

TR-I-0303

ATR 音声翻訳システム ASURA の実装  
Implementation of ATR Speech Translation System: ASURA

竹沢 寿幸    田代 敏久    菊井 玄一郎    鈴木 雅実    鷹見 淳一  
Toshiyuki Takezawa    Toshihisa Tashiro    Gen-ichiro Kikui    Masami Suzuki    Junichi Takami  
永井 明人    ハラルド・シンガー    三村 克彦    谷戸 文廣    高嶋 浩司†  
Akito Nagai    Harald Singer    Katsuhiko Mimura    Fumihiro Yato    Kouji Takashima†  
伴 敏雄†    石田 聡†    田川 博章†    栗原 一彦†    道尻 佳憲†  
Toshio Ban†    Satoshi Ishida†    Hiroaki Tagawa†    Kazuhiko Kurihara†    Yoshinori Michijiri†

1993. 3. 3

内容梗概

ATR 音声翻訳システム ASURA を構築した。本報告書はそのプログラムの実装方法および操作方法に関する解説書である。ASURA にはいろいろなバージョンが存在するので、自動翻訳電話国際共同実験のために特別に開発されたバージョンを中心に報告する。これまで謎に包まれていた音声認識、言語翻訳のプログラム構成とその接続方法をわかりやすく述べる。さらに、自動翻訳電話国際共同実験のために開発された通信機能と、日本語音声合成システム ATR *u*-Talk の接続方法についても詳しく記す。ASURA システムの動作試験などのために開発したローカルテスト用ツールについても記す。

ATR 自動翻訳電話研究所  
ATR Interpreting Telephony Research Laboratories  
†エスイーティ株式会社  
‡株式会社 東洋情報システム

© 株式会社 エイ・ティ・アール自動翻訳電話研究所  
© 1993 by ATR Interpreting Telephony Research Laboratories

# 目次

1	まえがき	1
2	自動翻訳電話国際共同実験用 ASURA の概要	3
2.1	動作環境	3
2.2	操作方法	3
2.2.1	ASURA の基本操作方法	4
2.2.2	音声入力方法	12
2.2.3	文節音声認識窓とその操作方法	13
2.2.4	音声認識結果の文候補表示窓とそこでの文候補選択方法	14
2.2.5	翻訳結果表示窓とその操作方法	14
2.3	話者適応	15
2.4	ローカルテスト用ツールの基本操作方法	19
3	自動翻訳電話国際共同実験用 ASURA の実装	23
3.1	動作環境	23
3.2	プログラムの構成	26
3.2.1	プロセス構成について	26
3.2.2	認識系システム構成	26
3.2.3	翻訳系システム構成	29
3.2.4	通信制御系システム構成	30
3.3	統括制御	33
3.3.1	概要	33
3.3.2	ディレクトリ構成	33
3.3.3	ディレクトリ説明	33
3.3.4	主なシェルスクリプト	34
3.3.5	翻訳コントロールプロセス	35
3.3.6	プロセス間のメッセージ書式	40
3.4	音声認識	42
3.4.1	音声入力窓	42
3.4.2	文節音声認識窓	42
3.4.3	音声認識結果の文候補表示選択窓	46
3.5	言語翻訳	50
3.5.1	翻訳部各プロセスの概要	50
3.5.2	翻訳部各プロセスの実行ファイルの所在地	51

3.5.3	翻訳部各プロセスのプロセス間通信	51
3.5.4	データの流れ	52
3.5.5	ドイツ語のキャラクターセットについて	53
3.5.6	翻訳結果表示窓	55
3.6	通信制御	58
3.6.1	公衆電話回線モデム接続	58
3.6.2	パケット回線 (DDX-P) 接続	59
3.6.3	通信メインコントロールプロセス	60
3.6.4	ソケットコントロールプロセス	61
3.7	日本語音声合成	62
3.7.1	ソケット通信ルーチン	63
3.7.2	文字列チェックルーチン	63
3.7.3	音声合成ルーチン	63
3.7.4	合成音声出力ルーチン	64
3.8	相手側の状態表示窓	65
3.8.1	画面構成の説明	65
3.8.2	機能説明	65
3.8.3	通信について	66
3.8.4	CMU/Siemens から受け取る日本語文字列の書式の違い	67
3.8.5	処理説明	68
3.9	オプション操作画面	71
3.9.1	画面構成の説明	71
3.9.2	機能説明	73
3.9.3	通信について	75
3.9.4	処理説明	77
3.10	システムパラメータの初期設定と変更	83
3.11	話者適応	86
3.12	ローカルテスト用ツール	87
3.12.1	プログラムの概要	87
3.12.2	プログラムで有効なオプション指定	87
3.12.3	メインウインドウ	88
3.12.4	オプションパネルウインドウ	89
3.12.5	シナリオセレクションウインドウ	90
3.12.6	シナリオウインドウ	91
3.12.7	メッセージウインドウ	92
4	自動翻訳電話国際共同実験用 ASURA で必要なファイル等	93
4.1	音声認識用辞書ファイル	93
4.1.1	文法ファイルの説明	93
4.1.2	文法の変換方法	94
4.2	日本語構文解析部の文法・辞書	96
4.2.1	文法・辞書のバージョン管理	96
4.2.2	文法・辞書のロード方法	96

