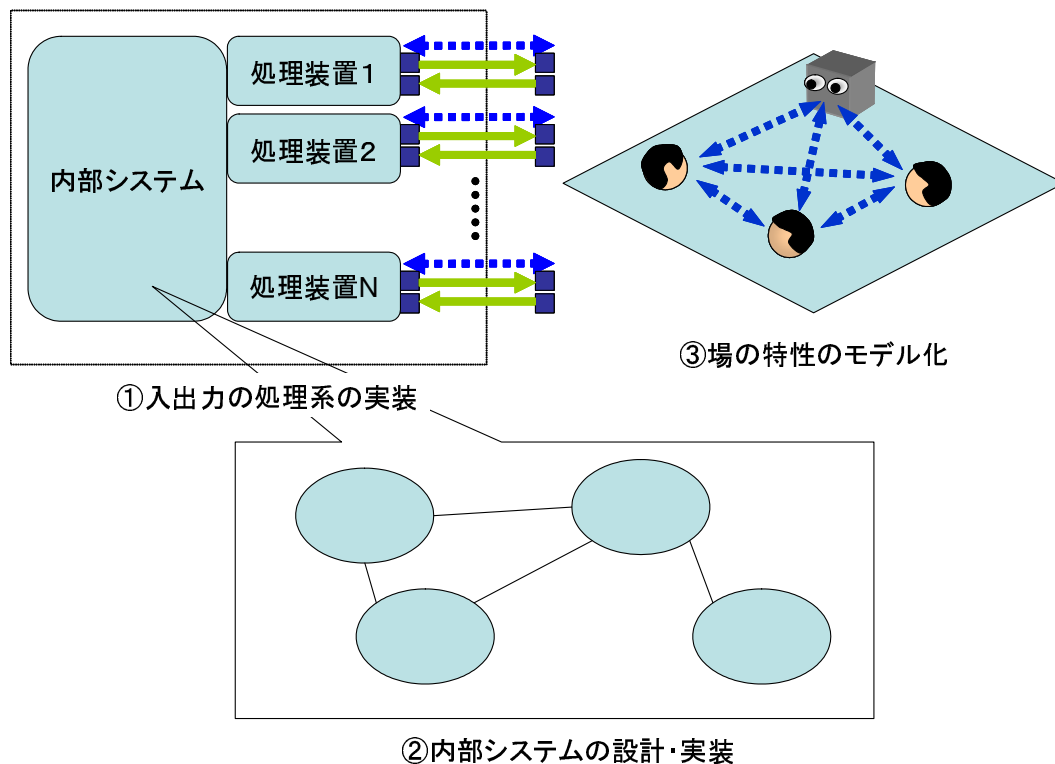


図 1.8: マルチモーダル対話システムの実現における検討事項

実現するシステムアーキテクチャの設計と実装，4章グループ会話に参加する対話ロボットの構築，5章まとめにより構成される．

2章においては，多機能システムの開発において必要となる，モジュール間接続アーキテクチャの設計と実装について述べる．

多機能システムの開発においては，要求される機能を分割した上で個別に開発するためのモジュール化が重要な要件となる．近年，マルチモーダルシステムの開発など，システム開発において要求される機能が膨大になり，システム開発は大規模化している．それに従い，システム開発の過程におけるモジュール化の要求が高まっている．システムをモジュールに分割し，再度統合するためには，モジュールとモジュールを接続する技術が必要となる．

モジュール間接続には，用途に応じて様々な方法が考えられる．接続法には向き不向きがあり，効率のよい動作スタイルや開発スタイルは，接続法によって決まってしまう．接続法の検討はシステム開発において非常に重要な要素である．本章では，様々な形態が考えられる接続法の中で，マルチモーダルシステム開発に

適した接続法を考える。

マルチモーダルシステムのアプリケーションは、一般的なシステム開発と比較して下記の点において特殊性を持つ。マルチモーダルシステムの例として、人との対話を行ないながら生活空間で活動するロボットを考えた場合、そのシステムは、人の生活空間で生起する様々な事象に対応するために、様々なタスクをダイナミックに切り替えながら活動していかなければならない。また、システムは、対話処理からモータ制御まで、様々な粒度にまたがる処理をする必要がある。システムが、ユーザからの要求や生起する事象に応えながらも、タスクを遂行し達成するためには、応答性と自律性の二つの要件を同時に満たす必要がある。

本章では、これらの要件を満たす接続アーキテクチャとして、モジュール同士のフラットな関係とモジュール自身による情報の流れのダイナミックな切替えを実現する黑板・publish/subscribeの複合サービス、密な統合とスケーラビリティの確保を両立できることを特徴とする2層のバスアーキテクチャを設計し実装する。システムに音声・音響・画像情報を統合的に利用した対話タスクを実装し、高密度に統合されたシステムが短期間で開発できることを確認した。

3章においては、インタラクティブシステムの共同開発に適した開発環境の設計・実装について述べる。

大規模統合システムの開発においては、複数の分野にまたがる専門知識が必要とされるため、単一の開発者がシステム全体を見通すことは困難となる。そのため、異なる専門を持った複数の開発者たちの共同作業が不可欠となり、それを円滑化させる枠組が必要となる。

これまでインタラクティブシステムの開発においては、開発がトップダウン的に行われることが主流であった。しかしながら、日常の雑多な作業へ対応するシステムへ求められる機能は多様であり、時とともに変化する可能性が高いことを考慮すると、システムデザイナーがあらかじめシステムの完成像を見通すことは難しく、従来のトップダウンの開発形態でユーザの要求に応えていくことは難しい。それらシステムの開発形態としては、タスク・機能を各々のモジュール開発者が各々の問題解決手段に基づいて分析・実装し、それら実装された機能を必要に応じて追加していけることが望ましい。また、システムが実世界で生起する様々なイベントに対処しながら行動するためには、その状況に適した行動が即時に選

ばれる必要がある．複数の開発者によって開発された複数のモジュールが互いに協調し，迅速に行動が決定される枠組みを用意する必要がある．

本章では，マルチモーダルシステムを対象とした大規模統合システムの開発を容易化するために，ボトムアップ的なシステム開発と迅速なモジュール間協調を可能とするシステムアーキテクチャを検討する．本章においては，1) 黒板モデルに基づく情報共有によるモジュール開発とスクリプトに基づくモジュール間優先度管理，2) メッセージ配信機能を拡張した黒板型の情報共有機能と CVS に基づく共有情報管理，3) 状況判断機構とスクリプト記述による即時的な行動選択により，機能モジュールによるシステム開発と即時的な行動生成を実現することができることを特徴とするシステムアーキテクチャを検討した．そのアーキテクチャを実現する基礎となる基本ソフトウェア，開発者の活動を支援するツールを実現し，音声・音響・画像情報を統合的に利用した対話ロボットの実装を通して，高密度に統合されたシステムが短期間で開発できることを確認した．

4章においては，グループ会話に参加できる対話システムの開発について述べる．

通常我々は複数人と共に話題を共有しながら多対多の会話（以下，グループ会話と呼ぶ）を行なっている．我々がこのようなグループでの情報交換を通じて，意思決定をしたり着想を得たりする機会は多く，グループ会話は人間同士のコミュニケーション形態として非常に重要性が高い．このため，複数の人と活動を共にするロボットシステム，あるいはグループ形態での人間の知的活動を支援する情報システムのように，人間同士のコミュニケーションに加わるシステムを実現しようとするならば，システムはグループ会話に参加できる能力を備える必要がある．

グループ会話に参加可能なシステムを実現するためには，従来の対話システムにおいては無視できたさまざまな問題を扱わなければならない．対話システムにおいては，個々の局面において各会話参加者がどのような役（発話者，聴者など）で対話に臨んでいるかといった対話の状況をシステムが理解するとともに，次の局面でそれがどのように変化をするかを適切に予想しながら会話を進行することが必要である．しかし，システムがただ一人のユーザを相手にする場合や，2人のユーザの対話をモニタする立場に徹する場合などでは，これらは発話者を手がかりとして一意に決めることができた．これに対し，グループ会話では観察者なる役が生じることで会話参加者間の関係は複雑化し，対話の状況は自明ではなく

なる．円滑に会話を進行させるためには，システムは正確に対話の状況を理解できるとともに，自らも状況の曖昧性を減少させる適切な行動をとることが必要となる．また，観察者である対話参加者が将来発話者として会話に参加する機会を得るためには，会話に対する協調的な態度を積極的に示すことで，現発話者の注意を喚起する必要がある．これらの問題は会話システムにおいて本来重要な課題であるべきにも関わらず，従来それが問題として現れない特殊な問題設定において対話研究が進められてきたことによって，十分な検討がなされて来なかった．

本章では，グループ会話の特徴の分析に基づいて，円滑なグループ会話を実現するために必要なシステムの振舞いについて検討し，さらに，マルチモーダルな情報処理機能を持つ人間型ロボットを用いて，グループ会話に参加できるシステムを実現した．ロボットは会話状況に関する判断能力を持つことで，望まれるタイミングで自然な応答をすることができた．また，ロボットが会話状態の明確化のために行なう身体表現の効果により，会話参加者も戸惑うことなくグループ会話を進めることができた．ロボットには，単に聞かれたことに答えるだけでなく，ユーザ同士の対話に割込んで発話を行なう機能も持たせた．また，ロボットが観察者の役に回っているときでも，アウェアネス改善に向けて対話当事者に働きかける機能を持たせた．これらのことにより，グループとしての会話を活性化することができた．

5章においては，上記の章における議論を総括し，マルチモーダル対話システムの開発に関しての残された問題を展望する．

## 第2章 情報共有アーキテクチャ

### 2.1 はじめに

多機能システム開発においては、要求される機能を分割した上で個別に開発するためのモジュール化が重要な要件となる。近年、マルチモーダル対話機能を持ったロボットの開発など、システム開発が多機能化・大規模化しており[橋本 97]、モジュール化への要請が高まりつつある。システムをモジュールに分割し、再度統合するためには、モジュールとモジュールを接続する技術が必要となる。

モジュール間接続には、用途に応じて様々な方法が考えられる[Tan95]。接続法には向き不向きがあり、効率のよい動作スタイルや開発スタイルは、接続法によって決まってしまう。接続法の検討はシステム開発において非常に重要な要素である。本章では、マルチモーダルシステムの中でも、実時間処理の必要性等の観点からとりわけ困難な課題を有するマルチモーダル対話ロボットをとりあげ、様々な形態が考えられる接続法の中で、それらシステムの開発に適した接続法を考える。

マルチモーダルシステムのアプリケーションは、一般的なシステム開発と比較して下記の点において特殊性を持つ。生活空間で活動する多機能ロボットを考えた場合、生起する様々な事象に対応するための、様々なタスクをダイナミックに切り替えながら活動していかなければならない。人との対話を行なうロボットを考えた場合、そのロボットは、対話処理からモータ制御まで、様々な粒度にまたがる処理をする必要がある。システムが、ユーザからの要求や生起する事象に応えながらも、タスクを遂行し達成するためには、応答性と自律性の二つの要件を同時に満たす必要がある。

また、マルチモーダルシステム開発におけるモジュールの取り扱いは、従来のソフトウェア開発において検討されてきたことに加えて、以下の点を検討する必要がある。システムの処理性能とリアルタイム仕様を考える必要から、モジュールをその機能を実現するソフトウェアだけでなく、その機能を動作させるハード

ウェア込みで考える必要がある．システム内部に搭載することを考えると，設置場所に対する制限がある．モジュール自身の大きさのみならず，その接続アーキテクチャに対する制約が生まれる．

本章では，これらの要件を満たす接続アーキテクチャを検討・設計し，実装する．

## 2.2 接続アーキテクチャの検討

### 2.2.1 従来研究

ロボットを対象としたアプリケーションにおいて，モジュール間通信技術を実装した研究がいくつかある．ASPIRE においては，ロボット上に VME バス接続された分散共有メモリを配置し，割り込み線を多数張りめぐらせることで，応答性の高い協調動作を行なう機能分割されたマルチプロセッサシステムが実装された [山崎 96]．また，Remote Brain においては，ロボット上のトランスピュータと，無線接続されたワークステーションによる階層構造を持つ分散システムとして実装された [Ina94]．Jijo-II においては，イベント処理機構を持つオブジェクト指向 Lisp の上に，オフィスサービスを行なうアプリケーションが実装された [MAHO97]．ISAC においては，COM が提供する環境の上にロボット用のモジュールを実装することが試みられた [PWGK99]．CORBA が提供する環境の上にモジュールを実装する試みも多く行なわれている [YKK+01]．一般的な OS に比較して，高いリアルタイム性能を持つ RT-Linux の提供する環境上にシステムを実装した例も数多く見られる [KNK+01]．HERMES [BJ99] や他の多くのロボットの実装において，ロボットの内に張られた CAN バスや I2C バス上にマルチプロセッサシステムを実装することが試みられている．

ロボットを対象としたアプリケーション以外にも，従来，システムの構成要素をモジュール化，コンポーネント化することについて数多くの研究がなされてきた．分散されたモジュール同士をつなぐ道具として，ORB 技術が発展した．CORBA [cor] や COM [com] においては，インターフェースを統一言語で記述し，ネットワークプロトコルにマッピングすることで，モジュールのネットワーク分散化，多言語対応を実現した．OAA においては，利用モジュールの選択を支援するシステムサービスを提供することで，ユーザへの多インターフェースサービスを実現した [oaa]．

MUSE においては、複数モジュールの構成において、階層化されたメッセージパッシングサービスを提供することが試みられた [YTM<sup>+</sup>91]。複数エージェントの協調や分散知識処理を行なうシステムにおいては、同報通知型の通信に基づく通信アーキテクチャが検討された [Smi80]。Multi-PSI においては、ネットワーク上に分散されたプロセッサと、同報通知機能を持つネットワークプロトコルの上に、分散知識処理システムが作られた [IMT87]。

### 2.2.2 本研究の立場と目的

従来研究と比較した、本研究の立場を述べる。従来のシステム開発における接続アーキテクチャに関する研究は、情報の流れの構造が決まっているところに、如何に効率よくモジュール同士の接続を実現するかが議論の主題にされてきた。しかしながら、必要とされるリソースや、要求される応答性能、行う作業は、タスクによって大きく異なる。モジュールの間に形成される最適な情報の流れの構造は唯一ではなく、タスク毎に存在すると考えられる。多くの機能を持ち、多くのタスクを効率的にこなせるシステムを考えると、タスク毎に存在する最適な情報の流れの構造を切り替えながら動作できることが望ましい。そのようなシステムを作るためには、与えられた特定の情報の流れの構造の中で効率的な動作を実現するだけでなく、タスクに応じて柔軟に情報の流れの構造を変化させていくことのできる接続アーキテクチャについての議論が必要である。

柔軟な構成の変更を特徴とするマルチエージェントシステムの研究において接続アーキテクチャについての議論がなされている [oaa][Smi80] が、多くのシステムが応答性能に関して比較的制約の少ないネットワークエージェントやユーザインターフェースエージェントをアプリケーションの対象としており、高度な応答性や実時間性を必要とするマルチモーダル対話システムのアプリケーションを考えた場合、それらの要件を十分に実現することができるか疑問が残る。マルチモーダルシステム開発において対話処理などの高度な機能を実現する部分に、これらマルチエージェント用の接続アーキテクチャを適用した例も見られる [PWGK99] が、同様の技術を使って対話からモータ制御など広い処理の粒度を持つロボットに必要とされる全ての機能を実現できるか考えると、多くの問題が生ずると考えられる。システムに必要とされる全ての機能の実現に足りる接続アーキテクチャ



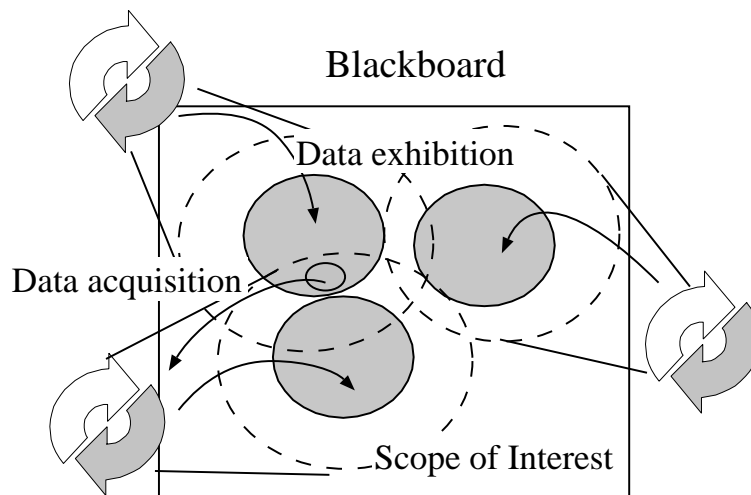


図 2.1: モジュールの動作形態

を考えた場合，それを効果的に実現するためには，ハードウェアのレベルから議論する必要がある．

本稿で考えるモジュールの動作形態を図 2.1 に示す．モジュールとは，システムを構成する分割再合成可能な基本単位であり，本稿においては，その機能を実現するソフトウェアとともに，そのソフトウェアを動作させるプロセッサを合わせて指す．システムの中心には，どのモジュールでも自由にアクセスできる共有エリアを設定する．これを黒板と呼ぶ．モジュールの基本的な動作プロセスは以下のようなになる．

1) 黒板を見る：黒板を見に行くとシステム上で公開されているすべての情報が一覧できる．

2) 必要な情報を選定する：黒板上から必要な情報を選定する．

3) 情報を配信する：モジュールで生成した新たな情報を黒板上に公開する．

システムの中心に黒板を配置することで，どのモジュールであってもどのデータに対しても自由にアクセス可能となる．これにより，モジュールの関係がフラットになり，各々のモジュールが各々の興味でデータを参照しに行くことで，情報の流れの構造のダイナミックな変化を容易に実現することができるようになる．

黒板モデルは 1970 年代に音声理解システムの構築モデルとして提案され，そこで構築されたシステムは，音響的知識，言語的知識，タスクの知識などを統合して音声理解をすることができた [EHRLR80]．黒板モデルは，ワーキングメモリを

持った知識処理系としての側面を持ち、その観点から、様々な議論が展開された。それとともに注目されたことに、モジュール化開発の方法としての側面がある。システム開発に必要な情報を黒板を使って集中開示することで情報の一覧性が向上し、システムを複数のモジュールに分割して開発しても、再びスムーズに統合することができた。文献 [LC81] において、オープンで制約の緩い統合システム開発方式としての黒板モデルについて議論されている。本稿では、このシステム開発方式としての黒板モデルに着目し、それをマルチモーダルシステムのモジュール開発に適用することを目指す。

上記のように、本稿では、接続アーキテクチャの検討は、システムの動作や開発モデルを規定する上でも重要であるという視点に立って議論を進める。この視点に基づき、本稿で検討を行なう接続アーキテクチャは、以下の条件を満たすことを目標とした。

a. マルチモーダル対話システムに要求される動作の実現に適した枠組みを提供する：タスクのダイナミックな変化に対する追従や、システムの自律性や応答性の実現を容易化する枠組を提供する。

b. マルチモーダル対話システムに要求される機能の統合に適した枠組みを提供する：多機能システムに要求される処理の粒度は、モータ制御から対話処理まで多岐にわたる。作成するモジュールは処理の粒度において、モータ制御から対話制御まで、幅広いスケーラビリティを持ち、それらがシームレスに統合される枠組みを提供する。

c. 開発者にとって自由度が高く負担の少ない開発モデルを持つ：モジュールの開発者が、自分の製作するモジュールとその周辺の知識を持つだけで開発に参加ができ、他のモジュール開発者の共同作業の中でシステムを組み上げていくことを容易にする開発環境を提供する。モジュールの変更や付加が、システム全体の性能に影響を及ぼさないようなアーキテクチャを定義し、そのアーキテクチャをできるだけ理想的に実現する接続アーキテクチャを提供する。

上記の目標を満たす接続アーキテクチャとして、以下のようなアーキテクチャを提案する。

黒板モデルと publish/subscribe モデルを複合させたシステムサービスを提供することで、情報の閲覧や購読の切替をモジュール側から行なうことができるよう

にする．これにより，応答性が高くフラットな関係のモジュールが情報の流れを動的に切替えながら動作する動作環境が実現される．情報を分散共有し，かつ割り込みをコントロールすることで応答性の高いネットワークを作ることに關しては，ASPIRE[山崎 96] と問題意識を共有するが，本稿では，ASPIRE において割り込み線を張りめぐらせることで実現された割り込み応答機能をバスコントローラのサービスとして提供し，動作中にでも動的にコントロール可能とすることで，情報の流れの動的な再構成においてより高い柔軟性を持ったシステムを実現する．

対話からモータ制御まで必要とされるシステム開発において，要求される機能の範囲を全てカバーすることを考えると，実時間性や応答性，統合の密度や必要とするリソースなどの要件は，各々の機能によって大きく異なることが考えられる．一枚岩の接続アーキテクチャでそれら全ての要件に答えることを考えると，特性によって適・不適のある接続アーキテクチャの中で，マルチモーダルシステム開発の要件全てを満たす特性を持った接続アーキテクチャを設計することは非常に困難である．本稿では特性の違う二つのバスによりモジュール間を接続することを考える．シェアードバスと point-to-point ネットワークからなる 2 層のバスアーキテクチャを構成し，各々のモジュールを各々の要求する条件に適合する特性を持ったバスに接続することで，スケーラビリティの高い接続アーキテクチャを実現する．それら 2 層のバスをブリッジで接続することで，特性の異なるモジュール間でもシームレスな統合を実現する．

開発者にとって自由度が高く負担の少ない環境を提供するために，開発者が数ある情報の中から，自らの専門性の範囲内の絞り込んでアクセスすることを可能にするシステムサービスの提供し，開発者の活動を補助するためのツールやライブラリの提供を行なう．また，リアルタイム性の仕様を満たしながらもシステムの拡張・再構成を行いやすい，マルチプロセッサのプラグイン型のアーキテクチャを提供する．

以下，提案アーキテクチャの詳細について記述する．

## 2.3 モジュール間接続アーキテクチャの検討

### 2.3.1 情報公開・メッセージ通知モデル

各々のモジュールで生成された情報をシステム内のどのモジュールからも参照できるようにすることで、モジュール同士のフラットな関係を実現する。このような情報共有を実現する枠組みとして集中・開示型の情報公開モデルである黒板モデルを採用した。黒板モデルでは、各々のモジュールが共有の情報の開示場所に情報を「タグ」に書き込むことで、どのモジュールからもそれらの情報を参照することが可能となる。黒板モデルはデータ開示モデルとしては優秀なものであるが、データの変化を検知する用途には非効率である。そこで、メッセージ通知に適した枠組みである publish/subscribe モデルをあわせて採用した。publish/subscribe モデルでは、興味のあるデータを購読 (subscribe) リストに入れておけば、そのデータに対する変化の通知 (publish) を受けることができる。情報公開サーバは、黒板モデルに基づく情報開示サービスを提供するサーバと、publish/subscribe モデルに基づくメッセージ通知サービスを提供するサーバとの組み合わせにより構成される (図 2.2)。

各々のモジュールは、情報開示サービスを利用して、自分にとって必要な情報を選定し、注目すべき情報はメッセージ通知サービスを利用して購読、新たに生成した情報を情報開示サービスを通して公開する。同じデータに対するアクセス手段として、情報取得とメッセージ配信が等価的に選択できるため、データを状態としてアクセスできると同時にイベントとしても取得できる。また、情報開示サービスには、センサーなどの外部情報だけでなく、システムの内部情報を区別なく開示できる。この枠組みによって、マルチモーダルシステム開発の要件となる自律性と応答性を両立できる環境でシステムを開発することが可能となる。

### 2.3.2 プロセッサ間の接続方法

上述した情報公開・メッセージ通知モデルを実装するにあたって、プロセッサ間の接続方法を検討する。プロセッサ間の接続方法には、主としてシェアードバス型と point-to-point (以下 P2P) ネットワーク型の接続アーキテクチャがある。

シェアードバス型の接続方法をとって実現を考えた場合、情報の伝達が常に定

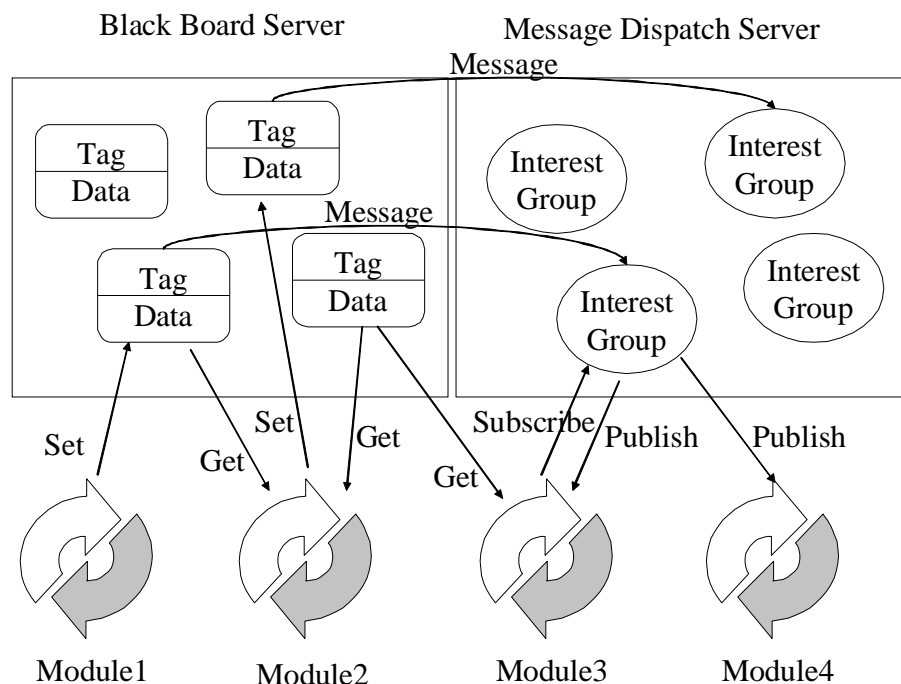


図 2.2: 情報公開・メッセージ通知モデル

時間で行なわれることが保証できる，サブスクライバがいくら増えてもシステムのパフォーマンスに影響がない，などの長所がある．しかしながら，バスという形状上の制約から，プロセッサ間の距離やモジュールの物理的な配置に制約が生まれる．

P2P ネットワーク型の接続方法をとって実現した場合，接続における物理的な制約が大きく緩和され，比較的に自由な空間に自由な大きさを持ったプロセッサを配置することができる．しかしながら，モジュール間を P2P で接続するため，サブスクライバの増加によって通信パフォーマンスが悪化していくこと，到達するまでのステップ数が多いプロセッサ間が通信する場合の通信時間の予測不可能性が問題として生ずる．

物理的な距離の短さを要求する実時間性を確保しながらも，物理的な体積を要求する高度な処理を実現する必要があるロボット開発においては，システムを階層化させ，各々の機能を実現するプロセッサをロボットの内と外に配置することが一般的に行なわれる．本システムにおいては，情報公開・メッセージ通知に基づく接続アーキテクチャを，ロボット上ではシェアードバスとして，ロボット外

では P2P ネットワークとして実装し、両者をブリッジによって接続する 2 層のバスアーキテクチャをとった (図 2.3)。

ロボット上では、シェアードバスの構成を採ることで、データの講読者がいくら増えようと、定時間でメッセージが配信されることを保証できる。また、シェアードバスの構成により実現される分散共有 DB は、全ての最新の情報を保持し、各々のプロセッサが、それら全ての情報に非同期にアクセス可能となる。これは、ロボットの実時間性を保証する上で重要な要素であり、実時間性に厳密な協調処理を行う必要のあるモジュールは、ロボット上にバス接続で実装される。

P2P ネットワークの構成は、接続における物理的な制約を軽減するのみならず、従来ハードウェアを活用できる面でも有効性が高い。本システムにおいては、実時間性に厳密でないモジュールや、密な協調処理を必要とせず、情報の更新頻度が低いモジュールは、ロボット外部に P2P ネットワーク接続で実装される。

### 2.3.3 処理の粒度の異なるモジュールの統合

各々のモジュール間のメッセージ送受信は、非同期に行われる。情報開示サーバからのメッセージは、まずモジュール側に用意されたバッファへ格納される。メッセージの受信に自由度のあるバッファを採用することで、処理の粒度の異なるモジュール間の通信が実現できる。たとえば、上位の粒度のモジュールが下位の粒度のモジュールが出力する情報を参照する場合は、送られてくるメッセージを一時的にバッファに溜め込んでおいて自らの処理の周期にあわせてまとめて処理すればよい。また、下位の粒度のモジュールが上位の粒度のモジュールが出力する情報を参照する場合は、バッファをポーリングしてメッセージが入っていないときには、自らの処理を継続することができる。

## 2.4 アーキテクチャの実装

### 2.4.1 システム構成

上記検討に基づき、ロボットと情報処理 PC からなる実験システムを構築した (図 2.4)。

ロボットは、頭部に 2 台のカメラ・2 台のマイク・2 本のアームを持ち、移動台

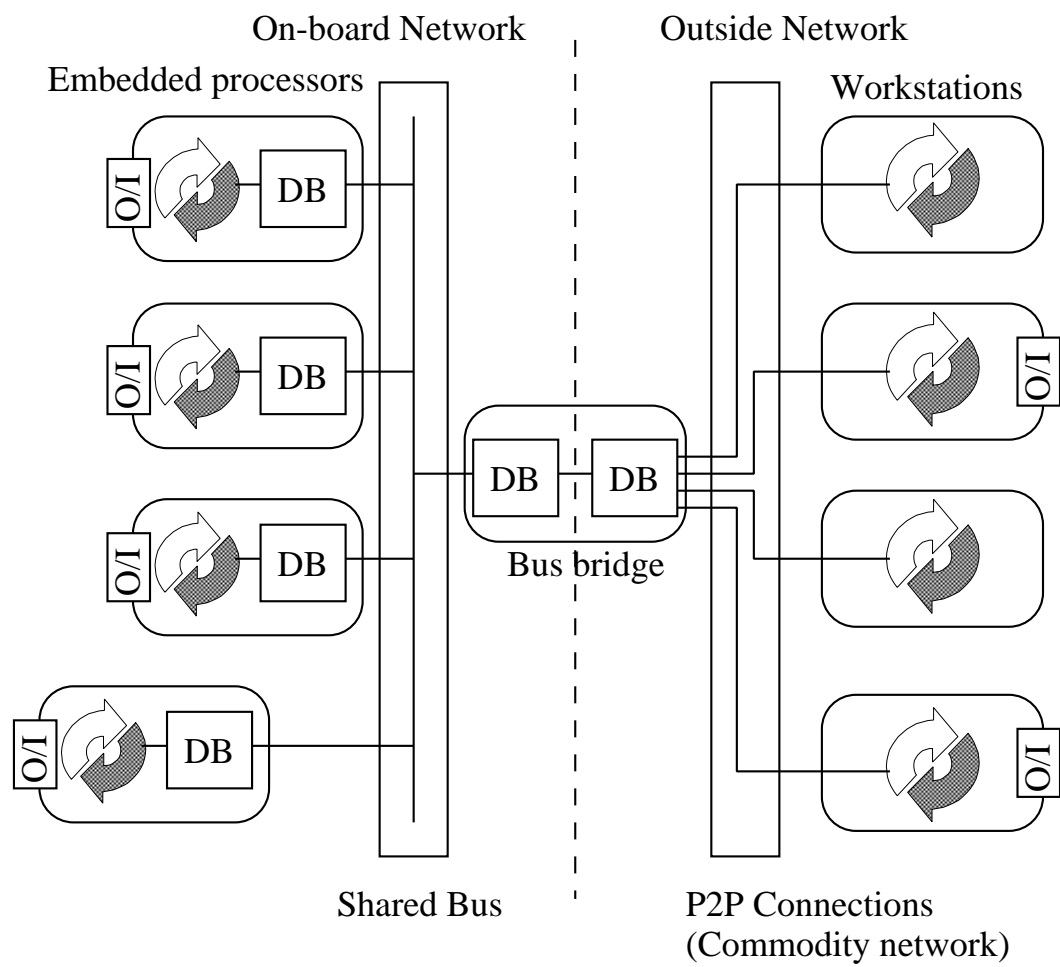


図 2.3: 2 層のバスアーキテクチャ



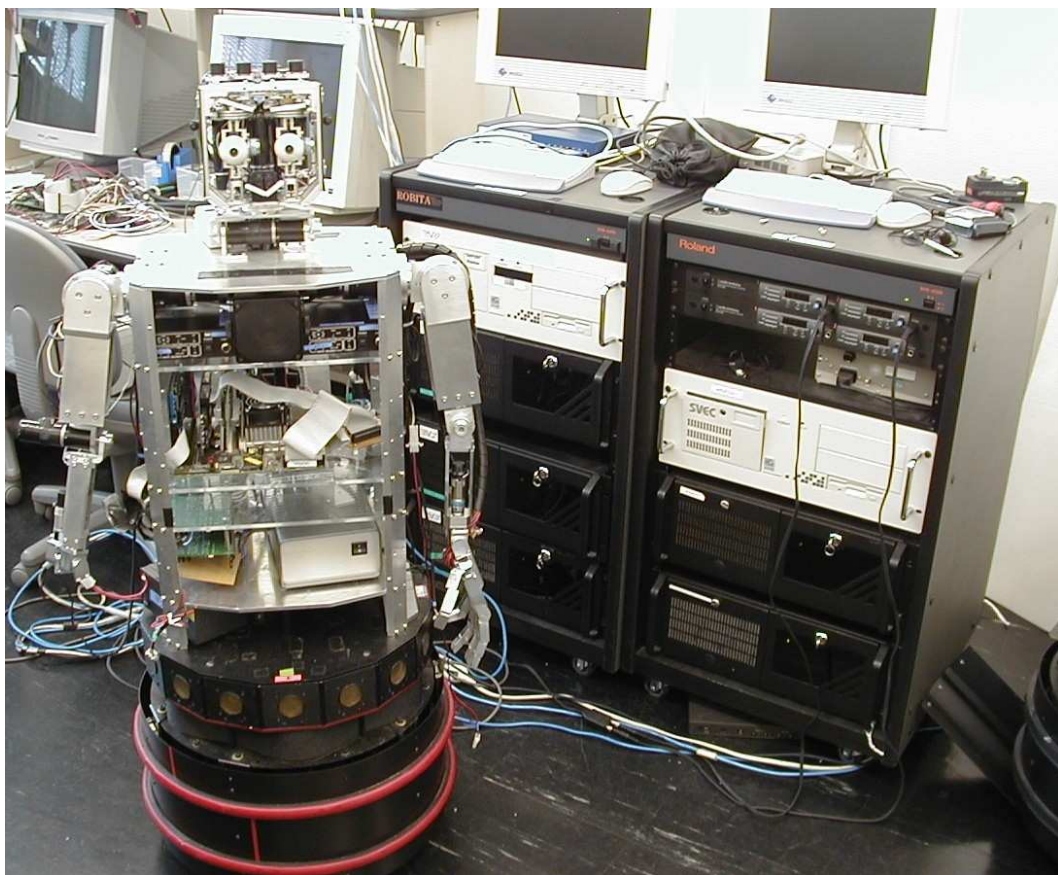


図 2.4: システム全図

車による自立移動が可能である．このロボットに，情報公開サーバを中心に 7 台のクライアント PC を接続した．モジュール作成者は，ロボット上に用意されたプロセッサや，ロボット外部のクライアント PC 上に各々の作成したモジュールを動作させる．

#### 2.4.2 オンボードバス

ロボット上には情報公開・メッセージ通知サービスが，シェアードバスによって実装される．

バスコントローラのデータ送信部は，アービトレーション機能を持ち，マルチマスタ構成による通信が可能である．バス上に配信される信号は，データの種別を識別する 8bit の識別子と，16bit のデータからなり，8MHz のバスクロックで同



期転送される。

バスコントローラのデータ受信部は、1Kbyte のデュアルポートメモリと、ビットマスクに基づくデータ受信検出器からなる。バス上をデータが流れると、各々のモジュールは、逐一データを読み込んで、デュアルポートメモリ上の相当するアドレスに書き込む。これにより、非同期にどのデータに対してもアクセス可能な黑板サービスが、デュアルポートメモリにより実現される。また、データの受信時に、デュアルポートメモリ上の特定アドレスに対する書き込みを検出し、プロセッサに対して割り込みを発生させる機能を付加することで、メッセージ配信 (購読) サービスが実現される (図 2.5-a)。

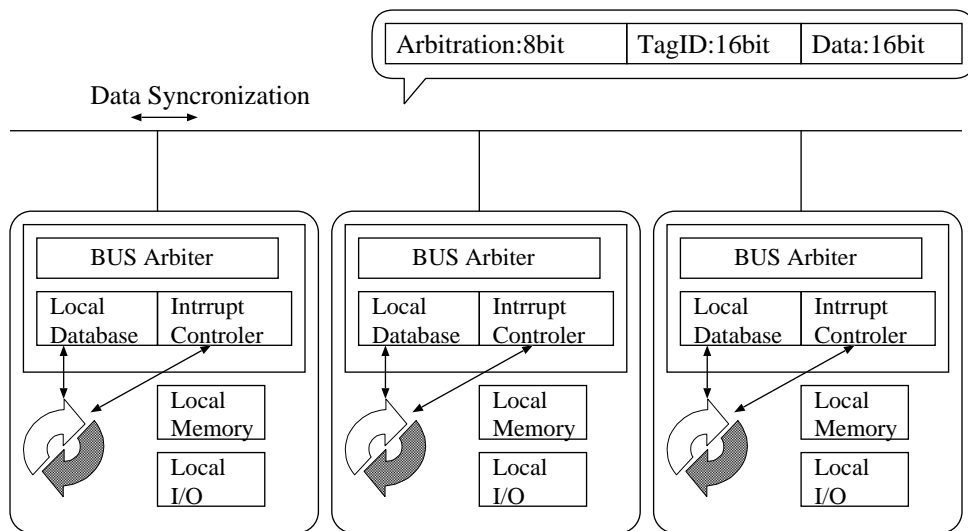
オンボードプロセッサは、SH2 プロセッサと FPGA により実装されたバスコントローラ、モータドライバ、アンプ・AD 変換器、ビデオデコーダなどの周辺回路からなり、ロボット上に設置されたスロットに差し込めるよう、一様な形状をしている (図 2.5-b)。開発言語としては、C 言語が利用可能であり、開発者にはバスコントローラへのアクセスを簡便にするライブラリを提供した。

### 2.4.3 ネットワークサーバ・クライアント

外部ネットワークに対するシステムサービスを提供するサーバプログラムは、2 つのプロセスからなる。ひとつは情報開示サービスを提供するものであり、ひとつはメッセージ配信サービスを提供するものである。

表 2.1 と図 2.6-b に、クライアントからの操作コマンドとそのコマンドに対するサーバの応答を示す。黑板サービスを提供するサーバは、クライアントからの黑板への書き込み (“SET”)・読み出し (“GET”) の要求に対して応答を返す。黑板上のどのデータに書き込む [読み出す] かは、ユーザ指定可能な文字列 (“[tag]”) により区別される。各々の書き込み・読み出し要求には、書き込み先 [読み込み元] を指定するタグと、それに引き続いて書き込むべきデータ (“[data]”) が送信される。また、書き込み・読み出し要求以外に、現在黑板に書き込まれているタグの名前の種類を取得する (モニタする) ための API (“MON”) を用意した。モニタ要求が入力されると、指定された数字 ID から、その番号をインデックスとするタグ文字列を返す。

メッセージ配信サービスを提供するサーバは、クライアントからの購読要求



(a) Shared-bus style topology.



(b) Embedded processor boards and bus bridge.

図 2.5: シェアードバス型のトポロジとオンボードプロセッサ。オンボードプロセッサは、モータドライバ(左下)、画像処理ボード(左上)、音響処理ボード(左中)の種類があり、ロボット上に複数搭載可能。右上はバスブリッジ、右下は12cmのCD。

(“SUB”)・メッセージ配信要求 (“POST”) を受け付け、各々のクライアントにメッセージを配信する。購読要求は正規表現による指定を使ったグループ単位での購読が可能である。メッセージの配信は、UDP を利用したパケットのブロードキャストが有効であるが、パケットの到着の確実性を保証できないことから、本システムでは TCP による配信を複数回行うという形で実装を行った。

各々のモジュールは、上記したサーバに接続されたクライアントであり、ソケット通信によってサーバにアクセスする。黒板サーバへのアクセスは非同期に動作する。メッセージ配信サーバからのメッセージは、まずクライアント側に用意されたバッファへ格納される。クライアントプログラムは、メッセージがバッファに届いているかいないかをブロッキングなしで確認する (ポーリングする) か、メッセージが届くまで指定時間ブロッキングするかを引数として指定できる。また、UNIX のプロセス管理の枠組みを利用して、下位の作業を担当する子プロセスを生成した上で、親プロセスがブロッキング読み出しでメッセージ受信を検出し、子プロセスにシグナルを送ることで、受信割り込みを実現することができる。

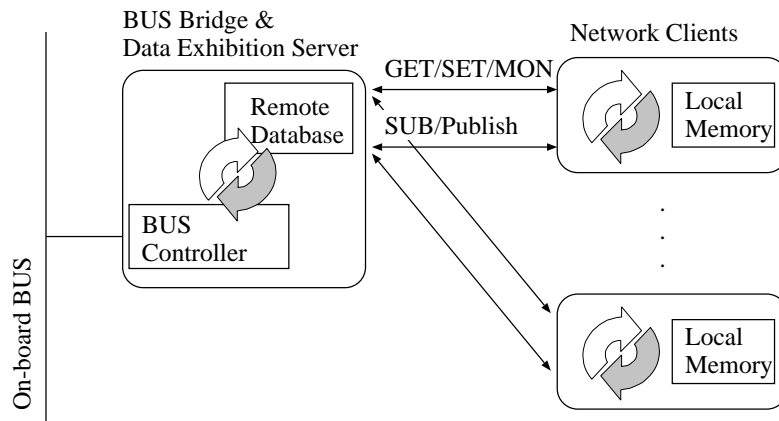
高級開発言語とのインターフェースとして、C++、Perl、JAVA 用のクライアントライブラリを用意した。各々のライブラリはクラスライブラリとして定義され、言語による API の差はない (図 2.7-a)。

表 2.1: API 一覧

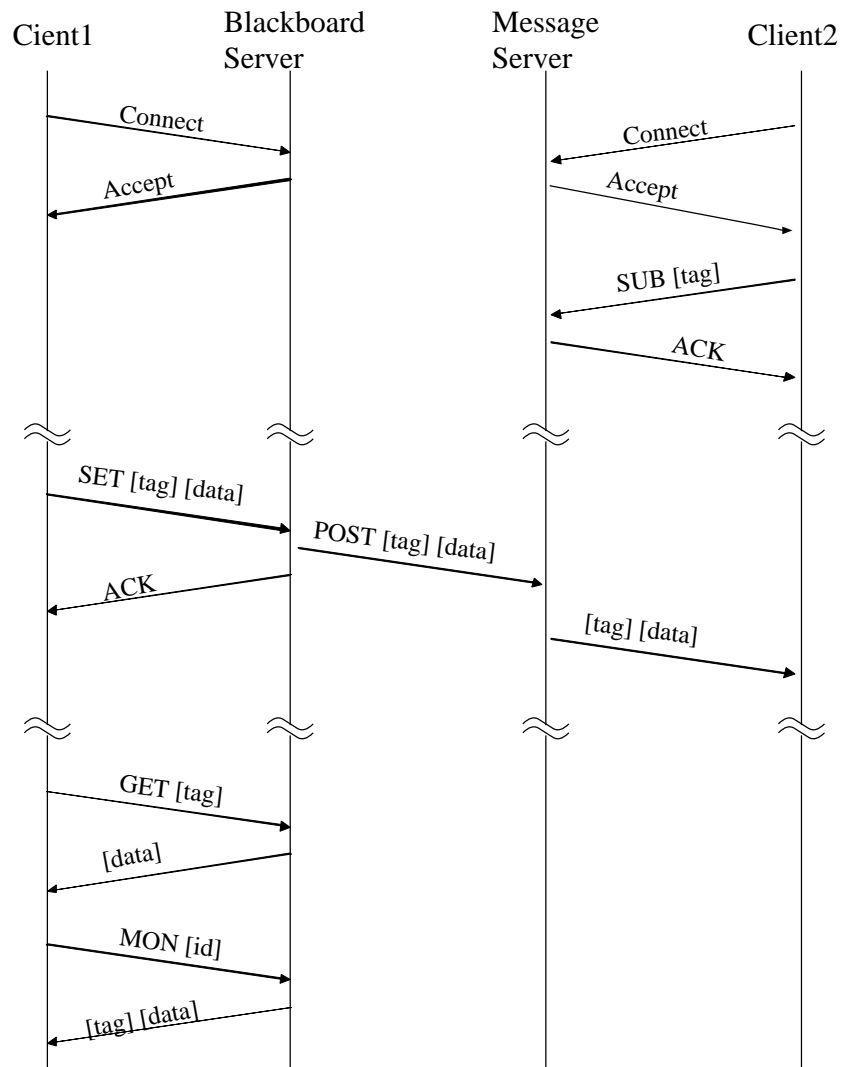
API 名	機能	応答	ポート
SET	黒板から情報を取得	ACK	8027
GET	黒板に情報を書き込み	[data]	8027
MON	黒板上にあるタグ名を取得	[tag+data]	8027
POST	メッセージを配信	なし	8028
SUB	タグを購読	ACK	8028

#### 2.4.4 バスブリッジ

オンボードバスとネットワークは、ブリッジによって接続される。ブリッジにおいては、オンボードバス上を流れるデータのネットワークプロトコルへの変換、オンボードバスで数字で識別されているデータとネットワーク上で使われるテキ



(a) P2P style topology.