4.3	変換用規則ファイル	99
4.3.1	変換規則のバージョン	99
4.3.2	変換規則のロード方法	99
4.3.3	変換規則を更新する場合の注意点	99
4.4	生成用文法・辞書ファイル	101
4.4.1	生成用文法・辞書ファイルの概要	101
4.4.2	文法・辞書ファイルのロードの仕方	102
4.4.3	初期設定ファイル	102
4.4.4	文法・辞書ファイルを更新する場合の注意点	104
4.5	日本語音声合成システムに必要なデータ／ファイルフォーマット	106
4.5.1	CMU 側入力データ	106
4.5.2	Siemens 側入力データ	106
4.5.3	単語情報ファイル (*.word)	106
4.5.4	アクセント句情報ファイル (*.phra)	106
4.5.5	マッチング用リストファイル (CHK_list)	106
4.5.6	辞書ファイル	107
4.5.7	CMU 側入力と Siemens 側入力の違い	107
4.6	その他、特記事項 (トラブルシューティングなど)	108
5	自動翻訳電話国際共同実験のための対話コーパス	109
5.1	CSTAR プロジェクト開始までの経緯	109
5.2	CSTAR プロジェクト発足後の経緯	110
5.3	デモ用コーパスの決定	110
6	自動翻訳電話国際共同実験におけるテレビ会議システム等	113
6.1	自動翻訳電話国際共同実験におけるテレビ会議システム	113
6.2	自動翻訳電話国際共同実験における音響系等	115
7	daily demo 用 ASURA の概要	117
7.1	動作環境	117
7.2	操作方法	117
7.2.1	起動方法	117
7.2.2	基本的な操作方法および音声入力方法	117
7.2.3	終了方法	117
7.3	話者適応	118
7.4	システムパラメータの初期設定と変更	118
8	むすび	119
	謝辞	120
	参考文献	121

A	国際共同実験用コーパス	123
A.1	2JE (参加料の問合せ)	124
A.2	6JE (市内観光)	125
A.3	3GJ (会議のテーマについて)	126
A.4	6GJ (市内観光)	127
B	CSTAR 用 会話文 合成音声リスト	129
C	CSTAR 用 通信規約 (プロトコル)	131
C.1	メッセージ書式	131
C.2	メッセージ種別	131
C.3	メッセージ交換例	132
C.4	通信規則	133
C.5	タイミングチャート	134

## 目次

2.1	SpeechPress という音声入力窓の動作例。2 文節を発話した状態で、陰影の付いた部分が切り出された音声区間を表している。陰影の付いた部分は X 画面上では淡い青色。 . . . . .	12
2.2	SpeechPress における開始およびヘルプ画面 . . . . .	13
2.3	音声認識結果の文候補表示選択ウインドウ . . . . .	14
2.4	翻訳結果表示ウインドウ . . . . .	14
2.5	話者選択画面 . . . . .	15
2.6	話者名入力画面 . . . . .	16
2.7	システムメニュー画面 . . . . .	16
2.8	話者適応メニュー画面 . . . . .	17
2.9	話者適応入力画面 . . . . .	18
2.10	ローカルテスト用ツールのメインウインドウ . . . . .	21
2.11	ローカルテスト用ツールのオプションパネルウインドウ . . . . .	21
2.12	ローカルテスト用ツールのシナリオセクションウインドウ . . . . .	22
2.13	ローカルテスト用ツールのシナリオウインドウ . . . . .	22
2.14	ローカルテスト用ツールのメッセージウインドウ . . . . .	22
3.1	マシン環境図 . . . . .	25
3.2	認識系プロセス構成図 . . . . .	26
3.3	翻訳系プロセス構成図 . . . . .	29
3.4	通信制御系プロセス構成図 (対アメリカ) . . . . .	30
3.5	通信制御系プロセス構成図 (対ドイツ) . . . . .	31
3.6	システムの同期の取り方 . . . . .	38
3.7	音声認識、言語解析、変換生成の間のやりとり . . . . .	39
3.8	文節音声認識窓プログラム構成図 . . . . .	43
3.9	音声認識結果の文候補表示選択窓プロセスのしくみ . . . . .	47
3.10	翻訳部各プロセス . . . . .	50
3.11	翻訳部各プロセスのプロセス間通信 . . . . .	52
3.12	翻訳部におけるデータの流れ . . . . .	53
3.13	翻訳処理プロセスと文字コードの関係 . . . . .	54
3.14	翻訳結果表示窓プロセスのしくみ . . . . .	55
4.1	Siemens システムにおける日本語生成用辞書 (lexicon) フォーマット . . . . .	107
6.1	テレビ会議システムのネットワーク構成 . . . . .	114