(b) Network protocol.

図 2.6: P2P 型のトポロジとネットワークプロトコル

スタグを相互にマッピングする処理が行われる。

バスブリッジとネットワークサーバは、できるだけ近距離に配置される。バスブリッジとネットワークサーバの間は、通常は、有線のネットワークでつながれるが、ロボットの移動を伴う実験を行なう場合は、無線ネットワークで接続される。

#### 2.4.5 タグのグループ購読

外部ネットワークに接続されたモジュールにおいて、データの「タグ」には、ユーザが自由に定義できる文字列が使われる。通常、データへのアクセスは、これらの文字列の直接指定により行なうが、イベントの購読サービスに関して、正規表現によるタグのグループ指定での購読サービスを提供した。この機能を利用することで、開発者は全体のデータを自らの興味のある領域のデータを絞った上で概観することができる。

ロボット上でバス接続されたモジュールにおいては、同様の機能が数字 ID のビットマスクにより実現される。

#### 2.4.6 開発ツール

開発者用のツールとして、コマンドライン動作するツールをいくつか提供した。黒板に書き込まれているタグとデータのペアをリアルタイムにモニタするツール、黒板の内容を標準出力にダンプ [アンダンプ] するコマンド、購読したメッセージへの通知を標準出力に出力するコマンドなどがあり、標準出力のパイプやリダイレクトなどを使って、UNIX の標準コマンドと組み合わせることで、動作のログをとって分析したり、状況の再現をしたりすることができる (図 2.7-b)。

#### 2.4.7 黒板のデータ構造と操作コマンド

外部ネットワークにおいては、黒板のデータに対して以下のような高度な操作を行うことができる。

```

Example of Tag
"facereco.dir" [60]
"taiwa.stream.who2who" [A:Hello!>B]

```

```

API (case of C++)
dir = bbc.GetAsDouble("facereco.dir");
bbc.SetAsString("taiwa.stream.who2who",
               "B:Good bye!>A");
mmc.Subscribe("taiwa\stream\.*");
mmc.RecvMessage(10.0);

```

(a) Example of tags and APIs.

Command line tools

```

bbmon    :Real-time monitoring of black board.
mmcat    :Output subscribed events to stdout.
bbdump / bbundump
          :Dump/Un-dump a content of blackboard.

```

Usage

```

bbdump > hoge.txt    :Save current state.
bbundump < hoge.txt  :Restore a state.
mmcat "taiwa\stream.*" | timestamp > event.log
                  :Save an event.
timesync event.log | bbundump
                  :Restore an event.

```

(b) Command line tools and its usage.

```

|-- [tojo    ] control
|   |-- [tojo    ] arm
|   |-- [tojo    ] eye
|   `-- [tojo    ] nomad
-- [omoto    ] deco8
-- [ishii    ] e_taiwa
|   |-- [ykansaku] db_response
|   |-- [miyanaga] dialogue_act
|   |-- [ishii    ] emotion
|   |-- [fujie    ] parser
-- [kubota   ] facereco
|   |-- [yosuke   ] detect
|   |-- [kubota   ] dir
|   `-- [yosuke   ] track
`-- [yosuke   ] videod

```

(c) Modules in CVS tree (part).

図 2.7: タグ・API とコマンドラインツールの例

## データ型

黒板に書き込まれるデータは、浮動小数点型、文字列型などの基本的なデータ型によって定義される。文字列型のデータの場合、開発者間の合意の元であればどのようなフォーマットのデータも書き込むことが許される。

開発事例 (2.5章参照) においては、データのやり取りをするフォーマットとして、単純な文字列型、特殊記号で仕切られた文字列型、CSV 形式によるリスト型の他、XML により構造化されたデータ型などが用いられた。用いられたデータの実例を図 2.7-a に示す。

## タグの命名規則

本システムにおいては、ソフトウェアのソースコードのバージョン管理、並行開発などの為に、UNIX の標準的なバージョン管理システムである CVS の機能を利用している。各々のモジュールにおいて定義するタグの名前には、「CVS に登録されたモジュール名 + ユーザ定義文字列」とする命名規則を課した。この命名規則により、CVS に登録されたモジュール間では、名前の重複が起こることはない。

これらタグの命名規則と正規表現に基づいたグループ購読機能を用いることによって、例えば、“`facereco\track\.*`” として情報を購読すればモジュール “`facereco.track`” の出力する全ての情報を概観することができる。また、例えば、画像認識による個人識別結果と音声認識による個人識別結果を統合したい場合は、“`(facereco|speech)\.*\id`” として情報を購読すれば、音声・画像の両情報源からの個人 ID の最新情報を取得することができる。

開発事例 (2.5章参照) において用いられた CVS ツリーの一部を図 2.7-c に、定義されたタグの一部を図 2.7-a に例示する。

## 2.5 システムの稼動状況

これまでのシステムの稼動期間において、ロボットに、音声・音響・画像・言語処理を行いながら多人数の話者との会話に参加するシステム [松坂 01] や、状況に応じた表情生成を行いながらユーザとの対話を行うシステム [TMIK00]、ロボットアームを通した指示表現と音声による言語表現を用いてユーザと空間的情報を共

有する対話システム [古川 99] , インターネット経由でのロボットの遠隔操作システム [中野 99] などの多数のシステム [vid] を実装する機会を得た .

多人数の話者との会話に参加するシステムを例としてあげると , 開発の結果 , 約 15 種 (図 2.7-c) のモジュールと 100 種を超えるタグが定義された . 最終的なモジュール間の参照関係を書き下したものを図 2.8 に示す .



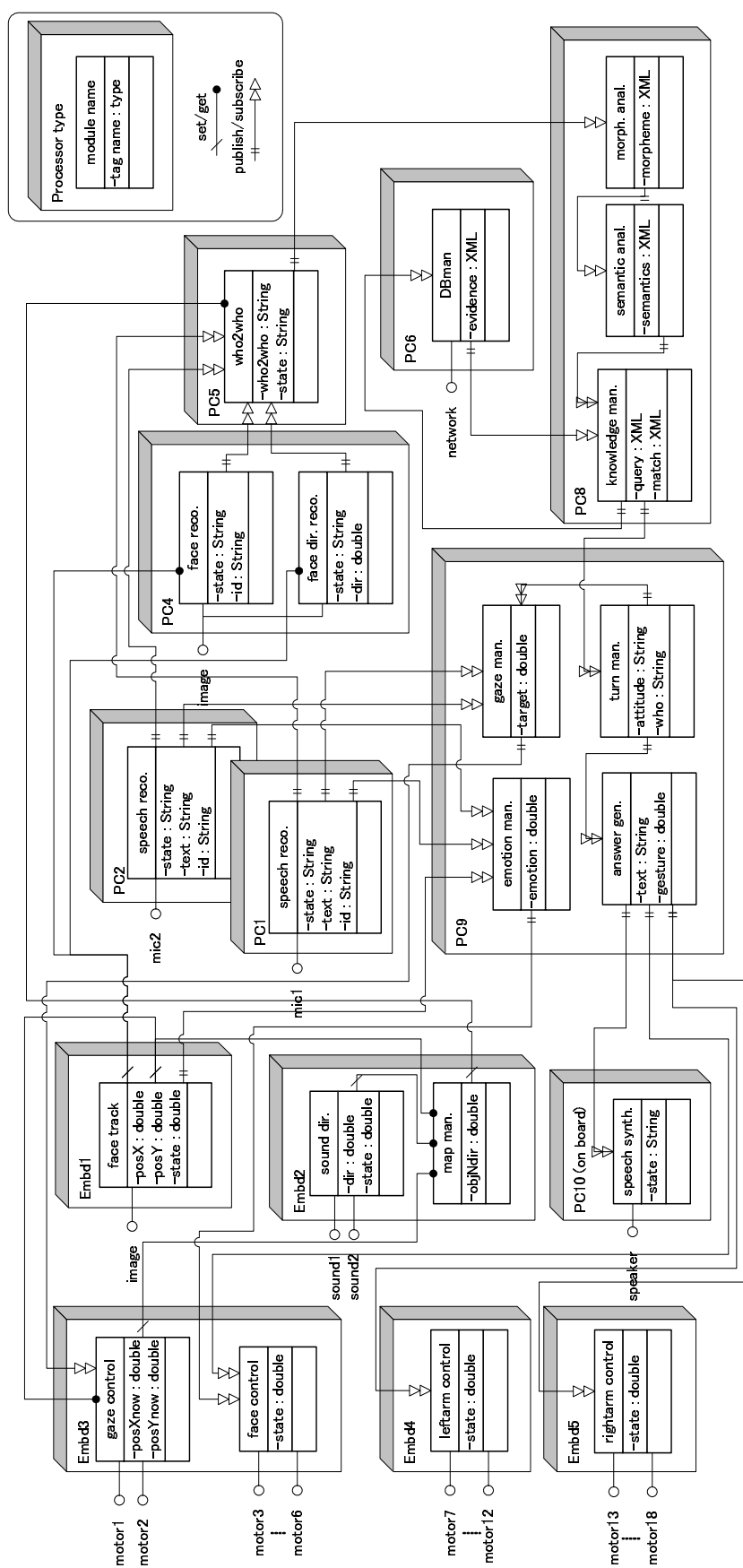


図 2.8: 生成されたモジュール構成とデータの参照関係

このタスクにおいては、10msec 周期で動作する音声・音響・モータ制御モジュールやフレームレートで動作する画像処理モジュールなどの下層の処理を統合的に利用しながらも、ユーザとの会話のための言語処理などの低頻度で高度な処理を行なう必要がある。最終的なモジュール間の参照関係において、ユーザの追跡のための画像処理とモータ制御の間的高速なフィードバックループがロボット上に配置されたプロセッサ群で形成される一方で、音声・画像認識など中粒度のモジュール、処理結果を統合する非同期動作のモジュール、対話を担当する大粒度のモジュールなどがロボット外部に形成された。自ら情報を取得しに行く黒板型の情報アクセスは、主に制御モジュールからの画像処理結果の参照などの一定周期で動作するプロセス間での情報共有で用いられた。購読した情報が自動で通知される publish/subscribe 型の情報アクセスは、主に、対話などの更新が低頻度な情報の参照や、制御モジュールの上位プロセスからの制御コマンドの取得などの応答性を要求される部分に用いられた。

多くのモジュールは、自らが購読した情報の更新を待つ待機状態にあっても、プロセスの自己ループを継続し、定時間毎に現在購読している以外のタグを参照し、時に新たな行動を起こすための情報を出力するようにプログラムされた。結果としてシステムは、高い応答性を持ちながらも、時に、黙ってしまったユーザに対して自主的な語りかけを開始したり、ユーザが去った後にマップ更新のために周囲を見回すなど、自律的な動作を起こすことができた。

本システムにおいて、開発者となったのは、音声画像処理・パターン認識・人工知能・機械制御などを専門とする学生たちであり、どのシステムも平均して同時に8人ほどの開発者がシステム開発・統合作業に関わり、現在3年を超えようとしている稼動期間を通してのべ15人ほどの開発者がシステム開発にかかわってきた。開発者たちは、明示的なシステム設計者が存在しない状況で、タグの策定やデータのフォーマットの定義などの開発者同士の情報交換によりシステムを組み上げていった。各々のモジュールの統合作業の開始から、システムが満足のいく動作をするまでにかかる時間は、長い場合で1ヶ月ほどであり、特に既存のシステムの拡張や手直しによって対応できる課題である場合は、黒板サービスを経由して既存のシステムの出力する情報を参照・利用したり、不足した処理能力を補う新たなプロセッサをバスへ増設することによって、1~2週間の短期間でシステム

を構築することができた。

## 2.6 議論

本章においては、多機能ロボット開発に適した接続アーキテクチャとして、モジュール間のフラットな関係、モジュール自身による情報のダイナミックな切替え、密な統合とスケーラビリティの確保を両立できることを特徴とする2層のバスアーキテクチャを提案し実装した。

まず、拡張性と応答性に長けたシステムの実現に関して、シングルプロセッサにリアルタイム OS を搭載すれば十分という考え方がある。シングルプロセッサシステムとマルチプロセッサシステムを比較した場合、システムの拡張性や、リアルタイム性の確保のしやすさにおいて、後者に大きな分がある。マルチプロセッサの構成でシステムに機能を拡張することを考えた場合、機能に必要な I/O をプロセッサと組で付加することで、容易に I/O の増設を行なうことができる。また、機能とそれに必要な処理能力を組で付加することで、既存のシステムの処理性能に与える影響を最小にすることができる。しかしながら、マルチプロセッサの構成は、プロセッサと周辺回路、プロセッサ間の接続アーキテクチャを必要とするため、シングルプロセッサの構成に比較して設置スペースを必要とするデメリットがあり、システムの設置スペースに制約を持つロボットを開発する場合、個々のプロセッサの小型化や接続アーキテクチャの柔軟性が重要な要件となる。本稿において提案・実装したバスアーキテクチャは、ロボットへの搭載に当たって妥当な大きさと柔軟性を実現することを目指した。ロボット内部に搭載したバスにおいては、組み込みプロセッサによりプロセッサボードを実装することで、各々のモジュールを小型化した。また、それを外部ネットワークに接続することで柔軟性を確保した。

また、モジュール間の接続トポロジに関して、山崎らによるレスポンスブリンク [山崎 01] や IEEE-1394[iee] などの、ネットワーク状のトポロジによる接続アーキテクチャが提案されている。これらネットワーク状のトポロジは、情報の共有の際に情報のルーティングを必要とする。そのため、ロボットのタスクの変更など、情報の流れを変える必要のある際に、各々のインターフェースの持つルーティン

グテーブルを書き換えるためのオーバヘッドが発生する．また，各々のモジュールは，ルーティングされてこない情報に対しては，無知である他にない．これは，各々のモジュールが上位から指定された情報のみにしかアクセスできず，タスクスイッチに関する権利を持つ中央集権的なモジュールに対して，従属的な関係を持つことを強制されることを意味する (図 2.9:a-2)．一方，自律システムにおいて，高いモジュラリティと柔軟性を実現するためには，上位から動作形態が決定される動作環境 (図 2.9:a-1) ではなく，それぞれのモジュールによる行動決定がシステム全体の行動に反映される分散協調的な動作環境 (図 2.9:b-1 b-2) が必要である (詳細については，3 章で議論する)．これらの観点から，本研究では，バス状のトポロジによる情報共有法を選択した．

本研究において設計・実装したオンボードバスを ASPIRE の実装 [山崎 96] と比較した場合，ASPIRE が割り込み機能の実現のために，割り込み線を P2P で張りめぐらせるのに比べ，信号をディジーチェーン接続し，各々の割り込みの有無を動作中にも動的にコントロールできるため，割り込み制御の柔軟性が高い．その反面，割り込みが発生するためには，その前にバスのアービトレーションに勝つ必要があり，これが応答性の低下の原因となる．十分に機能分散されたシステムにおいては，情報の共有は抽象化された状態で行なわれるため，未処理の音声・画像情報などのような大量の情報がバス上を流れることは一般に稀である．その場合，それぞれのトランザクションでやりとりされるデータは小さく，バスの解放率は高くなり，十分な転送能力を持ったバスであれば，割り込みの応答性に与える影響は小さいものとなる．したがって，バス状のトポロジにおいて高い応答性を実現するためには，十分な機能分散を実現し，扱う情報を抽象化させることが重要な要件となる．本研究では，組み込みボードと外部 PC の両者において，音声・画像・制御の入出力インターフェースを持つプロセッサを用意し，取り込んだ情報をシステムバスに流れる前段階で処理することで，これら抽象化の要件に応えた．

## 2.7 まとめ

多機能システム開発に適した接続アーキテクチャとして，モジュール間のフラットな関係，モジュール自身による情報の流れの構造の動的な切替え，密な統合と

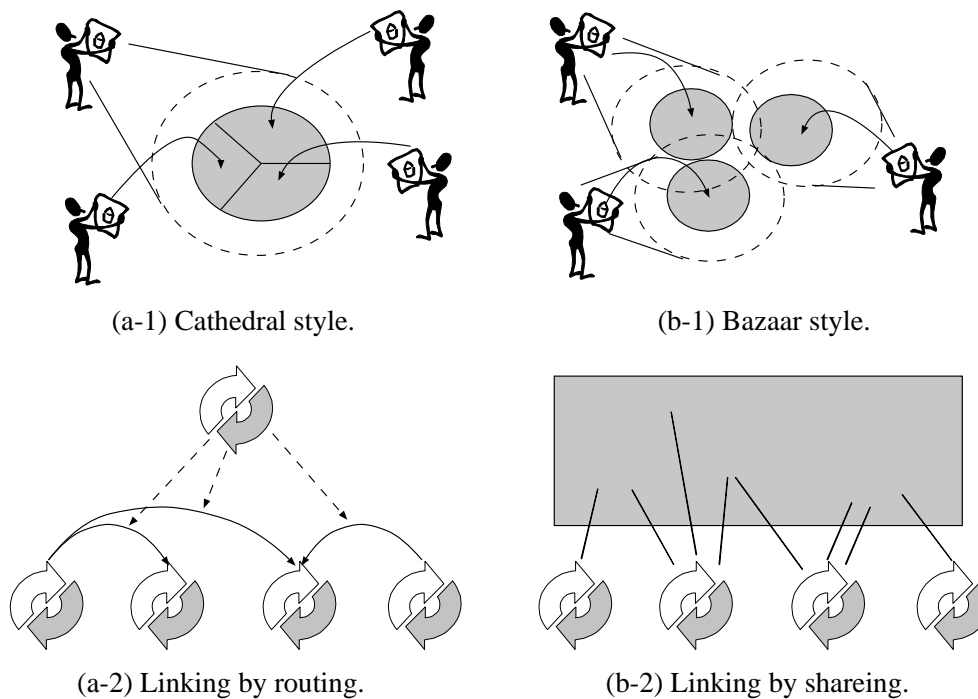


図 2.9: システム動作モデルとモジュール間接続方法

スケーラビリティの確保を両立できることを特徴とする2層のバスアーキテクチャを提案し実装した。実装したシステムにおいては、人にとってその情報の持つ性質をイメージしやすいテキストタグによってデータを区別し、それらを情報の公開と共有に適した黒板上に書くことによって、モジュールの統合活動を容易にした。また、そのテキストタグを正規表現によってグループで購読できるようにすることでメッセージ通知を実現し、モジュール同士の高密度な情報統合や、自立性・応答性を容易に実現可能な枠組みを提案した。多人数の開発者によるマルチモーダル対話ロボットの実装を通して検討したシステム開発環境が良好に動作することを確認した。

## 第3章 基本ソフトウェアと開発環境

### 3.1 はじめに

2章においては、多機能システムのモジュール化開発を実現するハードウェアアーキテクチャについて述べた。本章においては、多機能でインタラクティブなシステムの共同開発を円滑化する、基本ソフトウェアと開発環境について述べる。

多機能を持つシステムを構築するためには、さまざまな分野の知識が必要とされ、到底一人の力では構築しきることができない。システムの開発は、それぞれのモジュールを担当する複数の開発者同士の共同作業となり、共同作業を円滑化する枠組みが必要となる。

これまでインタラクティブシステムの開発においては、開発がトップダウン的に行われることが主流であった。すなわち、システムデザイナーによる、動作環境・タスクの分析に基づいたシステム設計プロセスの完了の後に必要な機能の実装が行われていた。しかしながら、日常の雑多な作業をこなすシステムへ求められる機能が多様であり、時とともに変化する可能性が高いことを考慮すると、システムデザイナーがあらかじめシステムの完成像を見通すことは難しく、従来のなトップダウンの開発形態でユーザの要求に応えていくことは難しい。それらシステムの開発形態としては、タスク・機能を各々のモジュール開発者が各々の問題解決手段に基づいて分析・実装し、それら実装された機能を必要に応じて追加していきることが望ましい。これらの開発形態の差異をオープンソースモデルにおける議論 [Ray99] に習って前者を「伽藍方式」、後者を「バザール方式」の開発モデルと呼ぶことにする。