6.2	ATR サイトにおけるテレビ会議システムの機器構成 . . . . .	115
6.3	ミキサーの接続構成 . . . . .	116
6.4	大会議室のビデオプロジェクタとの接続 . . . . .	116

## 表目次

3.1	ドイツ語非標準アスキー文字のコード	54
3.2	日本語音声合成モジュールの構成	62
4.1	日本語音声合成のための辞書ファイル例	107



# 第1章

## まえがき

異なる言語を話す人の間のコミュニケーションが円滑に行なえることを目指して、音声認識・言語翻訳および音声合成を一貫して結合した音声翻訳システム ASURA (Advanced Speech Understanding and Rendering System of ATR) を構築した [1, 2, 3]。本報告書では、そのプログラムの実装方法および操作方法について述べる。ASURA にはいろいろなバージョンが存在するが、ここでは、自動翻訳電話国際共同実験のために特別に開発されたバージョンを中心に報告する。自動翻訳電話国際共同実験のために開発された通信機能と、日本語音声合成システム ATR *u*-Talk [4, 5] の接続方法についても詳しく記す。

第2章は自動翻訳電話国際共同実験用 ASURA のユーザ用マニュアルを目指している。概要を知りたい人や操作のみする必要がある人はここだけを読めばよい。第3章は同じもののプログラマ用マニュアルを目指している。実際のプログラムの実装方法を知りたい人やプログラムに改良を加える必要がある人はさらにここも読むとよい。第4章は同じものの管理者・保守担当者用の手引である。文法ファイルの差し替えをするような場合はプログラムの内容を詳しく知る必要はなく、第2章と第4章を読めばその作業ができるはずである。第5章では自動翻訳電話国際共同実験のための対話コーパスについて説明する。第6章では自動翻訳電話国際共同実験で使ったテレビ会議システムおよび音響系等について述べる。第7章ではプロトタイプ室で日々の来客のためのデモを行なうための (daily demo 用) ASURA に関する簡単な説明を与えている。

ちなみに、各著者が担当した範囲は次の通りであるが、各著者が執筆した原稿に竹沢が変更を加えていることもある。用語や表現もなるべく統一するよう努めているが、必ずしも十分に調整されてはいない。

名前	担当	主に執筆したところ
竹沢 寿幸	全体の編集と調整	第1章, 第7章, 第8章, その他
田代 敏久	言語翻訳	3.5.1節~3.5.4節, 4.2節
菊井 玄一郎	言語翻訳	3.5.5節, 4.4節
鈴木 雅実	言語翻訳, コーパス	4.3節, 第5章
鷹見 淳一	話者適応	2.3節, 3.11節
永井 明人	音声認識	4.1節
ヘラルド・シンガー	音声入力	2.2.2節, 3.4.1節 (日本語への翻訳は竹沢)
三村 克彦	日本語音声合成, 音響系	3.7節, 4.5節, 6.2節
谷戸 文廣	テレビ会議システム	6.1節
高嶋 浩司	統括制御など	3.1節, 3.2節, 3.3節, 3.10節, 4.6節
伴 敏雄	プログラム	2.2.3節, 3.3.4節, 3.4.2節
石田 聡	通信制御など	2.2.1節, 3.6節
田川 博章	画面など	3.8節, 3.9節
栗原 一彦	ローカルテスト用ツールなど	2.2.4節, 2.2.5節, 2.4節, 3.4.3節, 3.5.6節, 3.12節
道尻 佳憲	日本語音声合成	3.7節, 4.5節