バザール方式のシステム開発を考えた場合、各々の開発者によって開発された機能が全体の中で統合されるためには、開発者が全体のシステムの中から自ら利用可能な情報を選び出し、選び出した情報に自由にアクセスできる枠組みが必要である。また、機能の付加を考えた場合、すでに動作しているシステムに機能を追

加していくわけであるから、付加した機能が、既存の機能の動作を妨げずに、かつ既存のシステムの中で密に統合されることを可能にするシステムアーキテクチャを検討する必要がある。また、システムが実世界で生起する様々なイベントに対処しながら行動するためには、その状況に適した行動が即時に選ばれる必要がある。また、複数の開発者によって開発された複数のモジュールが互いに協調し、迅速に行動が決定される枠組みを用意する必要がある。

本章では、マルチモーダル対話ロボットを対象とした大規模統合システムの開発の容易化を目的として、バザールのシステム開発と迅速なモジュール間協調を可能とするシステムアーキテクチャを検討する。それを実現する基本ソフトウェア、開発者同士のインテグレーション活動を支援する開発環境を実装し、その開発環境を、多人数の開発者によるマルチモーダル対話ロボットの開発において適用する。開発を通して提案したシステムの妥当性を検討する。

## 3.2 開発における問題点と開発環境への要件

上述の開発手法の問題点を整理し、本研究の立場を明確化する。本稿において提案する開発環境は、以下の点を特徴とする。

黒板モデルに基づく情報公開・共有機能を持ち、ロボットの認知機能、行動生成機能の開発において、機能分割に基づく開発を行うことができる。機能が互いに黒板上で情報交換しつつ認知プロセスが進行し、認知プロセスによる仮説、行動候補、ロボットの各自由度の状態など、黒板上に書き込まれたデータすべてを、システムの状態として等価に扱うことができる。3.3節において、この枠組について詳述する。

また、状況に適した即時的な行動の選択を可能とするために、黒板の監視に基づく状況判断の階層と、それら状況判断に基づき、行動として発動するモジュールの選定を優先度に基づいて行う階層の、二層の階層を持ったアーキテクチャを検討する。開発者によるスクリプト記述と黒板の監視に基づく状況判断機能と、各々の状況においてモジュール間の優先度を管理する優先度管理機能を実装することで検討したアーキテクチャを実装する。3.4節において、この枠組について詳述する。

## 3.3 機能モジュールの開発環境

### 3.3.1 バザールのシステム開発の実現

バザールの統合システムを開発することを考えた場合、開発者の好き勝手に開発が進められると、その開発が破綻するのは自明である。開発をスムーズに進めるためには、開発者を統制する枠組みが必要となる。幾つかの観点から、守るべき最低限の原則を以下のように定めた。

第一に既存のシステムの動作を保証する観点から、

- 付加したモジュールは既存のモジュールに影響を与えてはならない。(不可侵の原則)

第二に後から付加されるモジュールへの配慮から、

- モジュールの生成する情報は、後から付加されるモジュールを含むどのモジュールに対しても公開されていなければならない。(公開の原則)

以下、これらの原則を満たす開発環境について議論する。

### 3.3.2 基本構成・基本的デザインプロセス

バザールの開発環境を実現するためには、各々のモジュールで生成された情報は、システム内のどのモジュールからも参照できなければならない。このような情報共有を実現する枠組みとして集中・開示型の情報公開モデルである黒板モデル [EM88] を採用した。黒板モデルでは、各々のモジュールが共通の情報の開示場所書き込むことで、どのモジュールからもそれらの情報を参照することが可能となる。基本的なデザインプロセスは以下ようになる。

#### 1) 黒板を見る

黒板を見に行くとシステム上で公開されているすべての情報が一覧できる。

#### 2) 必要な情報を選定する

黒板上の自分の興味の範囲の中からモジュールの構成に必要な情報を選定する。

#### 3) 情報を生成するモジュールを作成する

自分の専門の範囲で、選定した情報や、新たに取り込んだセンサ入力から新たな情報を生成するモジュールを作成する。



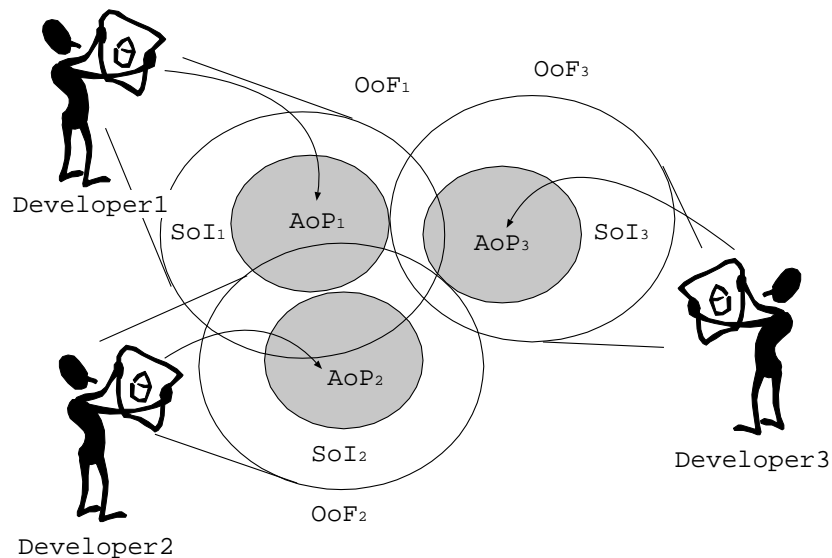


図 3.1: 基本的デザインプロセス．開発者の興味の範囲 (SoI) と自分の専門の範囲 (AoP) と専門外の領域 (OoF) の関係と，黒板を介した間接的な情報共有法．

#### 4) モジュールを動作させる実行環境を用意する

モジュールを動作させるのに足りる実行環境 (プロセッサ) を用意する．

#### 5) システムにプラグイン・情報を配信する

モジュールをプロセッサとともにシステムにプラグインする．モジュールで生成した新たな情報を公開・配信 (以下，コミット) する．

#### 6) 全体の動作を見て調整する

モジュールのシステム全体の中での振る舞いを検証し，問題のある部分があった場合は修正する．

ロボットの動作を付加的に開発していくことを考えると，既存の動作モジュールに対して何らかの影響を与えるモジュールを作るとは不可避であり，3.3.1で述べた原則のうち，不可侵の原則を遵守することは難しくなる．原則を遵守するためには，何らかの工夫が必要となる．本稿においては，ロボットの動作を生成するモジュールは，すべて，動作のマッピングを行うシステムサービスを通して，間接的に実際の動作にマッピングされるようにすることで，これらの問題を回避する (3.4節参照) ．

また，すでに動作しているモジュールに対して，新たな負荷を加えることで，動作を乱すことは許されない．モジュールの付加は，ソフトウェアの付加だけでな

く、それに見合うプロセッシングパワーの付加とともに行なわれる (2 章参照)。

## 3.4 インタラクティブシステムの実現

ロボットを始めとするインタラクティブシステムが持つべき重要な性質として、自律性と応答性の 2 つが挙げられる。自律性は、システムの自発的な動作を表し、ユーザの指示を受けない状況下での活動や、ユーザに対する自発的な働きかけを実現するのに重要である。応答性は、システムを取り囲む状況の変化や、ユーザからの指示などを確実に受理し、それに対する応答を生成できる能力のことを指す。システムの開発について考えた場合、この 2 つの要素を容易に実現できる枠組みを提供する必要がある。

### 3.4.1 自律性・応答性の実現

自律性を実装するためには、ステート遷移によりシステムの行動が制御される枠組み (以下、ステート駆動) が有効であり、応答性を実装するためには、イベント発生によりシステムの行動が制御される枠組み (以下、イベント駆動) が有効であることが一般に知られる。

本稿で採用した黑板モデルはデータ開示モデルとしては優秀なものであるが、データの変化を検知する用途には非効率である。そこで、メッセージ通知に適した枠組みである publish/subscribe モデルをあわせて採用した。publish/subscribe モデルでは、興味のあるデータを購読 (subscribe) リストに入れておけば、そのデータに対する変化の通知 (publish) を受けることができる。この枠組みの実現については、2 章において詳述している。

モジュール開発者は、情報開示サービスを利用して、自分にとって必要な情報を選定し、注目すべき情報はメッセージ通知サービスを利用して購読、新たに生成した情報を情報開示サービスを通してコミットする。開発者は、同じデータに対するアクセス手段として、能動的な情報取得と受動的なメッセージ配信の二つの手段を用いることができるため、インタラクティブシステム開発の要件となる自律性と応答性を実現しやすい環境で開発することができる。

### 3.4.2 即応的なモジュール選択と動作のマッピング

即応的な行動の選択と正しい状況判断を両立させる手段として、本稿では、二層に階層化されたアーキテクチャをとる。上位層は、黒板に書き込まれた情報全体から現在の状況を判断する層であり、下位層は、その判断された状況を元に、即時的な行動選択を優先度に基づいて行う層である（図 3.2）。

各々の状況においては、開発者によるスクリプト記述（以下、状況依存スクリプト）により、モジュールがグループ化されて動作する。グループ化されたモジュールの間には、優先度が定義され、優先度の高いものから順に実行権の付与についての問い合わせが行なわれる。この枠組により、サブサンプションアーキテクチャ [Bro86] のような、上位からの抑制に基づく行動生成が実現される。

各々の状況において、最適に動作するモジュール群の組合せをスクリプトに記述することで、多様な状況に対処可能な階層化システムが実現される。モジュール自体は各々が各々のコンテキストで動作するが、各々の状況において決定された優先度により、ロボットの行動に実際に選択されるモジュールの種類が変化する。行動の変化により、認知プロセスによって黒板上に書き込まれる情報が変化する。これら変化の連鎖により、大局的な行動の遷移が実現される。

ロボットの動作候補を生成するそれぞれのモジュールは、黒板上で重複しないそれぞれの領域に動作候補を書き込む。ここで、黒板の別領域に書き込まれた動作候補を実際の動作へと繋げる作業が必要となる。この作業のことを本稿ではマッピングと呼ぶ。黒板上に書き込まれた複数の動作候補は、動作のマッピングサービスを通してロボットの実際の動作に選択的にマッピングされる。動作のマッピングを行うシステムサービスは、黒板の監視による状況判断機能と、優先度管理から、どのモジュールから生成された動作候補を実際の動作にマップするかを決定する。これらシステムサービスの詳細については、3.5節で述べる。

## 3.5 システムアーキテクチャの実装

### 3.5.1 情報公開・メッセージ通知サーバ

情報公開サーバは、黒板モデルに基づく情報開示サービスを提供するサーバと、publish/subscribe モデルに基づくメッセージ通知サービスを提供するサーバとの

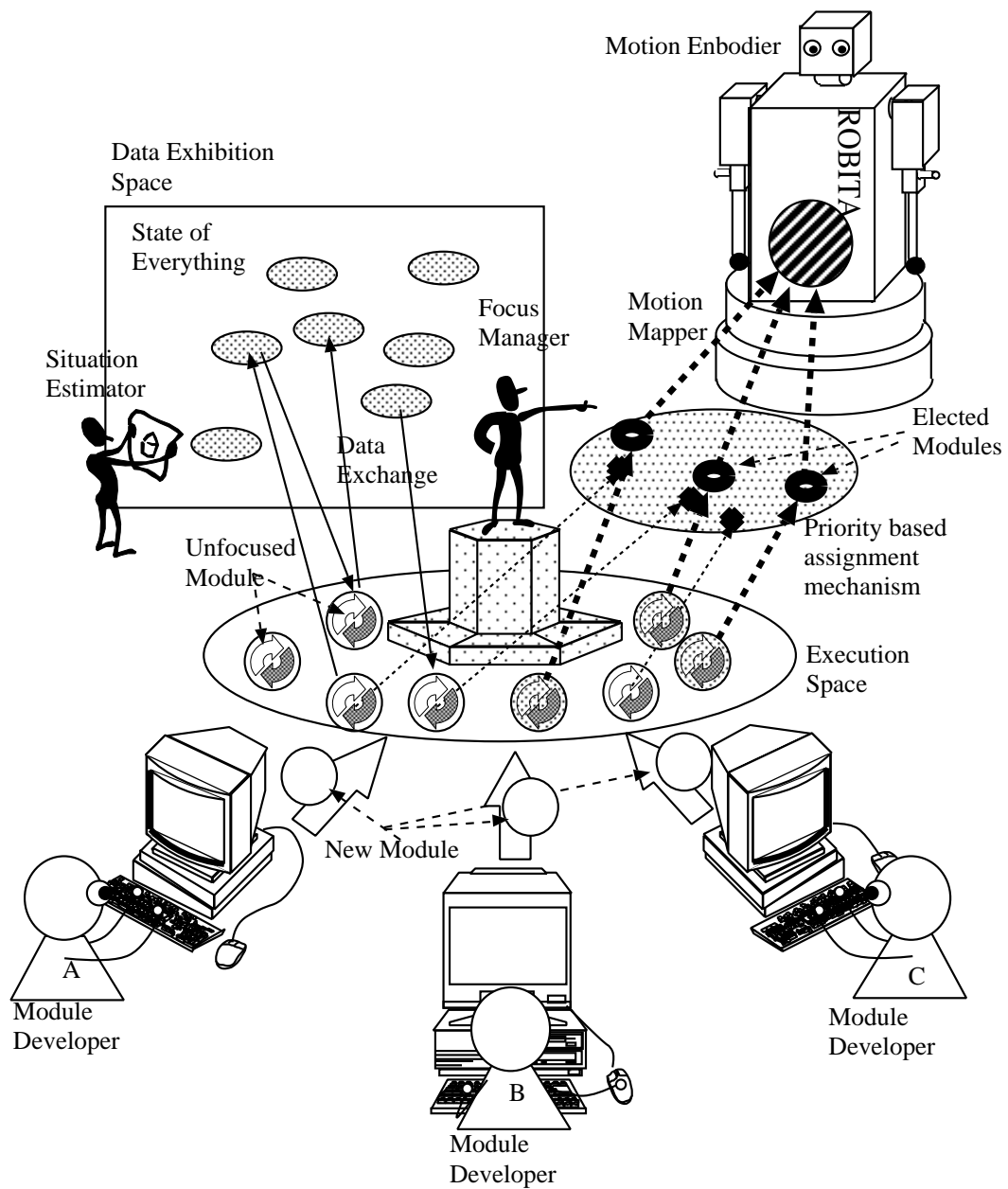


図 3.2: 黒板ベースの統合システム開発と優先度による即時的な行動の選択を実現するシステムアーキテクチャ

組合わせにより構成される (図 2.2) . 情報は , ユーザが定義可能な文字列「タグ」によって区別される . 各々のモジュールは , 上記したサーバに接続されたクライアントであり , ソケット通信によってサーバにアクセスする . 高級開発言語とのインターフェースとして , C++ , Perl , JAVA 用のクライアントライブラリを用意した . 各々のライブラリはクラスライブラリとして定義され , 言語による API の差はない .

黒板に書き込まれるデータは , 浮動小数点型 , 文字列型などの基本的なデータ型によって定義される . 文字列型のデータの場合 , 開発者間の合意の元であればどのようなフォーマットのデータも書き込むことが許される .

### 3.5.2 タグの命名規則・グループ購読

本システムにおいては , ソフトウェアのソースコードのバージョン管理 , 並行開発などの為に , UNIX の標準的なバージョン管理システムである CVS の機能を利用した . 各々のモジュールにおいて定義するタグの名前には , 「CVS に登録されたモジュール名 + ユーザ定義文字列」とする命名規則を開発者に対して課した . この命名規則により , CVS に登録されたモジュール間では , 名前の重複が起こることはない . 一般に CVS ツリーの構造はモジュール間の関連性の距離によって決定される . システムインテグレーションを考えると , モジュール間の関連性が高いほど頻繁にデータを交換する必要がある . データの「タグ」の命名規則を CVS ツリーの構造に習ったものにした上で , 正規表現によるタグのグループ指定での購読サービスを提供することで , ツリーのノード単位のメッセージ購読を可能としている (2.4.7 節参照) .

### 3.5.3 モジュールの優先度管理機構

各モジュール間の協調処理のために , 優先度によりモジュールの選択を行う .

ロボットへの動作のマッピングを希望するモジュールは , マップサービスにモジュール名をあらかじめ登録しておかなければならない . 登録された複数のモジュールのうち , どのモジュールが実際にマップされる権利を得るかは , 以下の手順によって判断される .

1. 状況判断スクリプトによる現在の状況判断．
2. 判断された状況における，モジュールの優先度記述の取得．
3. 優先度に基づくマップ権の判定．

3. の優先度に基づくマップ権の判定において，優先度の高いモジュールから順に，マップ権要求のスキャンが行われる．優先度の高いモジュールがマップ権の放棄を行った場合，次の優先度のモジュールに対するスキャンが順々に行われる．

#### 3.5.4 状況判断機構

本システムは，状況の判断とその状況における優先度を元にした行動選択の二層の行動選択を行うことを特徴とする．状況に応じて各モジュールの優先度を動的に管理するためには，システムが置かれた状況を判断する必要があり，そのための情報源が必要となる．状況判断のための情報源として，情報開示サービスの黒板上のデータの値を利用する．状況監視機構は，黒板上の複数のデータを常に監視し，それらのデータの値の組合せにより状況を判断する（図 3.3）．

監視すべきデータセット，データの値の組合せと状況の対応関係，各状況におけるモジュールの優先度は，XML 形式で状況依存スクリプト内に記述される．状況依存スクリプトにおいて利用できるオペレータの種類や記述法は，3.6.2節において例示する．

全体情報からの状況の抽出に関して，本システムにおいては，スクリプトにおいて指定された部分情報から，適合する状況をリストアップし，適合するすべての状況において，記述の中で指定されたモジュール群を動作候補とした．

#### 3.5.5 動作のマップ機構

動作のマップ機構においては，状況判断機構と優先度管理機構において選択されたモジュールの出力する動作候補を実際の動作にマッピングする．マッピングは，タグの命名規則と黒板のグループ購読機能を用いて行われる．各々の動作候補を生成するモジュールは，自分の生成した行動を「自分のモジュール名 + ”’ + 操作したいのタグ名」で示されるタグに書き込む．マップ機構は，選択したモジュール

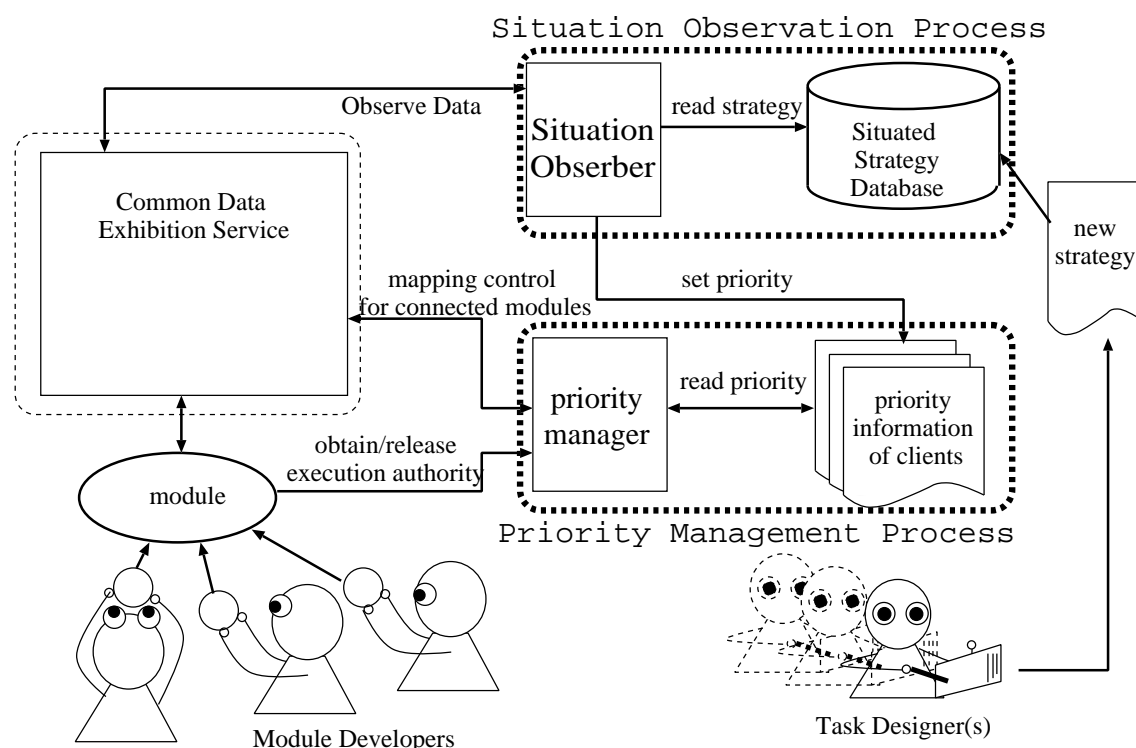


図 3.3: 優先度管理機構及び状況監視機構

名とタグ名の書き込まれたノードをグループ購読機能を用いて購読し，タグ名を変換して再度黒板上の実際の動作を指定するタグに書き込む．その概要を図 3.4 に示す．

## 3.6 システムの運用

### 3.6.1 システム構成

上記検討に基づき，ロボットと複数の情報処理 PC からなる実験システムを構築した (図 3.5) ．

ロボットは，頭部に 2 台のカメラ・2 台のマイク・2 本のアームを持ち，移動台車による移動が可能である．このロボットに，CVS サーバと情報公開サーバを中心に 7 台のクライアント PC を接続した．モジュール作成者は，これら 7 台のクライアント PC 上に各々の作成したモジュールを動作させる．