## 第 2 章

### 自動翻訳電話国際共同実験用 ASURA の概要

第 2 章は自動翻訳電話国際共同実験用 ASURA のユーザ用マニュアルを目指している。概要を知りたい人や操作のみする必要がある人はここだけを読めばよい。

#### 2.1 動作環境

基本的には、ワークステーション上で動作する。音声認識処理に HP9000/750 を 1 台 (atrp11)、言語翻訳処理に HP9000/750 を 1 台 (atrp19)、通信制御に SUN SPARCstation2 を 1 台 (as22)、音声合成のために HP9000/730 を 1 台 (atrp17) 利用した。詳細は、3.1 節に記す。

#### 2.2 操作方法

## 2.2.1 ASURAの基本操作方法

ASURAの起動方法、起動後の会話方法、終了までのプロセスを説明します。

- ・ ASURAを起動する。
- ・ 回線を接続する。
- ・ 相手と会話する。
- ・ 回線を切断する。
- ・ ASURAを終了する。

### ASURAを起動する

ここでは、ATR内でのASURAの起動、CMU、SIEMENSとのASURAの起動方法について説明します。

1. as22 と atrp11 に cstar でログインする。
2. モデムの電源を入れる。(DDX-Pの場合不要)
3. as22 で "cd ~cstar/DEMO\_R1.0/COMMU/EXE" と入力してディレクトリ移動。
4. as22 で "ASURA" を起動する。
  - ・ ATR内での時 "ASURA" を起動する。翻訳システムは英語版
  - ・ CMUと接続する時 "ASURA\_JANUS.csh" を起動する。
  - ・ SIEMENSと接続する時
    - モデム経由の時 "ASURA\_SIE\_ANAL.csh" 又は "ASURA\_SIE\_ISDN.csh" を起動する。(この2つの違いは電話番号だけです。)
    - DDX-P経由の時 "ASURA\_SIE\_x25.csh" を起動する。



5. atrp11 で話者選択ウィンドウが表示されるので、話者を選択する。  
(図2.2.1-1)

6. as22 に下記ウィンドウが表示されるのを待つ。

メッセージウィンドウ (図2.2.1-2)  
オプションパネルボタン (図2.2.1-3)

図2.2.1-1 話者選択ウィンドウ

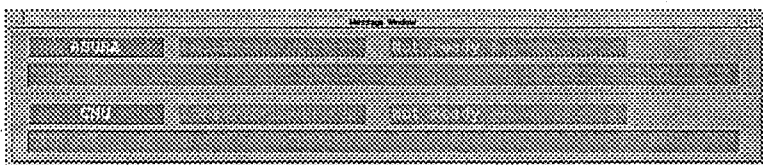


図2.2.1-2 メッセージウィンドウ

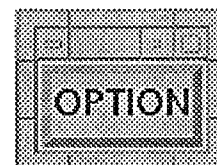


図2.2.1-3 オプションパネルボタン



図2.2.1-4 認識文節表示ウインドウ



図2.2.1-5 認識文表示ウインドウ



図2.2.1-6 翻訳結果表示ウインドウ

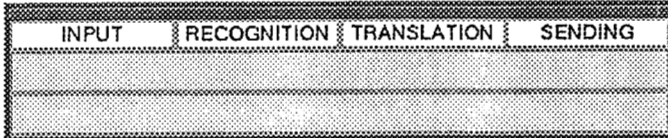


図2.2.1-7 相手側状態表示ウインドウ

7. atp11 の下記ウインドウが表示されるのを待つ。

認識文節表示ウインドウ (図2.2.1-4)

認識文表示ウインドウ (図2.2.1-5)

翻訳結果表示ウインドウ (図2.2.1-6)

相手側状態表示ウインドウ (図2.2.1-7)

8. as22 のメッセージウインドウ (図2.2.1-8) に "Ready for Connect" が表示されれば起動完了。

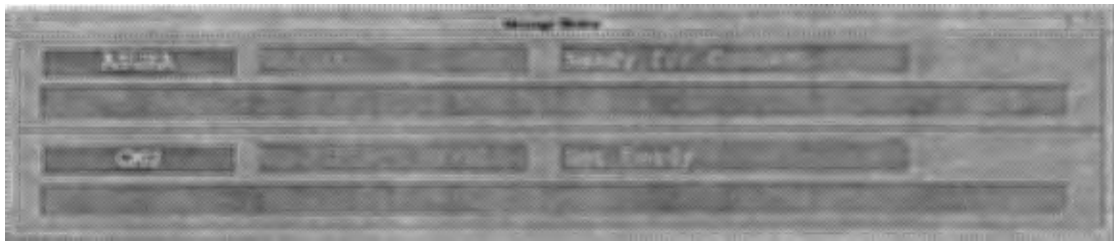


図2.2.1-8 メッセージウインドウ (起動完了)

## 回線を接続する

ここでは、モデム、又はDDX-Pを経由した接続方法について説明します。

1. as22 の右上に表示されているオプションパネルボタン (図2.2.1-3) をマウス左ボタンでクリックしてオプションパネルを開く。(図2.2.1-9、図2.2.1-10)

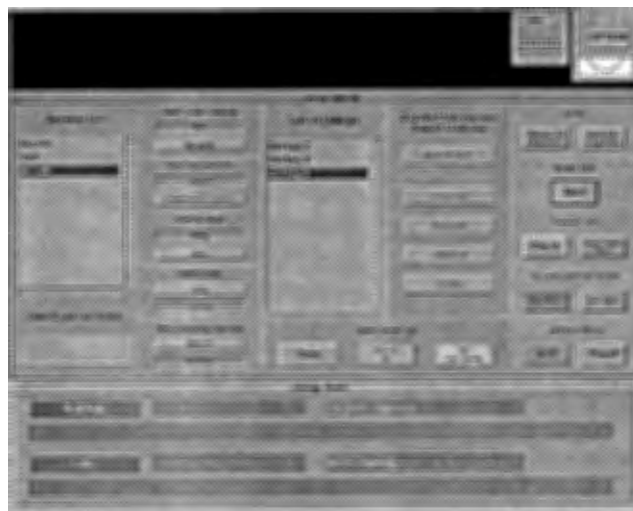


図2.2.1-9 as22の画面

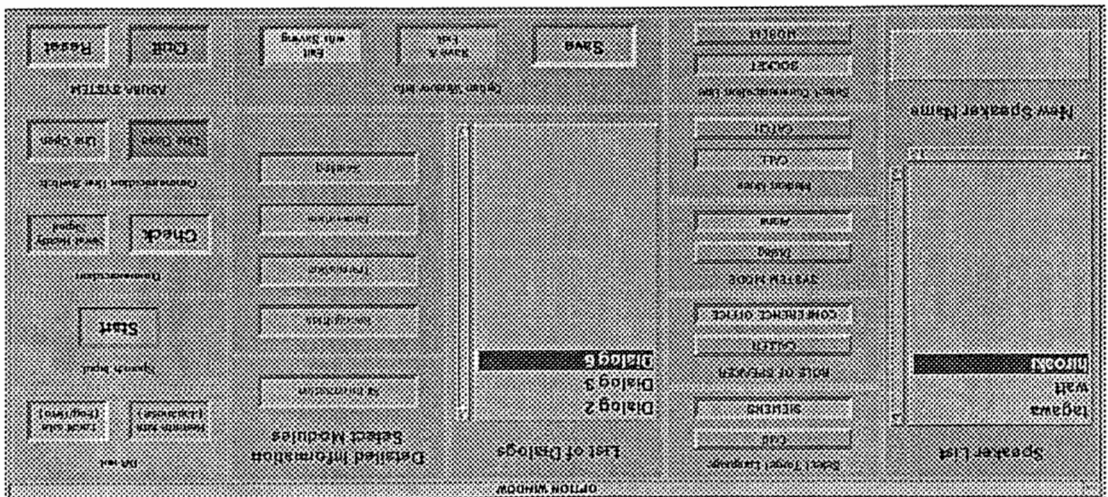


図2.2.1-10 オプションパネル

2. 下記項目を設定し(図2.2.1-10-A、図2.2.1-10-B)、[Save]ボタン(図2.2.1-10-C)をクリックする。  
左ボタンでクリックする。

Select Target Language  
 ROLE OF SPEAKER  
 SYSTEM MODE  
 Modem Mode  
 Select Communication Line  
 List of Dialogs

CMU -- SIEMENS の設定  
 CALLER -- CONFERENCE OFFICE の設定  
 Dialog -- Alone の設定  
 CALL -- CATCH の設定  
 SOCKET -- MODEM の設定  
 シナリオの設定