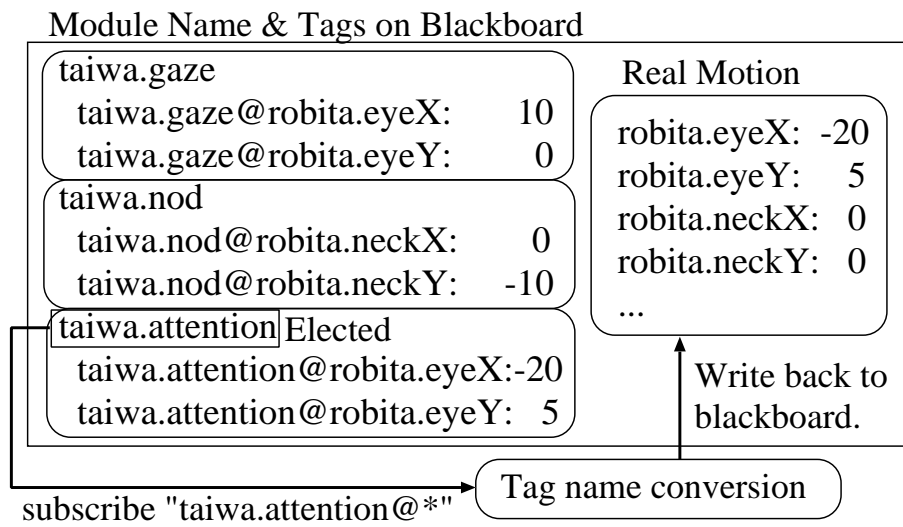


図 3.4: 動作のマッピング機構



図 3.5: システム全図



### 3.6.2 状況切替の動作例

本節では、状況切り替えと即時的な動作選択の動作例を紹介する。対象とするのは、ユーザとの対話を行なうタスクであるが、タスクの実行中に、ロボットの視界内で対話者以外の第3者がロボットに向かって手を振る動作を行う視覚的なイベントと、ロボットの近くで大きな音を立てるという聴覚的なイベントが発生する。ロボットは、ユーザとの対話を行ないつつも、これらのイベントに対処し、状況によってはタスクを遷移させる必要がある。

タスクにおいては、対話タスクにおける発話者の顔方向を追跡するモジュールと、対話内容からうなずき動作を行うモジュール、及びアテンションを向ける処理を行うモジュールの3個が頭部(首)及び眼部カメラ(視線)を同時に使用することになる。今回の実験では、アテンションイベントの発生時に対話タスクを実行するモジュールからアテンションイベント処理モジュールへ実行権限が移行するよう状況依存スクリプトを記述した(図3.6)。

アテンションイベント検知モジュールからの検知結果が黒板に書き込まれ、状況監視プロセスがアテンションイベントの発生を認識し、スクリプトに記述された各モジュールの優先度を優先度管理プロセスに送信する。優先度管理プロセスは、優先度情報から、3つのモジュールの中から、アテンション処理実行モジュールの動作へのマッピングを開始する。

状況切り替えの動作例を図3.7に示す。認知プロセス、行動生成プロセスが並列的に動作する中で、状況に応じた優先度設定に基づく行動生成が行なわれていることが確認できた。

### 3.6.3 システムの稼動状況

これまでのシステムの稼動期間において、ロボットに、音声・音響・画像・言語処理を行いながら多人数の話者との会話に参加するシステム[松坂01]や、状況に応じた表情生成を行いながらユーザとの対話を行うシステム[TMIK00]、ロボットアームを通した指示表現と音声による言語表現を用いてユーザと空間的情報を共有する対話システム[古川99]、インターネット経由でのロボットの遠隔操作システム[中野99]などの多数のシステム[vid]を実装する機会を得た。

```

<Situation>
  <Conditions>
    <!-- Looking at event -->
    <Condition Tag="(image|speech).*.event"
      Operator="==" Value="slap">
  </Conditions>
  <Priorities>
    <Priority ModuleName="taiwa.attention"
      Priority="100"/>
    <Priority ModuleName="taiwa.gaze"
      Priority="80"/>
    <Priority ModuleName="taiwa.nod"
      Priority="50"/>
  </Priorities>
</Situation>
<Situation>
  <Conditions>
    <!-- Face towarded user -->
    <Condition Tag="image.tracker.status"
      Operator="!=" Value="stable">
  </Conditions>
  <Priorities>
    <Priority ModuleName="taiwa.gaze"
      Priority="90"/>
    <Priority ModuleName="taiwa.nod"
      Priority="50"/>
    <Priority ModuleName="taiwa.attention"
      Priority="10"/>
  </Priorities>
</Situation>
<Situation>
  <Conditions>
    <!-- Agreeable user utterance -->
    <Condition Tag="image.tracker.status"
      Operator="==" Value="stable">
    <Condition Tag="taiwa.content.attitude"
      Operator="==" Value="agreeable">
  </Conditions>
  <Priorities>
    <Priority ModuleName="taiwa.nod"
      Priority="90">
    <Priority ModuleName="taiwa.gaze"
      Priority="70"/>
    <Priority ModuleName="taiwa.attention"
      Priority="10"/>
  </Priorities>
</Situation>

```

図 3.6: 記述した状況依存スクリプト (主要部)

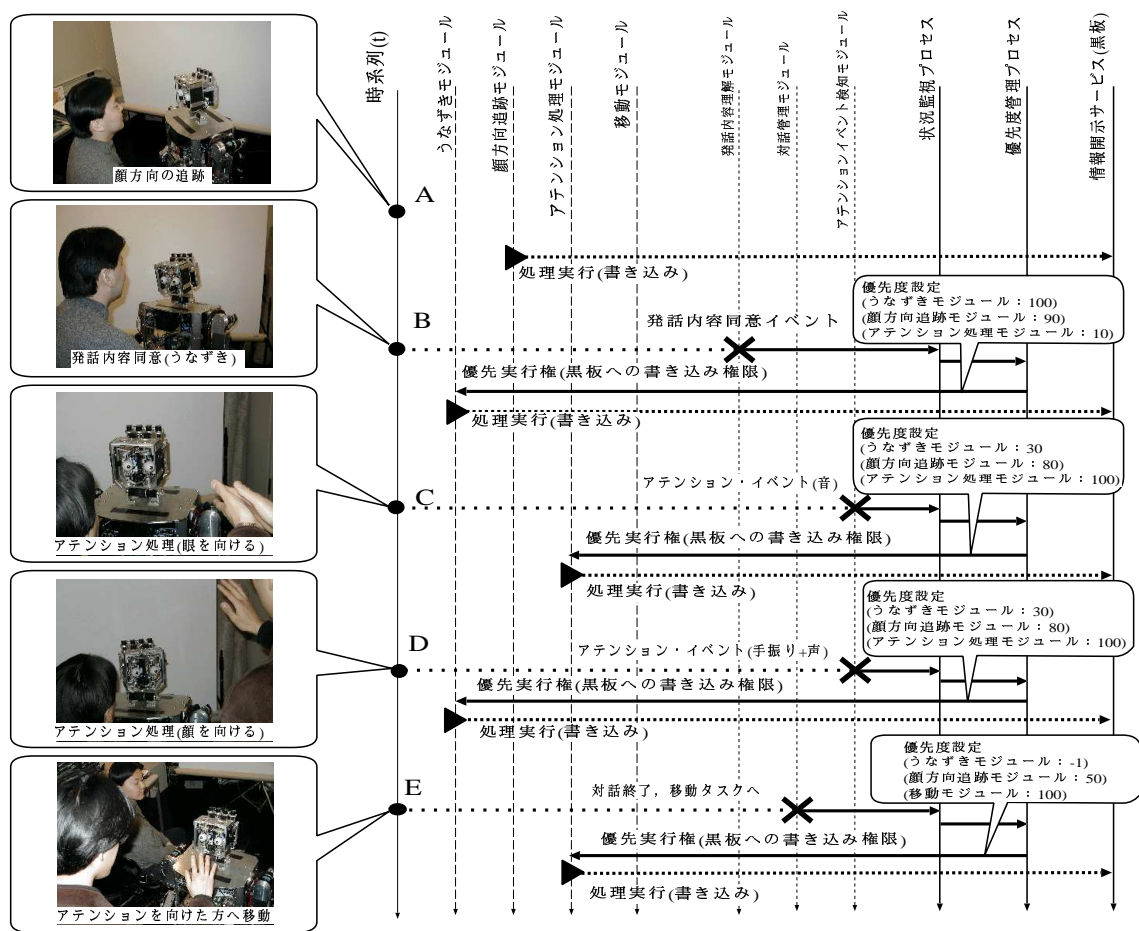


図 3.7: システムの動作状況

開発者となったのは、音声画像処理・パターン認識・人工知能・機械制御などを専門とする学生たちであり、どのシステムも平均して同時に 8 人ほどの開発者がシステム開発・統合作業に関わった。現在開発期間は 3 年を超えようとしている。この間、15 人ほどの開発者がシステムにかかわってきた。

開発者たちは、明示的なシステム設計者が存在しない状況で、開発者同士の情報交換によりシステムを組み上げていった。各々のモジュールの統合作業の開始から、システムが満足のいく動作をするまでにかかる時間は、長い場合で 1 ヶ月ほどであり、特に既存のシステムの拡張や手直しによって対応できる課題である場合は、1～2 週間の短期間でシステムを構築することができた。

### 3.7 議論

本章においては、ロボットのバザールの開発において、黑板モデルに基づく機能分割的な開発と、黑板の監視に基づく状況判断、分割された各々の状況における優先度記述に基づく階層的行動生成を行なうシステムアーキテクチャを提案し、実装した。

従来ロボットの開発方法として、さまざまな方法が試みられてきた。

ひとつの方向としては、ロボットが対面するそれぞれの状況（移動を考えた場合「ドアを開ける」「ものをよける」「障害物を動かすなど」）に対して、その状況に対処するように特化してつくられたモジュールをシステムに付加していくという考え方がある。これら考え方に基づき、状況に依存したモジュールをつなぎ合わせることで、行動を拡張していけるアーキテクチャがいくつか提案されている [IKKI99][星野 01]。この方法でシステムを開発することを考えた場合、状況の記述の方法が問題となる。多くの場合、状態の遷移をマルコフモデルによって記述する方法が取られるが [IKKI99]、多機能かつダイナミックな動作をするロボットを考えた場合、取り得る状況は非常に多様になる。そのような場合、取り得る状況とその遷移をすべて記述しきれぬか疑問が残る。

ロボット開発のもうひとつの方向として、並行動作する複数の機能モジュールによってシステムを作り上げてゆくというアプローチがある。Brooks によるサブサンクションアーキテクチャ [Bro86] は、下位層のモジュールに上位層を積み上げ

ていくことで、全体の動作を作り上げていくという手法であり、システムを機能ごとに縦列的に接続されたモジュールに分割することの得失について論じた初めての試みであったと考えられる。機能分割的にシステムを開発することを考えた場合、要求される機能の範囲が広く、ロボットの取り囲まれる状況が大きく異なってくる問題に対して、いかにモジュールを分割・階層化するかが問題となる。文献 [Bro86] において実現されたロボットの機能は、障害物回避を行ないながら進行する機能に限られている。複数の階層化システムを並列に接続することで多機能化を指向した試み [NN98] もあるが、モジュールの分割・階層化における構造設計の困難は依然として残っている。

ロボットが実世界で生起する様々なイベントに対処しながら行動するためには、状況に適した行動が即時に選ばれる必要がある。元村らは、状況に適した行動の選択を行うために、信念ネットワークを用いて選ぶべき最適な行動を推定する方法を提案した [本村 99]。これら、信念ネットワークや契約ネットワーク [Smi80] などの方法は、状況判断をモジュール化し組み合わせた上で最適な状況を推定できる利点を持つ。しかしながら、最適な状況の推定に効用関数の繰り返し計算を必要とするため、即応性を得ることは難しい。

即応性を実現することを考えた場合、システムはより絞り込まれた情報を用いて、より少ない判断基準で状況を判断し行動に移る必要がある。それゆえに、即応性の実現と正しい状況判断の実現には、一般にトレードオフの関係がある。これら即応性の実現と正しい状況判断の実現を両立させるために、システムを階層化させることが一般に行われる。システムを階層化させることで、上層における正しい状況判断の結果を用いて、下層の即応性を実現するモジュールの判断を補助し、より正しい判断をするように導くことができる。ロボット開発における代表的な階層化手法としては、Gat によって議論された 3T アーキテクチャ [Gat97] などがあげられる。

複数モジュールによる協調的行動生成に関して、提案アーキテクチャは、分割された状況において、静的な優先度設定を行なった上で、モジュール個別の判断による動的な行動生成をおこなう。外部から効用関数を決めて調停的な行動生成を行なうアーキテクチャが多く提案されているが [伊藤 01]、特にロボットなどの応用を考えた場合、システムの処理速度までを考慮に入れてアーキテクチャの妥

当性を判断する必要がある．情報の公開に優れたモデルを採用したとしても，公開される情報には限界があり，モジュールの内部状態すべてが公開されているわけではない．例えば，事象と事象の間のコンテキストなどの情報は，組合せにより膨大な数の情報が生成される可能性を持った情報であり，モジュールの外に公開したり，調停作業時に問い合わせすることは，通信・保持コストから考えて現実的でない．モジュールが活動すべきか否かを本当の意味で把握しているのは，モジュール自身であるとの考えのもと，それらの判断を状況によってまとめられた小グループの中でモジュール自身によって行わせたのが本アーキテクチャである．すべてのモジュールに対する問い合わせと，効用関数の繰返し演算を必要とする調停型のアーキテクチャに比較して，関連モジュールへの順次問い合わせのみにより行動が決定される提案アーキテクチャは，システムの応答性の良さに関して有利である．

黒板の監視に基づく状況判断に関しては，本稿において状況判断に用いた方法が最適であるとは必ずしも考えていない．例えば，状況の重なりや判別の難しい状況に対して，本稿では，適合するすべての状況記述で指定された内容を合わせて優先度として用いるという実装を行なったが，これ以外にも例えば，信念ネットワークを用いることで状況の判別を行なう方法 [本村 99] など，様々な方法が考えられる．この問題に関しては，今後の検討課題としたい．

また，本稿において提案した開発の原則（3.3章）は，システムを無限に冗長な方向に向かわせる．現在これら冗長性を許容する接続アーキテクチャの検討を進めているが [於久 01]，開発の極限において破綻が起こることは避けられない．本当の意味でのバザール的な開発を実現するためには，システムをユーザによる自由な開発に任せながらも，さらにそこから不要な機能を切捨て，有用な機能をオーサライズしていく枠組が必要となる．これらオーサライズの枠組も今後の重要な検討課題である．

### 3.8 まとめ

開発者のシステムインテグレーション活動において，バザール的なシステム開発と状況に応じた即時的な行動の選択を可能とするシステムアーキテクチャと，開

発を支援する開発環境について検討を行った．多人数の開発者によるマルチモーダル対話ロボットの実装を通して検討したシステム開発環境が良好に動作することを確認した．



## 第4章 グループ会話に参加する対話 ロボットの構築

### 4.1 はじめに

本章においては、2章、3章において検討したアーキテクチャと開発環境の上に、マルチモーダル対話システムを構築する。音声・画像・音響チャンネルの持つ特性、人間の対話におけるそれらチャンネルの利用の分析を行うとともに、それら分析に基づき、グループの形態での会話に参加可能な対話システムを構築する。

通常我々は複数人と共に話題を共有しながら多対多の会話（以下、グループ会話と呼ぶ）を行なっている。我々がこのようなグループでの情報交換を通じて、意思決定をしたり着想を得たりする機会が多く、グループ会話は人間同士のコミュニケーション形態として非常に重要性が高い。このため、複数の人と活動を共にするロボットシステム、あるいはグループ形態での人間の知的活動を支援する情報システムのように、人間同士のコミュニケーションに加わるシステムを実現しようとするならば、システムはグループ会話に参加できる能力を備える必要がある。

しかしながら、人間の行なうグループ会話に関する心理学的立場の研究や、人間のグループ会話を支援するシステムについての研究はいくつかなされているものの [Ver98, 中西 98]、グループ会話に参加する会話システムを実現する立場の研究は皆無であった。従来の対話システム研究は、人と機械が一对一で情報交換することを前提としたものがほとんどである [白井 85, GBG<sup>+</sup>94]。いくつかのシステムは複数のユーザを相手にすることを目標として設計されているが [nua, 松井 00]、それらにおいても個々の局面においては一人のユーザと対話をしているにすぎず、グループ会話の重要な要件となるユーザ間の対話に対し関与できない。これらは、グループ会話というよりは一对多の対話として分類されるべきものである (図 4.2)。人間同士の対話をモニタし、対話内容に応じて適切な情報を提供することを目的とするシステムも存在するが [NT94, 城塚 98]、これらのシステムは対話をモニタ



する立場に徹しており，システムが複数の人間と対等な立場で会話することを目指すものではない．

グループ会話に参加可能なシステムを実現するためには，従来の対話システムにおいては無視できたさまざまな問題を扱わなければならない．対話システムにおいては，個々の局面において各会話参加者がどのような役（発話者，聴者など，[4.2](#)に詳述．）で会話に臨んでいるかといった会話の状況をシステムが理解するとともに，次の局面でそれがどのように変化をするかを適切に予想しながら会話を進行することが必要である．しかし，システムがただ一人のユーザを相手にする場合や，2人のユーザの対話をモニタする立場に徹する場合などでは，これらは発話者を手がかりとして一意に決めることができた．これに対し，グループ会話では観察者（[4.2](#)参照）なる役が生じることで会話参加者間の関係は複雑化し，会話の状況は自明ではなくなる．円滑に会話を進行させるためには，システムは正確に会話の状況を理解できるとともに，自らも状況の曖昧性を減少させる適切な行動をとれることが必要となる．また，観察者である会話参加者が将来発話者として会話に参加する機会を得るためには，会話に対する協調的な態度を積極的に示すことで，現発話者の注意を喚起する必要がある．

これらの問題は対話システムにおいて本来重要な課題であるべきにも関わらず，従来それが問題として現れない特殊な問題設定において対話研究が進められてきたことによって，十分な検討がなされて来なかった．

そこで本研究では，グループ会話の特徴分析に基づいて，これに参加するシステムに求められる機能について検討を行ない，これを人間型のロボット ROBITA (Real-world Oriented BImodal Talking Agent: [図 4.1](#)) を用いて実現することを試みた．本稿では，分析・検討の結果およびそれに基づいたシステムの設計指針について述べるとともに，試作したシステムの概要とその動作例を紹介する．また，実ロボットを用いてグループ会話を実現することの意義についても検討を行なう．

## 4.2 用語の定義：グループ会話と会話参加者の役

本研究で扱うグループ会話とは，一つ的话题を複数人で共有しながら発話を交換しあっている状態を指すものとする．このような会話では，通常会話参加者の

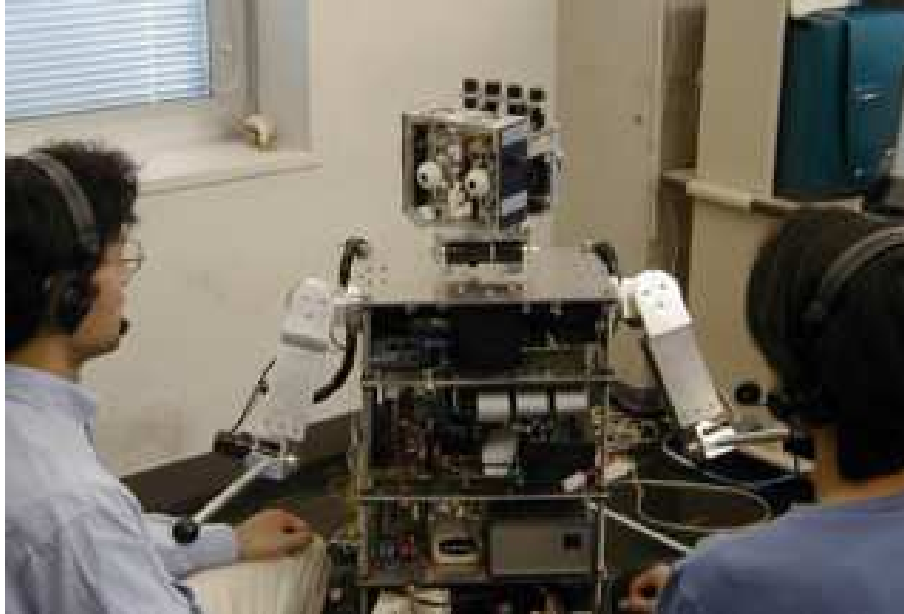


図 4.1: グループ会話の風景

誰か一人が他の誰かに注意を向けて発話を行い，それを他の会話参加者全員で聞くという形が繰り返されるのが一般的である．この形態で行なわれる会話に対し，本稿では以下の用語を定義する．

現在発話権を持って発話している人を発話者と呼ぶ．「発話権を持って発話する」とは，メッセージの送り手として発話することを指し，相槌などを挟んでいる人はこれに含まれない．発話者の発話にあたり注意を向けられた人を主たる聴者と呼ぶ．また，発話者と主たる聴者の対を当事者と呼ぶ．当事者間の発話の交換を対話と呼ぶ．当事者となることを前提として当事者の対話を周囲から観察している人を観察者あるいは従たる聴者と呼ぶ．当事者，観察者を合わせて会話参加者となる．発話者，主たる聴者，観察者，従たる聴者などを会話参加者に与えられた役と呼ぶ．

ある局面で誰がどの役を担っているかを会話の状況と呼ぶ．これらの用語を用いるとき，グループ会話とは，参加者の中で当事者が入れ替わりながら対話を繰り返す状況ということができる．図 4.3 に会話参加者の役の関係を模式的に示す．

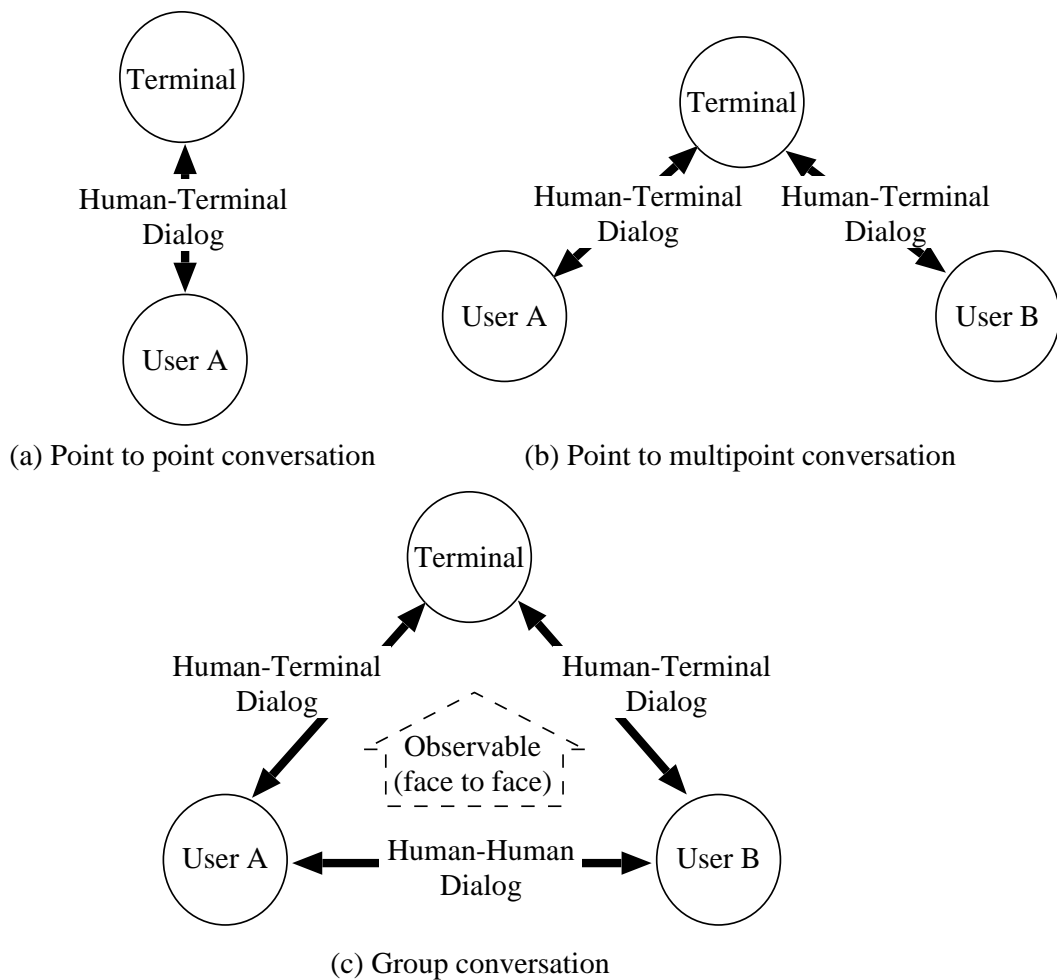


図 4.2: 多人数会話の形態

## 4.3 グループ会話の特徴

グループ会話を特徴付ける最も重要な点は、従たる聴者（あるいは観察者）の存在である。これが基となって以下のような特徴が生じる。

### 4.3.1 会話状況の曖昧性とその理解・明確化の必要性

一対一や一対多の対話では、観察者は存在しない。このため、自分自身の役は常に自明であり、会話の状況に曖昧性はなかった。これに対しグループ会話においては、観察者なる役が生じることによって、一つの局面でとりうる会話の状況に曖昧性が生じる。発話者でない場合は、自らが演じるべき役として、主たる聴

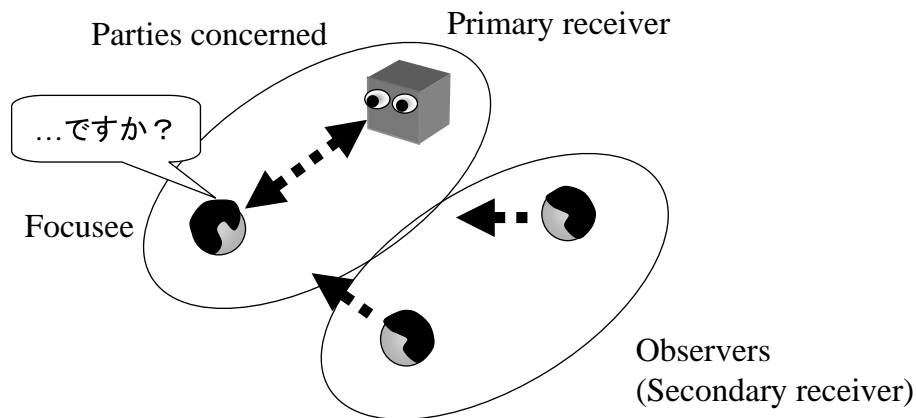


図 4.3: グループ会話における参加者の役．会話参加者は，今現在発話している人物（発話者）とその発話を受けている人物（主たる聴者）の対（当事者）と，その二人を周囲から観察している人物（観察者もしくは従たる聴者）に分類できる．

者と観察者の二つの可能性が生じる．

役に応じて求められる役割が異なるため，会話参加者が役の判断を誤れば円滑な会話の進行は望めない．例えば，観察者として会話を静観すべきことを理解できていなければ，不適切なタイミングで発話を開始してしまうことになるし，主たる聴者として次の局面で発話することが望まれていることを理解できていなければ，会話に不自然な沈黙が生じる．このため，会話参加者は自らの役を適切に判断する能力が求められるとともに，他の会話参加者が戸惑わないよう，相手に期待している役を明確に伝えられることが望まれる．

#### 4.3.2 発話の広報性と当事者間対話の理解に基づく発話の必要性

一対多の対話では，発話はその局面において何らかの方法で決められた唯一の聴者に向けて行なわれる．従たる聴者は存在せず，当事者以外は発話内容に気を留める必要がない．これに対し，グループ会話では発話者は主たる聴者を一人選定するものの，発話内容自体は従たる聴者を含む参加者全員に伝わるのが発話の前提となる．このとき，従たる聴者に直接投げかけられた発話ではなくても，次の局面でその内容の理解を前提とした発話が求められることがある．このため会話に参加するためには，従たる聴者も当事者間の対話内容の全体を理解し，会話の流れを把握する必要がある．従たる聴者が会話の流れを把握した上で適切な情

報を提供すべく積極的に発話権をとりに行く行動をとることができれば，グループ会話はより活性化されたものとなる．

#### 4.3.3 アウェアネスの不均衡性とその改善努力の必要性

アウェアネスとは，自然なコミュニケーションのきっかけとして必要とされる，「他者の存在・行動を知っていること」[VVvdV97, DB95, 中西 98, 本田 97]であるが，ここでは特に会話参加者が他の会話参加者を対話相手として意識していること，あるいはその意識の程度を指してアウェアネスという言葉を使う．

グループ会話において，観察者に対する当事者のアウェアネスは，当事者間のアウェアネスに比べ低くなり，不均衡となる可能性を持つ．即ち，観察者の態度いかんによっては，当事者だけで対話が進み観察者が疎外される危険性を持つ．グループ会話においては，観察者の様子は当事者の視界に写るのであるが，観察者の行動が当事者に対し会話に対する興味を感じさせないものであるならば，コミュニケーションのきっかけをつかむ（話しかけられる）ことは難しい．会話参加者全員が対等な関係をもって，一体感のある会話を行なうためには，観察者は当事者の高いアウェアネスを喚起しなければならない．

### 4.4 システムの設計

前節では，グループ対話を実現するシステムが持つべき能力や期待される役割について述べた．本節ではそれらをどのように実現するかを述べる．

#### 4.4.1 実ロボットの身体表現を用いた会話状況の明確化

前節に述べたように会話状況は曖昧なものであるから，参加者はその明確化を行なう必要がある．代表的には，発話者は主たる聴者として誰を選んだかを明確に伝えることが望まれるし，聴者は誰に発話者となってほしいかを明確に意思表示する必要がある．通常，人間同士で行なう対面対話においては，これらの意思伝達には各会話参加者の視線が重要な役割を果たしている [Ver98, 横山 99]．話者は相手を正視することによって，その相手に対して発話を行なっていることを表

現する．聴者は発話者を見つめることによって，その相手から情報を期待していることを表現する．通常これらの会話状況を明確化するための意志伝達は無意識のうちに行なわれる．これを人同士が通常用いる方法と異なる手法により伝えることにすると，処理が意識に登り違和感を感じる可能性がある．そこで我々が構築するシステムにおいても，擬人化した顔・身体を持つロボットの身体表現によって，会話状況の明確化を行なうこととした．具体的には以下の方法による．

#### a. 視線による明確化

システムが発話者となる場合主たる聴者に視線を向け，聴者となる場合発話者を正視することとした．発話が途切れたならば，予想される次発話者に視線を向け，次発話者として期待していることを明確化することとした．

#### b. 体の向きによる明確化

発話者の場合には，体の向きにも意味を持たせた．視線が実際に発話を向けた人を表現するのに対し，体の向きは対話相手として意識している人を表すものとした．通常両者は一致するが，例えば従たる聴者からの割り込みを制止する場合などでは，体を主たる聴者に向けたまま顔だけを従たる聴者に向け「少し待ってください」と発話することになる．このことによって，発話相手としての意識は主たる聴者に残し後から何か伝えることを表現しながら，「待て」という指示を割り込み者に伝えることができる．

#### c. うなづきによる明確化

聴者となる場合，聴者として動作していることを明確化するために，発話者の発話に合わせてうなづく機能を持たせた．通常，この目的には「はい」というような相槌を用いる方法が研究されている [War96, HMNK98] が，グループ会話で大勢の人が一斉に相槌をうったのでは騒がしくなり不適切である．よって，主たる聴者として振舞う場合は相槌とうなづきを，従たる聴者の場合にはうなづきだけを行なわせた．この区別は，単に適度な静寂性を保つだけでなく，システムが自身が主たる聴者と従たる聴者のどちらとして振舞っているかを明確化する役割も持つ．

#### 4.4.2 マルチモーダルな情報処理による会話状況の理解

前節に述べたように，会話参加者は状況の明確化を主に身体表現により行なっているため，システムは画像処理によってそれを理解する必要がある．会話の問題は当然音情報処理の問題を含むため，グループ会話システムは必然的にマルチモーダルシステムとなる．

##### a. 発話者の推定

発話者の推定に関しては2つのモードを用意した．最初のモードでは，ロボットに設置したマイクを使う．まず音源定位 (4.5.2参照) を行ないおよその到来方向を求める．次に，画像処理によってその方向にいる人物を探し，結果を発話者として定める．

もう一つのモードでは，各会話参加者に個別に持たせたマイクを使う．この場合は，まず誰がどのマイクを使い，どの位置にいるかの対応づけを行なう．各マイクに入力された声を話者認識 (4.5.2参照) し，各マイクと人の対応付けを行い，さらに全体を見渡しながら各話者の顔認識 (4.5.2参照) を行うことで，人とその位置を対応付ける．このようにするとき，入力のあったマイクを持っている人を発話者と決めることができ，さらにどちらに視線を向けるべきかも決めることができる．

現状では，音声認識をロボットに搭載したマイクで行なうことは困難であるため，各参加者にマイクを持たざるを得ない．このため，後者の方法が現実的といえるが，将来音声認識をロボットに搭載したマイクで行なうようになれば，前者の方法が重要となる．

##### b. 主たる聴者の推定

主たる聴者の推定については，発話者の視線を手がかりとする．視線の抽出は難しいので，実際にはこれを顔向きの認識で代用した．発話者の顔が向いている方の聴者を主たる聴者とし，それ以外を従たる聴者と判断した．発話者の視線は発話の終りに差し掛かったところで高い安定性を示す傾向がある [Ver98, WST94]．そこで，主たる聴者の判断も発話の終了時の顔向き情報を重視した．

##### c. 次発話者の推定

前節 a に述べたように，従たる聴者は次話者を予測する必要がある．4.4.5節 1) の原則からいえば，次発話者は現在の主たる聴者である可能性が高い．よって，発話者の顔向きから決めた現主たる聴者を次発話者とした．



#### 4.4.3 当事者対話の理解に基づく従たる聴者の割り込み発話

4.3.2に述べたように、グループ会話では当事者でない場合にも発話を理解する必要があり、その理解内容を踏まえて発話することが求められる。しかしながら、ユーザはシステムとは設定タスク内の対話をするのに対し、ユーザ同士の間ではかなり広範な話題について対話する傾向があり、ユーザ間の対話を含む全発話の理解をすることは極端に高度な要求である。そこで本システムでは、極限られた文章に選択的に反応するようにした。具体的には、「このシステムは4歳です」「英語が話せます」など、システムを説明する表現だけをとりあげ、これを選択的に理解することとした。また、仮に事実と反する説明が行なわれた場合、それを訂正するための割り込みを行なうようにした。

#### 4.4.4 アウェアネスの改善

4.3.3に述べたように、各々の参加者の視界からは、今現在注目している相手だけではなく、そのほかの参加者たちの存在と行動が見て取れる。会話に参加できるきっかけを増やす為には、会話の状況に応じて適切な行動をとることで、他の参加者の自らに対するアウェアネスを向上させる必要がある。これらアウェアネスの喚起の問題は、自らの意志の表現という観点から4.4.1で述べた会話状況の明確化と密接に関わっている。本システムにおいては、発話交替毎に会話において主たる役割を負っている発話者を視線で追うと同時にうなずきをすることによって、会話に積極的に参加する姿勢を他の参加者たちに示すこととした。

#### 4.4.5 システムの振舞い

Sacksらは、グループ会話において、次の原則が成り立つことを指摘している：1) もし現在の話し手が次の話し手を指定するならば、その選ばれた相手は次に話す権利と義務を持つ。2) もし現在の話し手が次の話し手を指定しないならば、最初に話し出した人が発話権を得、話者交代はそこで起きる。3) もし現在の話し手が次の話し手を指定せず、ほかの参加者が話さないならば、現在の話し手は話し続けることができる [SSJ74]。これらの原則を満たすよう、発話権の遷移を適切に制御する能力が必要となる。



以上を鑑み，システムには，担うべき役に応じて以下の振舞いをさせることとした．

a. 発話者としての振舞い：

- a) 発話内容と主たる聴者を決め，主たる聴者に体と視線を向けて発話を行なう．
- b) 発話途中に従たる聴者から割込みがあった場合，現在の当事者間の対話を中止できる状態であれば，割込み者に発話権を委譲する．
- c) 中止できなければひとまず対話を中断し，割込みを制止した上で，元の対話に戻る．制止は，体を主たる聴者に向けたまま視線を割込み者に向け，少し待つよう発話することで行なう．
- d) 発話の終了によって，発話権を主たる聴者に委譲する．

b. 主たる聴者としての振舞い：

- e) 発話者を探し，発話者に視線を向ける．
- f) 相槌とうなづきを交えながら発話を受理する．
- g) 発話が終了したら，発話者となる．

c. 従たる聴者としての振舞い：

- h) 発話者を探し，発話者に視線を送る．
- i) うなづきを交えながら発話を受理する．相槌はいれない．
- j) 事実と反する説明があったら，対話に割込んでそれを訂正する発言をする．
- k) 発話者が発話権を委譲したら，その時点における主たる聴者を次発話者と予想し，そちらに視線を向ける．発話者の発話権の委譲については，「××ですか？」など発話権を委譲を表す典型的な文末表現を含む文が発話された時点，あるいは発話者の発話終了後一定時間誰も発話しなかった時点で，委譲があったものと判断する．

## 4.5 システムの概要

本節では，前節に述べた振舞を実現するために用意した，ハードウェア，ソフトウェアモジュールの構成，および各ソフトウェアモジュールの内容について述べる．

### 4.5.1 ハードウェア

ロボットは人間を模した顔と身体を持ち，会話状況の明確化のための身体表現を行なう．二つの眼球にそれぞれ2自由度，首にピッチとロール軸の2自由度，腰に1自由度，両腕に各4自由度持ち，対話において必要とされる簡単な身体表現が可能となっている(4.4.1参照)．ロボットは全方位に移動可能な台車の上に載っており自律移動が可能である．ロボットの眼球部には小型のCCDカメラ2台を搭載しており，画像を取り込むことができる．ロボットの顔両脇には2系統のマイクが設置されており音源定位に用いられる．ロボットの胴体部には，バッテリー，制御機器の他，制御用，音声合成用の2台のPCを積んでいる．ロボット外部には，音声認識や画像認識用のPC，WSを置く．これら複数の計算機はネットワークを介して情報交換を行なう．

### 4.5.2 ソフトウェアモジュール構成

図4.4に本システムのソフトウェアモジュール構成を示す．ソフトウェアモジュールは，音声・画像処理を行なう認識系のモジュールと認識結果から身体表現や発話内容を決定する駆動系のモジュールに大別される．

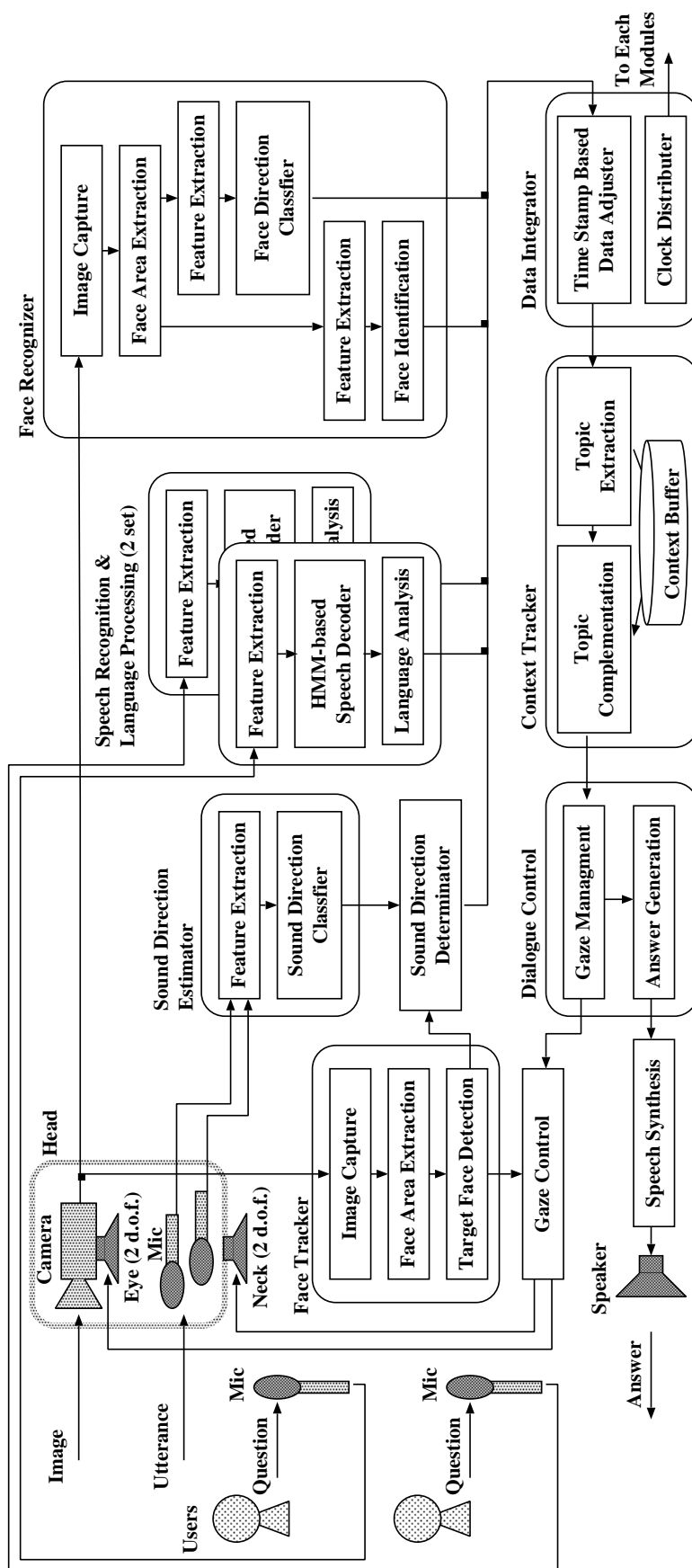


図 4.4: システムのソフトウェアモジュール構成図

音声処理としては、音声がどこから発せられたかを推定する音源定位モジュールと、各話者につき一つずつの音声認識モジュールを用意した。画像処理としては、顔の位置を検出するモジュールと顔・顔方向を認識するモジュールを分けて用意した。これは、視線の制御を高速な顔位置検出によるフィードバックループを形成することで話者とのスムーズなアイコンタクトを実現するためである。このループは、ほかの高度な情報処理（顔・音声認識など）とは独立して動作する。各情報は情報統合モジュールによって統合された後、視線等ロボットの制御、応答文の生成・発話が行なわれる。

以下主な処理モジュールについて概要を述べる。

### 顔画像による個人と顔向き認識

ロボット頭部に設置したカメラによって会話参加者の顔画像を取り込み、各会話参加者の個人認識（認識対象は研究室の学生 20 人）と顔向き識別（識別対象は、正面、斜め前左右 30 度、60 度、90 度の 7 方向）を行なう。

統計的手法によって肌色尤度を求め、肌色領域を顔領域として抽出する。得られた肌色領域を  $16 \times 16$  に正規化した後、独立成分分析にかけて、顔向き認識用には 7 次元、個人認識用には 20 次元の特徴ベクトルに落とす。得られた特徴量を統計的な識別器にかけ、顔向きと個人とを認識する。実環境における顔領域抽出のロバスト性向上のために肌色尤度情報は、フレーム毎に MAP 推定に基づく適応処理が行われる [久保 00]。