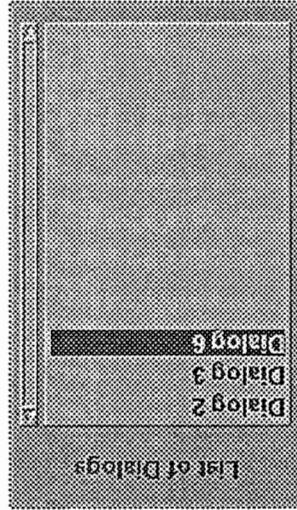


図2.2.1-10-B ダイアログ選択

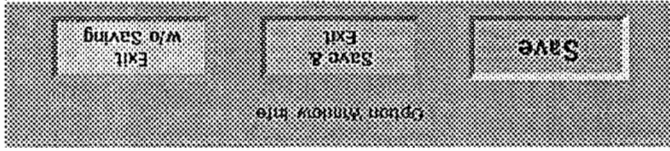
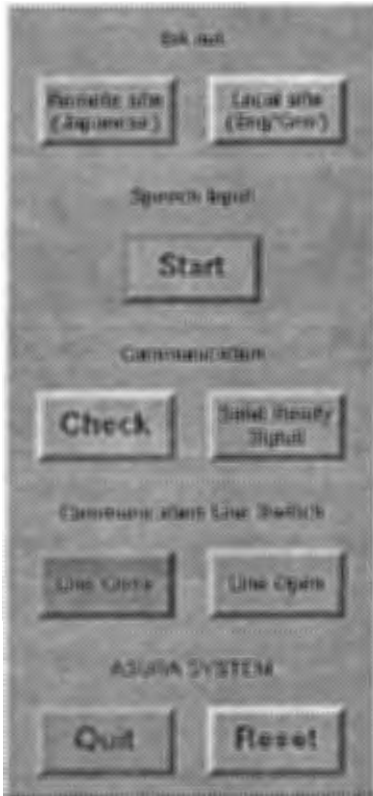


図2.2.1-10-C クリックボタン

図2.2.1-10-A



3. 接続相手の準備ができたなら [Line open] (図2.2.1-10-D) をマウス左ボタンでクリックする。

4. as22 のメッセージウインドウ (図2.2.1-8) で、相手が Ready になったら [Send Ready Signal] ボタン (図2.2.1-10-D) をマウス左ボタンでクリックする。

5. メッセージウインドウに (図2.2.1-11) "Input Ready" が表示されれば準備完了。

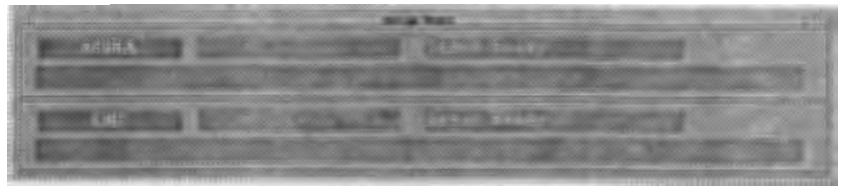


図2.2.1-10-D コマンドボタン  
相手と会話する

図2.2.1-11 メッセージウインドウ (準備完了)

1. 最初はこちらから話す場合 [Start] ボタン (図2.2.1-10-D) をマウス左ボタンでクリックする。
2. atp11 に音声入力ウインドウ (図2.2.1-12) と発話サイトウインドウ (図2.2.1-13) が表示されます。

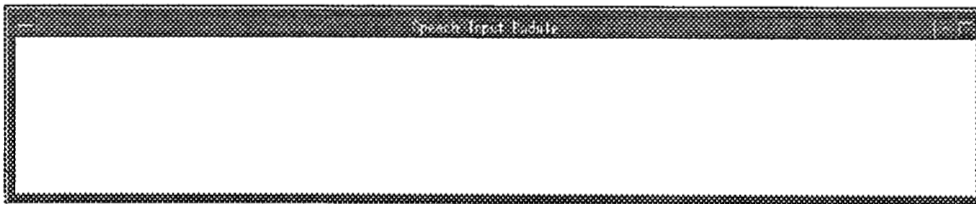


図2.2.1-12 音声入力ウインドウ  
(入力前)



図2.2.1-13 Turn Table  
ウインドウ

3. 発話サイトがこちらの場合 (Turn Table ウインドウが赤) マウスポインターを音声入力ウインドウへいれてマウス左ボタンを押しながらマイクより入力する。
4. 入力がOKならマウス右ボタンをクリック。音声認識処理が開始。

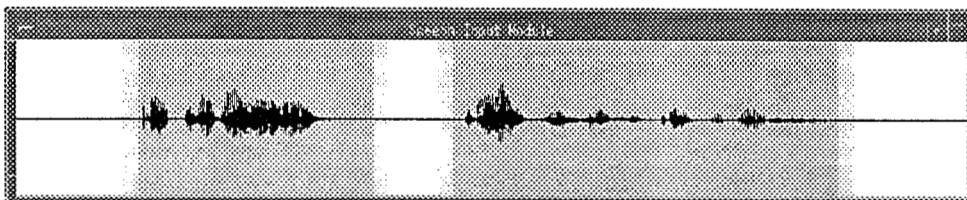


図2.2.1-14 音声入力  
ウインドウ (入力後)