こうして得た顔向きの情報は各会話参加者が誰を注目しているかの判断に用いられ、個人情報発話者の同定に用いられる (4.4.2 参照)。

### 音源の定位

頭部の両脇に設置した 2 つのマイクに入力される音の対数スペクトルの差を特徴ベクトルとし、統計的認識手法を用いて 10 度刻みの精度で音の到来方向を判定する。マイクに入力される音の周波数特性は、もともとの音源の周波数特性にロボットの頭部伝達特性の影響が畳み込まれた形となる。よって 2 系統の音の対数スペクトルの差をとれば、音源の特性によらず概ね左右のマイク位置での頭部伝達特性の差が得られることになる。この頭部伝達特性の差は音の到来方向に依存し

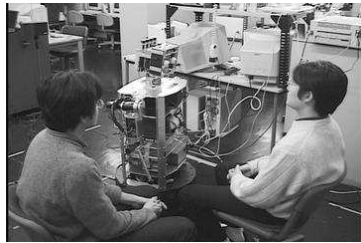
て決まるため，2 系統のマイクの対数スペクトル差を特徴ベクトルとして，統計的なパターン認識手法を適用することで音の到来方向を知ることができる [田宮 99]．音源の方向情報は，発話者の理解に用いられる (4.4.2.a. 参照)

#### 音声の認識と音声応答の生成

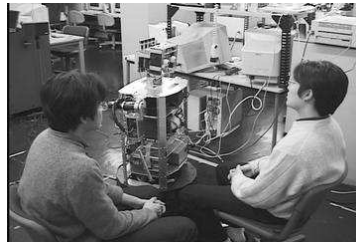
音声認識では，バイグラム言語モデル，HMM 音響モデルを用いた語彙量約七百のフレーム同期の連続音声認識を行ない音声単語の列へと変換する．また，GMM を用いた話者の認識 (対象は顔認識と同じ学生 20 人) を行う [村井 00]．それに続く言語処理では，認識の結果得られた単語列から，キーワード系列をスポッティングし，さらにテンプレート処理によりキーワード列を意味表現へと変換する．各意味表現と，それが入力されたとき行なうべき音声応答と動作の関係は，表形式で与えられており，認識理解処理結果に応じて決められた音声応答と動作を行なう．応答音声の合成に関しては Windows ベースのテキスト読み上げソフト [tos] を用いている．

## 4.6 システムの動作

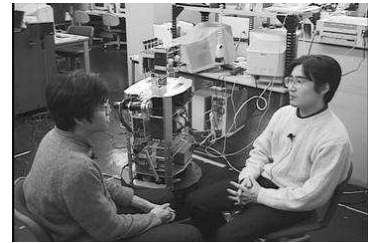
上記のシステムを統合して，グループ会話を実現した．会話参加者数は 2 名 (ロボットを含めて 3 名) である．会話内容としては，ロボットの持つ機能についての一問一答形式の質疑応答とした．各ユーザはロボットの前にならんで座り，任意の会話参加者を相手に発話を行なう．システムは，自由背景，室内照明の条件の下で動作する [vid]．



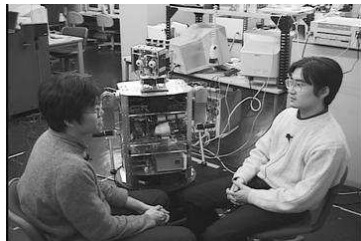
A:こんにちは>R  
R:こんにちは>A



A:何ができますか>R  
R:私は複数の話者と対話ができます>A



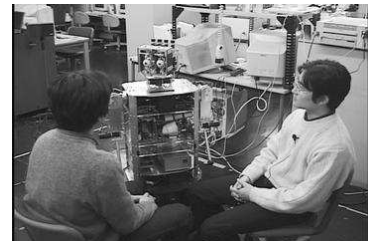
A:こんな感じでしゃべることができます>B



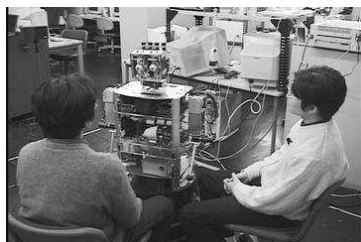
R:(interest)>B  
B:何を聞いても大丈夫ですか>A



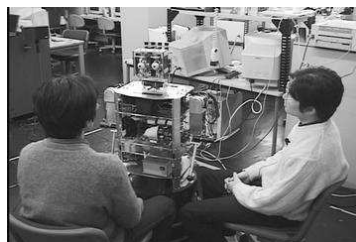
R:(interest)>A  
A:はい、何か聞いてみてください>B



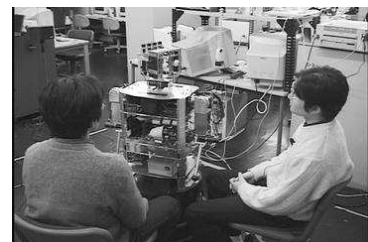
R:(interest)>B  
B:何歳ですか>R



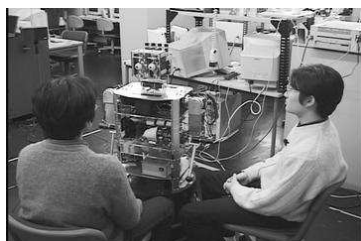
R:私は4歳です>B



B:背の高さは ...>R  
A:すいません>R

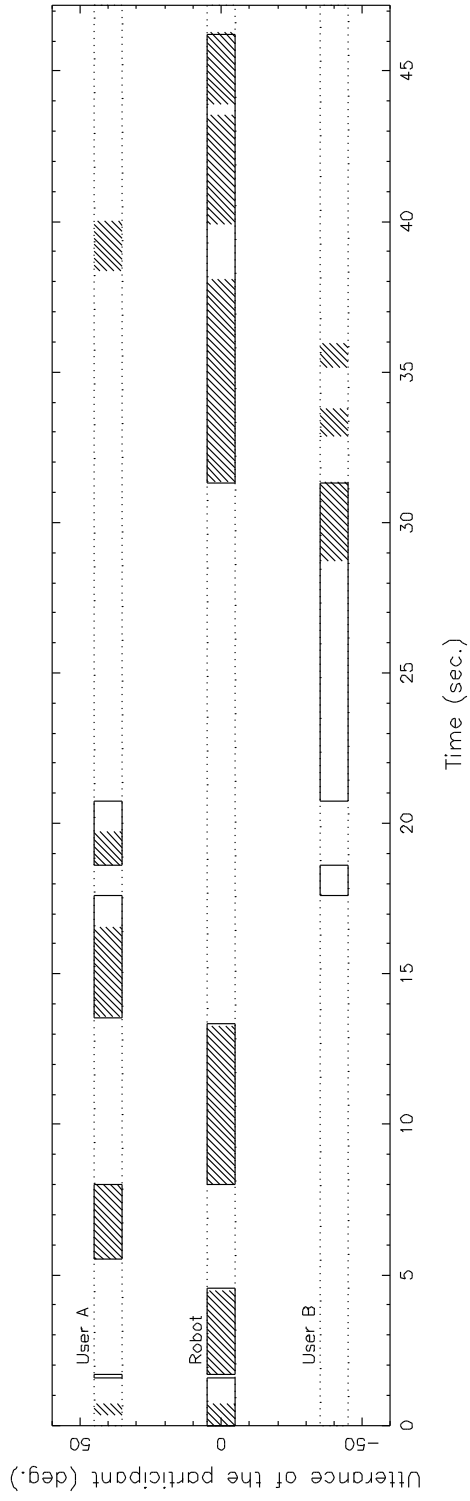
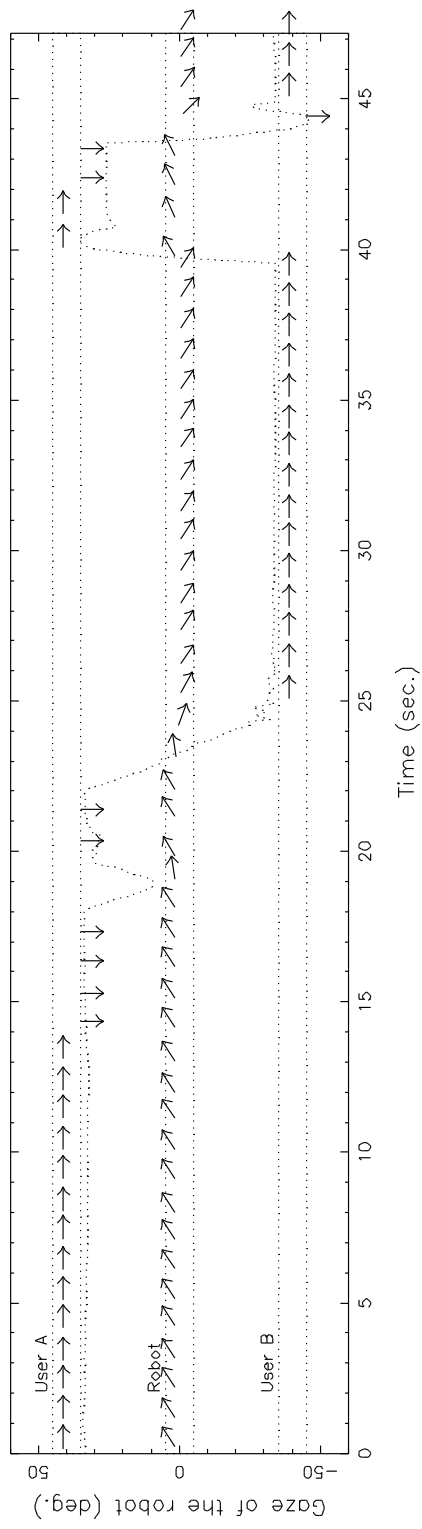


R:(side-glance)  
少し待ってください>A



R:何でしたっけ>A  
...

図 4.5: 会話例



上段:点線はロボットの視線，矢印は上からそれぞれ，認識されたユーザ A の視線，ロボットの視線，認識されたユーザ B の視線  
下段:斜線領域はそれぞれの会話参加者の発話区間，実線で囲まれた領域はシステムが予測した次の発話権保持者

図 4.6: 会話例におけるシステムの内部状態の変化

会話例を図 4.5 に示す．また，会話進行時におけるロボットの内部状態の変化の例を図 4.6 に示す．図上段はユーザとロボットの視線を表す．矢印は，ユーザの段ではユーザがロボットの方向を向いているとき水平になり，ロボットの段では，ロボットが 2 人のユーザの真中を向いているとき水平になる．この例においては 0 ～ 13sec の区間では，ユーザ A がロボットを正視して発話を行なったため，システムはその問いに対し応答を出力している．13 ～ 20sec の区間においては，ユーザ A がユーザ B に視線を向けて発話したためロボットは応答はせず，A の発話終了後ユーザ B に視線を切替えている．40sec の部分はロボットからユーザ B への発話中にユーザ A から割込みがあった例であるが，発話を制止した上で B への発話へ戻っていることがわかる．下段は，実線の囲みで次発話者に関するロボットの予測を表し，斜線で実際の発話者を示している．両者はほぼ一致していることが分かる．

対話実験を通じてシステムの各機能は安定に動作することが確認できた．ロボットは，発話者の顔方向を認識することで，発話が自分に向けられているのか，または他のユーザに向けられているのかを判断し，望まれたタイミングで適切な応答を返すことができた．会話参加者は，身振りや呼びかけを交えながらロボットに語りかけ，また，ロボット自身も身振りで内部の状態を表出することによって，ごく自然な対話をすることができた．さらには，ロボットが観察者の役を担うときも，発話者を予測しながら交互に視線で追うことによって，会話に対する参加意識を表現することができ，対話の当事者たちもロボットを協力的な会話参加者として意識することができた．

## 4.7 議論

前節に紹介したように，ロボットは，対等に近い関係で人間同士の会話に加わり，参加者の間で一体感のあるグループ会話を実現できた．

本システムの特徴的な点として，1) 人間型のロボットを用い，擬人化インタフェースを実現したこと，2) それを 3 次元空間上に構成したこと，3) 聴者のシステムにアウェアネス改善のための行動をとらせたこと，などが挙げられる．

一般にインタフェースは透過的でなければならない [Nor90, 西本 96]．この意味で会話参加者は会話の状況を明確化するなんらかの努力を求められることを既に



述べた．また，会話参加者が観察者の役を担うとき，自分に対するアウェアネスを高めるための努力をすることの必要性についても述べた．これらの目的のためにシステムがとる手法が人間が通常行なう手法と同じであれば，違和感なくシステムの意志が伝わることを期待でき好ましいことではあるが，そうしなければならない必然的な理由はない．例えば，対話が人と機械の一对一の関係で行なわれるのであれば，人間が機械の指定するやり方に合わせることが可能であり，人間はそれに慣れることができる．しかしながら，ここで扱ったグループ会話においては，人間と人間との対話に機械が加わる形で会話が構成される．状況の明確化やアウェアネスの改善のために人間が人間に対しとる手法が，加わった機械の都合で（人間同士にとっては不都合な）機械に合わせた手法になることは考え難い．また，会話全体としての一体感を実現するためには，局面毎にとられる手法に一貫性を持たせる必要がある．これらの意味で，グループ会話においては，擬人化インタフェースを用い，人間と同様な手法で状況の明確化を行なう能力を持たせることは非常に重要と考える．

また，この擬人化インタフェースを3次元の実空間上に実現したことの意味も大きいと考える．擬人化インタフェースとしては，2次元ディスプレイ上の擬人化エージェントを用いた研究が盛んに行なわれている [石塚00]．しかしながら，グループ会話では様々な場所に位置違ったユーザ個々に対して，その位置に依存した情報を正確に伝えることが重要であり，平面的な画像情報によって，これを伝えることは困難である．例えば，視線ひとつとってみても，2次元の情報で特定のユーザだけを見ている状態を全ての位置にいるユーザに対して正確に伝えることは非常に難しい．グループ会話にとって，実ロボットを使ったことは，この観点からも有効であった．

対話システムにアウェアネスの改善を意図した行動をとらせたことも他に例を見ない特徴である．会話実験の被験者によれば，被験者はロボットを協力的な会話参加者として意識することができ，共に会話をしているという感覚を持つことができたとしているが，これは，システムが観察者の役に回っているとき，アウェアネス改善の努力をしたことの効果によるものと考えられる．

今後の検討課題として，まずロボットの会話への参加態度の多様化の問題があげられる．

会話自体は、情報提供を目的としたもの(Q/Aのように、一方から一方へ所望の情報を伝える会話)、相互交流を目的としたもの(互いに知識が増えること自体が目的の会話:どのような知識を得るかについては明確な目標を持たなくともよい)、合意形成を目的としたものなどに分類ができる。また、会話への参加態度に関しては、主体的態度(自ら話題提供を積極的に行い、会話の流れを作る)と従属的態度(その場の話題について、流れを変えずに発話を返す)とに分類ができる。これらの分類からは、本研究は、相互交流型の会話に従属的態度で参加するロボットを扱ったといえる。今後、本研究で扱わなかった型の会話に、様々な態度で参加する場合のロボットの戦略について、検討を進める必要がある。

また、ロボットの誤動作の分析とその対処法の確立も重要な課題である。

ロボットの誤動作は、音声認識誤りによる発話内容把握のミスに起因するもの、画像認識誤りによる状況把握のミスに起因するものの2種類に大別される。

会話システム一般にいえることであるが、発話内容の把握に誤りがあった場合は、ロボットは不適切な応答を返し、会話がユーザの意図しない局面へと進むことになる。これを避けるためには、認識結果の信頼度が低い場合、確認の発話を行なうなども考えられる。しかし本ロボットを使った経験からは、若干不正確であっても返答の一候補をユーザに提示した方が、会話を楽しむ観点からは適しているとの印象を持った。むしろ間違いがユーザ間の対話を盛り上げる現象も多く見られた。本ロボットのような相互交流型の会話システムにおいては、誤解を生じないことよりは、後に続く会話によっていかに誤解を解消するかが重要な課題と考える。

状況把握を誤った場合、ロボットはその状況にそぐわない方向に視線を向けるという誤った行動をとるとともに、情報の流れを誤解することになる。誤った方向に視線を向けることは、アウェアネスの主張に失敗するというマイナスの側面を持つと同時に、状況理解を失敗していることをユーザに直接的に伝えるというプラスの側面も持つ。このことによって、ユーザはロボットに誤りを指摘することができ、情報の流れを誤ったまま対話を続けるという致命的な状況は回避することができる。しかしながら、状況把握の誤りは、会話を滞らせる原因になることには間違いはなく、この誤り自体を減じることは重要である。画像処理の精度をあげることがこれらの解決手段となることになるのはもちろんであるが、ロボッ

トの視線の移行速度を向上させることが状況把握の早さと状況判断後の対応の早さにつながるため、誤りの早期修正を含む円滑な対応を実現する上で有効な手段になると考える。今後これらの点についても、より深い検討を進めたい。

## 4.8 まとめ

グループ会話の特徴の分析に基づいて、円滑なグループ会話を実現するために必要なシステムの振舞いについて検討し、さらに、マルチモーダルな情報処理機能を持つ人間型ロボットを用いて、グループ会話に参加できるシステムを実現した。

ロボットは会話の状況に関する判断能力を持つことで、望まれるタイミングで自然な応答をすることができた。また、ロボットが会話の状況の明確化のために行なう身体表現の効果により、会話参加者も戸惑うことなくグループ会話を進めることができた。ロボットには、単に聞かれたことに答えるだけでなく、ユーザ同士の対話に割込んで発話を行なう機能も持たせた。また、ロボットが観察者の役に回っているときでも、アウェアネス改善に向けて対話当事者に働きかける機能を持たせた。これらのことにより、グループとしての会話を活性化することができた。

ここで実現したシステムでは、会話において人がとる振舞いをかなり単純化して捉えており、また扱った対話タスクも小規模なものではある。しかし、グループ会話の本質を捉えたシステム構成になっているものとする。今後このシステムを基礎として、より自然で有用なグループ会話システムを実現していきたい。

## 第5章 総論

本稿においては，マルチモーダル対話システム構築にかかわる諸問題の中で，複数メディアにまたがる迅速で柔軟な情報処理を可能とするマルチプロセッサアーキテクチャの設計と実装，複数の専門を持った開発者によるシステム開発とシステムのインタラクティブな動作を実現する内部システムの設計と実装，それら技術の応用としてのグループ会話に参加する対話ロボットの構築を行った．以下，各章ごとの成果および今後の課題をまとめる．

### 第2章 情報共有アーキテクチャ

多機能ロボットのモジュール開発において必要となる，モジュール間接続アーキテクチャの設計と実装を行なった．モジュール同士のフラットな関係とモジュール自身による情報の流れのダイナミックな切替えを実現する黒板・publish/subscribeの複合サービス，密な統合とスケーラビリティの確保を両立できることを特徴とする2層のバスアーキテクチャを提案し実装した．ロボットに音声・音響・画像情報を統合的に利用した対話タスクを実装し，高密度に統合されたシステムが短期間で開発できることを確認した．

本章における今後の課題のひとつに，シェアードバス型のトポロジの利点である情報の広報性を生かしつつ，その欠点であるアービトレーションの必要性をいかに縮小していくかという問題がある．この問題への対応として，近接グループほど情報の共有の必要度が高いという仮定の下にバスを多層化するアプローチや，クロスバススイッチ型のネットワークを構成しバスの占有権を用途ごとに振り分けるアプローチなどが考えられる．

### 第3章 基本ソフトウェアと開発環境

インタラクティブシステムの共同開発に適した開発環境を設計・実装した．大規

模統合システムの開発においては、複数の分野にまたがる専門知識が必要とされるため、単一の開発者がシステム全体を見通すことは困難となる。そのため、異なる専門を持った複数の開発者たちの共同作業が不可欠となり、それを円滑化させる枠組が必要となる。本章においては、1) 黒板モデルに基づく情報共有によるモジュール開発とスクリプトに基づくモジュール間優先度管理、2) メッセージ配信機能を拡張した黒板型の情報共有機能と CVS に基づく共有情報管理、3) 状況判断機構とスクリプト記述による即時的な行動選択により、機能モジュールによるバザールのシステム開発と、即時的な行動生成を実現することができることを特徴とする階層性をもったシステムアーキテクチャを検討した。そのアーキテクチャを実現する基礎となる基本ソフトウェア、開発者の活動を支援するツールを実現し、音声・音響・画像情報を統合的に利用した対話ロボットの実装を通して、高密度に統合されたシステムが短期間で開発できることを確認した。

本章における今後の課題として、状況判断機構の高機能化、優先度記述スクリプトの記述能力の向上が挙げられる。状況判断機構の高機能化に関して、現在の実装における条件記述とその一致度による状況の判別は、やや確定的であり、複雑な状況の判断において不安定な特性を持つ。近年、複数の事象からの状況の判断を行う確率的な枠組みとしてベイジアンネットワークなどの研究が進展しており、今後それらの導入も検討したい。また、優先度記述スクリプトの記述能力の向上に関して、現時点では、個々の状況につき個々のスクリプトを書く必要がある。状況間の包含関係や目的とする動作の類似度から記述を共有化し、状況の記述のモジュラリティを向上させる改良が望まれる。

#### 第4章 グループ会話に参加する対話ロボットの構築

グループ会話に参加できる対話システムを開発した。グループ会話とは、会話の参加者同士が、対等の関係で行なう多人数会話である。人と機械が一对一で会話することを前提としていた従来の人・機械の対話システムと異なり、グループ会話においては、投げかけられた声が誰によって発せられ誰に向けられたものか、それぞれの会話参加者は誰に注目しているかなど、会話の場に関する状況理解をするとともに、自らも適切な場の形成に努める必要がある。本章では、画像処理、音響処理などを併用することで状況理解を行なうとともに、身体表現によって会

話状況への働きかけを行う機能を実現し、これらを音声認識と組み合わせることで、複数の参加者を相手に会話できるロボットを作成した。

今後の検討として、会話に対する参与の態度、会話グループの形成と崩壊のモデル化の検討などが必要であると考えられる。会話に対する参与の態度は、会話において担う役割に幅を持たせるために必要であり、会話グループの形成と崩壊のモデル化は、グループ会話をより広い空間で把握し、より多人数の対話を扱うために必要である。

本研究はマルチモーダル対話システム構築における問題を、柔軟で効率的な情報処理システムを実現する視点、インタラクティブシステムとしてのダイナミクスを実現する視点、新たなモダリティの導入によりグループ会話を実現する視点から論じた。

グループ会話の実現は、人間同士のコミュニケーションにおいて複数存在するチャンネルのうちの一部のチャンネルを付加したに過ぎないが、対話システムの利用形態に与える変化としては非常に大きなものである。マルチモーダル対話システム研究は、まだ潜在的に大きな可能性を秘めていると考えられ、今後もその可能性を追求していきたい。

本稿では、擬人化された実世界志向の対話システムとしてヒューマノイドロボットを研究の対象とした。ヒューマノイドロボット研究は、実用システムの開発という側面のほかに、人間研究としての側面も持っている。人型のロボットを作ろうという試みは、人の形状を模擬するだけでなく、人の機能も模擬することを必要とする。人型のロボットのコミュニケーション機能の実現を扱った本研究は、人間のコミュニケーション機能の解明に関する研究に他ならない。本研究が人のコミュニケーション能力の理解に向けて少しでも貢献することを願う。

## 参考文献

- [BJ99] R. Bischoff and T. Jain. Natural communication and interaction with humanoid robots. In *Proc. Second International Symposium on Humanoid Robots*, pp. 121–128, 1999.
- [Bro86] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, Vol. 2, , 1986.
- [com] Microsoft corporation: Microsoft com technologies. <http://www.microsoft.com/com/>.
- [cor] Object management group: The omg’s corba website. <http://www.corba.org/>.
- [DB95] P. Dourish and S. Bly. Portholes: Supporting awareness in a distributed work group. In *Proc. ACM CHI’92*, pp. 541–547, 1995.
- [EHRLR80] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy. The hersey-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, Vol. 12, No. 2, pp. 213–253, 1980.
- [EM88] R. Englemore and T. Morgan, editors. *Blackboard Systems*. ADDISON-WESLEY, 1988.
- [Gat97] E. Gat. *Artificial Intelligence and Mobile Robots*, chapter On three-layer architectures. MIT/AAAI Press, 1997.



- [GBG<sup>+</sup>94] D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, S. Seneff, and V. Zue. Galaxy: A human-language interface to on-line travel information. In *Proc. ICSLP'94*, pp. 707–710, 1994.
- [HMNK98] J. Hirasawa, N. Miyazaki, M. Nakano, and T. Kawabata. Implementation of coordinate nodding behavior on spoken dialogue system. In *Proc. ICSLP'98*, Vol. 6, pp. 2347–2350, 1998.
- [jee] 1394 trade association home page. <http://www.1394ta.org/>.
- [IKKI99] H. Ishiguro, T. Kanda, K. Kimoto, and T. Ishida. A roboto architecture based on situated modules. In *IEEE Proc. IROS'99*, pp. 1617–1623, 1999.
- [IMT87] N. Ichiyoshi, T. Miyazaki, and K. Taki. A distributed implementation of flat ghc on the multi-psi. Technical Report TR-230, ICOT, 1987.
- [Ina94] M. Inaba. *Robotics Research*, Vol. 6, chapter Remote-Brained Robotics: Interfacing AI with Real World Behaviors, pp. 335–344. The International Foundation of Robotics Research, 1994.
- [Kat91] Ichiro Kato. Homini-robotism. In *Proc. of IEEE International Conference on Advanced Robotics*, pp. 1–5, 1991.
- [KNK<sup>+</sup>01] S. Kagami, K. Nishiwaki, J. Kuffner, Y. Kuniyoshi, M. Inaba, and H. Inoue. Design and implementation of software research platform for humanoid robotics : H7. In *Proc. IEEE-HUMANOIDS2001*, pp. 253–258, 2001.
- [LC81] V. R. Lesser and D. D. Corkill. Functionally accurate, cooperative distributed systems. *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 11, No. 1, pp. 81–96, 1981.
- [MAHO97] T. Matsui, H. Asoh, I. Hara, and N. Otsu. An event driven architecture for controlling behaviors of the office conversant mobile robot, jijo-2. In *Proc. IEEE-ICRA'97*, pp. 3367–3371, 1997.

- [Meh68] A. Mehrabian. Communication without words. *Psychology Today*, Vol. 2, pp. 52–55, 1968.
- [NN98] H. Nakashima and I. Noda. Dynamic subsumption architecture for programming intelligent agents. In *Proc. Int. Conf. on Multi-Agent Systems*, pp. 190–197, 1998.
- [Nor90] D. A. Norman. 誰のためのデザイン?—認知科学者のデザイン原論. 新潮社, 1990.
- [NT94] K. Nagao and A. Takeuchi. Social interaction: Multimodal conversation with social agents. In *Proc. AAAI’94*, pp. 22–28, 1994.
- [nua] Nuance home. <http://www.nuance.com/>.
- [oaa] Sri international: The open agent architecture. <http://www.ai.sri.com/oaa/>.
- [PWGK99] R. Peters, D. Wilkes, D. Gaines, and K. Kawamura. A software agent based control system for humanrobot interaction. In *Proc. Second International Symposium on Humanoid Robots*, p. Late Paper, 1999.
- [Ray99] E. S. Raymond. *The Cathedral & the Bazaar*. O’Reilly & Associates Inc., 1999.
- [Smi80] R. Smith. The contract net protocol: High-level communication and control in distributed problem solver. *IEEE Transactions on Computers*, Vol. 29, No. 12, pp. 1104–1113, 1980.
- [SSJ74] H. Sacks, E. A. Schegloff, and G. Jefferson. A simplest systematics for the organization of turn taking in conversation. *Language*, Vol. 50, No. 4, pp. 696–735, 1974.
- [Tan95] A. S. Tanenbaum. *Distributed Operating Systems*. Prentice Hall, Inc., 1995.

- [TMIK00] T. Tojo, Y. Matsusaka, T. Ishii, and T. Kobayashi. A conversational robot utilizing facial and body expressions. In *IEEE Proc. SMC2000*, pp. 858–863, 2000.
- [tos] 東芝音声システム. [http://www2.toshiba.co.jp/pc/service/download/mimi/index\\_j.htm](http://www2.toshiba.co.jp/pc/service/download/mimi/index_j.htm).
- [Ver98] R. Vertegaal. *Look who's talking to whom? Mediating Joint Attention in Multiparty Communication and Collaboration*. Cognitive Ergonomics Department University of Twente, 1998.
- [vid] Robita オンラインビデオライブラリ. <http://www.tk.elec.waseda.ac.jp/robita/>.
- [VVvdV97] R. Vertegaal, B. Velichkovsky, and G. van der Veer. Catching the eye: Management of joint attention in cooperative work. *SIGCHI*, Vol. 29, No. 4, 1997.
- [War96] N. Ward. Using prosodic clues to decide when to produce back-channel utterances. In *Proc. ICSLP'96*, pp. 1728–1731, 1996.
- [WST94] K. Watanuki, K. Sakamoto, and F. Togawa. Analysis of multimodal interaction data in human communication. In *Proc. ICSLP'94*, pp. 899–902, 1994.
- [YKK<sup>+</sup>01] K. Yokoi, F. Kanehiro, K. Kaneko, K. Fujiwara, S. Kajita, and H. Hirukawa. A honda humanoid robot controlled by aist software. In *Proc. IEEE-HUMANOIDS2001*, pp. 259–264, 2001.
- [YTM<sup>+</sup>91] Y. Yokote, F. Teraoka, A. Mitsuzawa, N. Fujinami, and M. Tokoro. The muse object architecture: A new operating system structuring concept. *ACM Operating Systems Review*, Vol. 25, No. 2, pp. 22–45, 1991.
- [伊藤 01] 伊藤孝行, 新谷虎松. マルチエージェントシステムのための実装技術とその応用. 人工知能学会誌, Vol. 16, No. 4, pp. 469–475, 2001.

- [於久 01] 於久健太郎, 松坂要佐, 小林哲則. ロボット用情報共有型システムバスと組み込み cpu 一体型小型モータコントローラの開発. 第 19 回 ロボット学会学術講演会論文集, pp. 1271–1272, 2001.
- [横山 99] 横山真男, 青山一美, 菊池英明, 帆足啓一郎, 白井克彦. 人間型ロボットの対話インタフェースにおける発話交替時の非言語情報の制御. 情処学論, Vol. 40, No. 2, pp. 487–496, 1999.
- [久保 00] 久保田千太郎, 松坂要佐, 小林哲則. グループ会話ロボットにおける顔画像処理システム. 信学技法, pp. pp.49–56, 2000.
- [橋本 97] 橋本周司, 成田誠之助, 白井克彦, 小林哲則, 高西淳夫, 菅野重樹, 笠原博徳. ヒューマノイド –人間型高度情報処理ロボット–. 情報処理, Vol. 38, No. 11, pp. 959–969, 1997.
- [古川 99] 古川賢司, 東條剛史, 早田啓介, 藤江真也, 松坂要佐, 小林哲則. 空間情報共有のための身体表現機能を有する対話ロボット. 第 17 回 日本ロボット学会学術講演会予稿集, pp. 1181–1182, 1999.
- [山崎 96] 山崎信行, 安西祐一郎. パーソナルロボット用機能別並列計算機アーキテクチャ : Aspire. 情報処理学会論文誌, Vol. 37, No. 1, pp. 81–91, 1996.
- [山崎 01] 山崎信行, 松井俊浩. 並列分散リアルタイム制御用レスポンスプロセッサ. 日本ロボット学会誌, Vol. 19, No. 3, pp. 68–77, 2001.
- [松井 00] 松井俊浩, 麻生英樹, John Fry, 浅野太, 本村陽一, 原功, 栗田多喜夫, 速水悟, 山崎信行. オフィス移動ロボット jijo-2 の音声対話システム. 日本ロボット学会誌, Vol. 18, No. 2, pp. 300–307, 2000.
- [松坂 01] 松坂要佐, 東條剛史, 小林哲則. グループ会話に参加する対話ロボットの構築. 信学論, Vol. J84-D-II, No. 6, pp. 898–908, 2001.
- [城塚 98] 城塚音也, 桑田喜隆, 安地亮一, 小泉宣夫. 遠隔会議を対象とした音声対話モニタリングによる対話支援システム. 情処学論, Vol. 39, No. 5, pp. 1240–1247, 1998.

- [星野 01] 星野由紀子, 福島崇文, 金広文男, 稲葉雅幸, 井上博允. 全身行動ロボットにおける視聴触覚を通じた対話による行動獲得システム. 第 6 回ロボティクスシンポジア, pp. 205–210, 2001.
- [西本 96] 西本卓也, 志田修利, 小林哲則, 白井克彦. マルチモーダル入力環境下における音声の協調的利用—音声作図システム s-tgif の設計と評価—. 信学論, Vol. J79-D-II, No. 12, pp. 2176–2183, 1996.
- [石塚 00] 石塚満. マルチモーダル擬人化エージェントシステム. システム/制御/情報, Vol. 44, No. 3, pp. 128–135, 2000.
- [村井 00] 村井則之, 小林哲則. 話者性と発話交代を考慮した複数話者対話音声の認識. 信学論, Vol. J83-D-II, No. 11, pp. 2465–2472, 2000.
- [中西 98] 中西英之, 吉田力, 西村俊和, 石田享. Freewalk: 3 次元仮想空間を用いた非形式的コミュニケーションの支援. 情処学論, Vol. 39, No. 5, pp. 1356–1364, 1998.
- [中野 99] 中野鐵兵, 山田泰資, 松坂要佐, 東條剛史, 古川賢司, 久保田千太郎, 小林哲則. 自律ロボットの利用によるインターネットの実空間への拡張. 情報処理学会 INTERACTION'99, pp. 167–168, 1999.
- [田宮 99] 田宮大介, 松坂要佐, 小林哲則. ロボット頭部に設置した 2 系統のマイクによる音源定位. 日本音響学会 春季研究発表会 講演論文誌, pp. 469–470, 1999.
- [白井 85] 白井克彦, 小林哲則, 岩田和則, 深沢克夫. ロボットの柔軟な対話を目的とした音声入出力システム—wabot-2 における会話系. 日本ロボット学会誌, Vol. 3, No. 4, pp. 104–113, 1985.
- [本村 99] 本村陽一, 原功, 田中久美子. 学習知能ロボットにおける状況依存エージェントの協調. 日本ソフトウェア科学会マルチエージェントと協調計算ワークショップ (MACC'99), 1999.

- [本田 97] 本田新九郎, 富岡展也, 木村尚亮, 岡田謙一, 松下温. 在宅勤務者の疎外感の解消を実現した位置アウェアネス・アウェアネススペースに基づく仮想オフィス環境. 情処学論, Vol. 38, No. 7, pp. 1454–1464, 1997.

# 関連する研究業績

## 論文

- (1) 松坂要佐, 於久健太郎, 小林哲則, 多機能ロボット開発のための情報共有アーキテクチャの設計と実装, 電子情報通信学会論文誌 D-I, (掲載決定).
- (2) Yosuke Matsusaka, Tsuyoshi Tojo, Tetsunori Kobayashi, Conversation Robot Participating in Group Conversation, IEICE Trans. on Information and Systems Vol. E86-D, No. 1, pp. 26-36, (2003.1).
- (3) 松坂要佐, 東條剛史, 小林哲則, グループ会話に参与する対話ロボットの構築, 電子情報通信学会論文誌 Vol. J84-D-II, No. 6, pp. 898-908, (2001.6).
- (4) KyeongJu Kim, Yosuke Matsusaka, Tetsunori Kobayashi, Inter-Module Cooperation Architecture for Interactive Robot, Proc. IROS2002, Vol. 3, pp. 2286-2291, (2002.10).
- (5) Yosuke Matsusaka, Tetsunori Kobayashi, System Architecture to Realize Widely Applicable and Interactive Behavior of the Robot, Proc. PRICAI-02 WS-3 International Workshop on Lifelike Animated Agents, pp. 77-82, (2002.8).
- (6) Yosuke Matsusaka, Tetsunori Kobayashi, System Software for Collaborative Development of Interactive Robot, Proc. IEEE-HUMANOIDS2001, pp. 271-277, (2001.11).
- (7) Yosuke Matsusaka, Shinya Fujie, Tetsunori Kobayashi, Modeling of Conversational Strategy for the Robot Participating in the Group Conversation, Proc. ISCA-EUROSPEECH2001, pp. 2173-2176, (2001.9).



## 国際会議

- (8) Tsuyoshi Tojo, Yosuke Matsusaka, Tomotada Ishii, Tetsunori Kobayashi, A Conversational Robot Utilizing Facial and Body Expressions, IEEE Proc. International Conf. on System, Man and Cybernetics SMC2000, pp. 858-863, (2000.10).
- (9) Y.Matsusaka, T.Kobayashi, Human Interface of Humanoid Robot Realizing Group Communication in Real Space, Proceedings of the Second International Symposium on Humanoid Robots, pp. 188-193, (1999.10).
- (10) Y.Matsusaka, T.Tojo, S.Kubota, K.Furukawa, D.Tamiya, K.Hayata, Y.Nakano, T.Kobayashi, Multi-person Conversation via Multi-modal Interface -A Robot who Communicate with Multi-user-, Proc. Eurospeech 99, Vol. 4, pp. 1723-1726, (1999.9).
- (11) Y.Matsusaka, S.Kubota, T.Tojo, K.Furukawa, T.Kobayashi, Multi-person Conversation Robot using Multi-modal Interface, Proc. SCI/ISAS '99, Vol. 7, pp. 450-455, (1999.7).

## 講演発表

- (12) 松坂要佐, 江尻康, 小林哲則, 合意形成型対話における声・顔の機能的表情の分析と認識, 電子情報通信学会技術研究報告 PRMU2002-118, pp. 31-36, (2002.11).
- (13) 於久健太郎, 清水健二, 松坂要佐, 小林哲則, マルチモーダルロボット用マルチプロセッサアーキテクチャ, 情報処理学会研究報告 2002-ARC-150-10, pp. 47-52, (2002.11).
- (14) 松坂要佐, 東條剛史, 小林哲則, 論文賞記念講演:グループ会話に参加する対話ロボットの構築, 電子情報通信学会技術研究報告 SP2002-101, pp. 39-40, (2002.10).

- (15) 江尻康, 松坂要佐, 小林哲則, 対話中における頭部ジェスチャの認識, 電子情報通信学会技術研究報告 PRMU2002-61, Vol. 102, No. 218, pp. 31-36, (2002.7).
- (16) 松坂要佐, 小林哲則, チュートリアル: 対面对話の収録と解析のための信号処理とパターン認識, 人工知能学会研究会資料 SIG-SLUD-A201, pp. 27-32, (2002.6).
- (17) 金景柱, 松坂要佐, 小林哲則, インタラクティブロボットのための複数機能モジュールの協調処理アーキテクチャ, 第7回ロボティクス・シンポジウム予稿集, pp. 215-220, (2002.3).
- (18) 松坂要佐, 小林哲則, 異分野エキスパート集団によるインタラクティブロボットの共同開発のための基本ソフトウェア, 情報処理学会研究報告 2001-ICS-126, pp. 43-48, (2001.10).
- (19) 於久健太郎, 松坂要佐, 小林哲則, ロボット用情報共有型システムバスと組み込み CPU 一体型小型モータコントローラの開発, 第19回ロボット学会学術講演会論文集, pp. 1271-1272, (2001.9).
- (20) 内田亮, 松坂要佐, 小林哲則, 自律移動ロボットにおける環境認識及び行動計画の多階層化の評価, 第19回ロボット学会学術講演会論文集, pp. 1009-1010, (2001.9).
- (21) 松坂要佐, 小林哲則, ROBITA: グループ会話ロボット, 人工知能学会研究会資料 SIG-Challenge-0113, pp. 1-8, (2001.6).
- (22) 松坂要佐, 小田勇一郎, 小林哲則, グループ会話におけるマルチモーダル会話データの収録・解析システムの構築, 第15回人工知能学会全国大会 3B2-02, (2001.5).
- (23) 松坂要佐, 小林哲則, 研究会セッション (SLUD): 実世界におけるコミュニケーション研究の現在 -ロボットとのコミュニケーション-, 第15回人工知能学会全国大会 1D3-04, (2001.5).

- (24) 東條剛史, 松坂要佐, 小林哲則, 対話ロボットの機能的身体表現に適した手指部の開発, 第6回ロボティクス・シンポジウム予稿集, pp. 280-285, (2001.3).
- (25) 松坂要佐, 於久健太郎, 小林哲則, マルチモーダル対話ロボットのための情報共有・メッセージ通知システム, SICE SI2000 (システムインテグレーション部門学術講演会), pp. 191-192, (2000.12).
- (26) 久保田千太郎, 松坂要佐, 小林哲則, グループ会話ロボットにおける顔画像処理システム, 電子情報通信学会技術報告 NLC99-85, PRMU99-268, pp. 49-56, (2000.3).
- (27) 東條剛史, 松坂要佐, 小林哲則, 身振りによる対話調整機能を持つ対話ロボット, 情報処理学会 研究報告 SLP-30-6, pp. 33, (2000.2).
- (28) 松坂要佐, 東條剛史, 古川賢司, 藤江真也, 小林哲則, ジェスチャの表現・理解機能を有するロボットによる空間情報共有型対話の実現, 音響学会 秋期研究発表会 講演論文集, Vol. 1, pp. 111-112, (1999.9).
- (29) 古川賢司, 東條剛史, 早田啓介, 藤江真也, 松坂要佐, 小林哲則, 空間情報共有のための身体表現機能を有する対話ロボット, 第17回ロボット学会学術講演会予稿集, pp. 1181-1182, (1999.9).
- (30) 田宮大介, 松坂要佐, 小林哲則, ロボット頭部に設置した2系統のマイクロフォンによる音源の定位, 音響学会春季研究発表会 講演論文集, Vol. 1, pp. 469-467, (1999.3).
- (31) 松坂要佐, 中野裕一郎, 早田啓介, 久保田千太郎, 田宮大介, 東條剛史, 小林哲則, 画像・音声・音響情報を統合した複数話者との会話ロボット, 音響学会春季研究発表会 講演論文集, Vol. 1, pp. 95-96, (1999.3).
- (32) 松坂要佐, 東條剛史, 久保田千太郎, 田宮大介, 古川賢司, 早田啓介, 中野裕一郎, 小林哲則, 複数話者による対話システム, インタラクシオン'99 論文集, 情報処理学会シンポジウムシリーズ 99-4, pp. 33-34, (1999.3).
- (33) 中野鐵兵, 山田泰資, 松坂要佐, 東條剛史, 古川賢司, 久保田千太郎, 小林哲則, 自律ロボットの利用によるインターネットの実空間への拡張, インタラ

クシヨン'99 論文集, 情報処理学会シンポジウムシリーズ 99-4, pp. 167-168, (1999.3).

- (34) 松坂要佐, 東條剛史, 古川賢司, 久保田千太郎, 藤江真也, 早田啓介, 川崎明彦, 益満健, 中野裕一郎, 小林哲則, アイコンタクト機能を持つ複数話者との対話ロボット, ロボット学会学術講演会, pp. 1219-1220, (1998.9).

# 謝辞

本研究の遂行にあたり貴重な御指導と御助言を賜った早稲田大学教授・小林哲則先生に感謝する。

システム構築にあたり、学科の枠を超えて、御教示、ご理解をいただいたヒューマノイドプロジェクトの先生方、および研究室の方々に感謝する。特に、ロボット開発初期のハードウェアの設計・開発において、元となる設計図を提供いただき、数多くの御助言・御指導をいただいた、高西敦夫教授、山口仁一助手に感謝する。また、研究を深く理解し、デモ展示等の多くの発表の機会を与えていただいた、橋本周司教授、ピトヨ ハルトノ助手に感謝する。プロジェクト事務局の太田久子氏には、事務面にとどまらない多くの助力をいただいた。

本研究は、システム構築にかかわった、早稲田大学知覚情報システム研究室の学生の方々の努力なしには、成り立たなかったものである。彼らの熱意に深く感謝したい。