5. 認識文節表示ウインドウ (図2.2.1-15) に次々と認識した文節が表示されていきます。

この時、認識文節表示ウインドウの結果をマウス左ボタンでダブルクリックすると、認識候補ウインドウ (図2.2.1-16) が表示されます。これは認識スコアの高いものから順に表示されます。もう一度ウインドウをダブルクリックすると、ウインドウは閉じます。

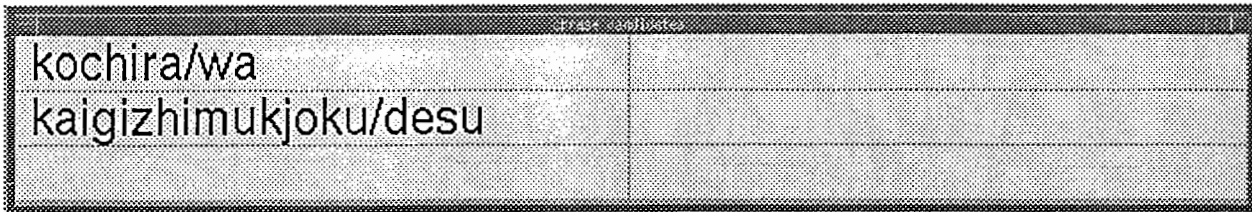


図2.2.1-15 認識文節表示ウインドウ (動作例)

	Recognition	Translation	Score
1	kochira/wa	こちらは	-36.359283
2	kochira	こちら	-35.016647
3	kochira/de/wa	こちらでは	-34.791962
4	kochira/ni/wa	こちらには	-34.768162
5	kochira/o	こちらを	-34.653969

図2.2.1-16 認識候補ウインドウ

6. 認識文表示ウインドウ (図2.2.1-17) に、スコアの高いものから順に文が表示されていきます。

7. 認識文表示ウインドウに表示された候補の中から、正しいものを選び、マウス左ボタンでクリックします。

候補数が多い場合、複数頁に表示されるので次候補を見る必要が出てきます。次頁、前頁はそれぞれマウス右ボタン、マウス中ボタンをクリックすると表示することができます。

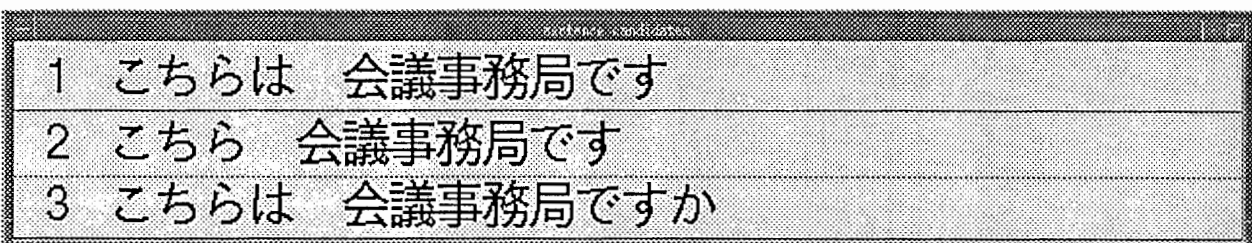


図2.2.1-17 認識文表示ウインドウ (動作例)

8. 認識文が選択されると、翻訳処理が起動され、翻訳結果表示ウインドウ (図2.2.1-18) に翻訳結果が表示されます。

ウインドウ内に入らない文の場合、右スクロール、左スクロールをマウス中ボタン、マウス右ボタンで行なうことができます。

9. 3. ~ 8. を繰り返して行なうことにより連続して入力を行なえます。



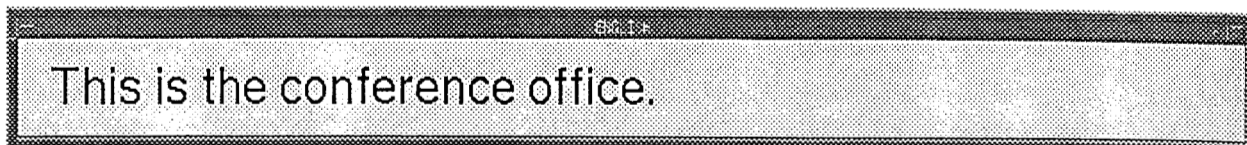


図2.2.1-18 翻訳結果表示ウィンドウ（動作例）



10. こちらの話すことがなくなれば、サイトの交替を行ないます。

右上の Turn Table ウィンドウ（図2.2.1-13, 図2.2.1-19）をマウス左ボタンでダブルクリックします。青色になればOKです。

図2.2.1-19 atrpl1の画面

### 回線を切断する

1. オプションパネルの [Line Close] ボタンをマウス左ボタンでクリックします。
2. メッセージウィンドウが図2.2.1-20の様になれば切断完了です。

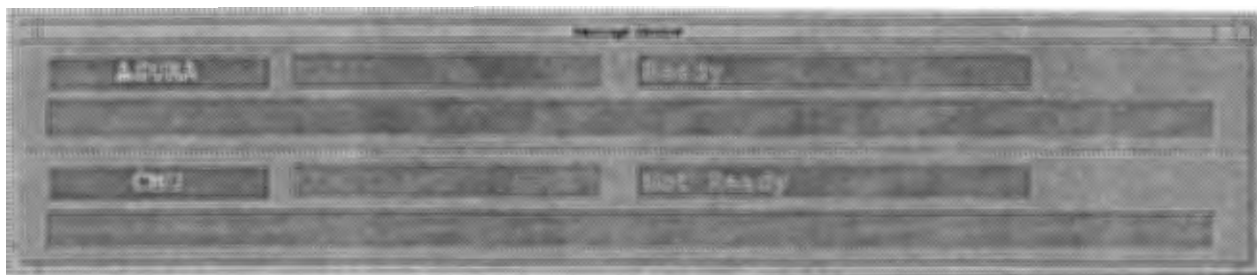


図2.2.1-20 メッセージウィンドウ（回線切断完了）

### ASURAを終了する

1. オプションパネルの [Quit] ボタン（図2.2.1-10-D）をマウス左ボタンでクリックします。
2. すべてのウィンドウがクローズされれば終了です。

## オプションパネルの設定項目

## Select Target Language

接続相手を指定します。翻訳システムを切替えるものではありません。

- [ CMU ] : 米CMUと接続する時に指定
- [ SIEMENS ] : 独シーメンスと接続する時に指定

## ROLE OF SPEAKER

自分の役割を指定します。

- [ CALLER ] : 問合せ者
- [ CONFERENCE OFFICE ] : 会議事務局

## SYSTEM MODE

動作モードを指定します。現在単独モードは指定できません。

- [ Dialog ] : 双方向モード
- [ Alone ] : 単独モード

## Modem Mode

モデム接続時電話の発呼側か、着呼側かを指定します。

- [ CALL ] : 発呼側
- [ CATCH ] : 着呼側

## Select Communication Line

相手との接続方法を指定します。ソケットについては動作を確認していません。

- [ SOCKET ] : ソケット
- [ MODEM ] : モデム

## List of Dialogs

シナリオを指定します。

- [ Dialog 2 ] : (2 J E : 参加料の問い合わせ)
- [ Dialog 3 ] : (3 G J : 会議のテーマについて)
- [ Dialog 6 ] : (6 J E : 市内観光、6 G J : 市内観光)

## Option Window Info

設定値をセーブ、ウインドウクローズの指定。

- [ Save ] : 設定値セーブ
- [ Save & Exit ] : 設定値セーブ、ウインドウクローズ
- [ Exit w/o Saving ] : ウインドウクローズのみ

## DA out

合成音声の再出力の指定

- [ Remote site (Japanese) ] : 日本語音声合成の再出力
- [ Local site (Eng/Grm) ] : 英語の音声合成の再出力(D e c t a l k)
- : 独語は未対応

## Speech Input

音声入力ウィンドウのオープン

- [ Start ] : 音声入力ウィンドウを開く

## Communication

回線接続後の状態チェック、準備完了信号送信

- [ Check ] : 回線状態チェック
- [ Send Ready Signal ] : 準備完了、シナリオ番号、自分の役割送信

## Communication Line Switch

回線オープン、クローズ

- [ Line Open ] : 回線オープン
- [ Line Close ] : 回線クローズ

## ASURA SYSTEM

ASURAシステムの終了、リセット

- [ Quit ] : ASURAシステムの終了
- [ Reset ] : ASURAシステムのリセット

### 2.2.2 音声入力方法

発話する前に、カーソルが音声波形窓の中にあることを確認してください。マウス左ボタンを押しながら、発話を開始してください。文節間に適当なポーズを挿入しましょう。発話が終了したら、マウス左ボタンをはなしてください。平滑化された対数パワーとゼロ交差数をもとに、プログラムは音声区間境界を計算し、それを画面に表示します(図2.1参照)。マウス右ボタンを押すと、切り出された音声データが標準出力に書き出され、音声認識処理が開始します。

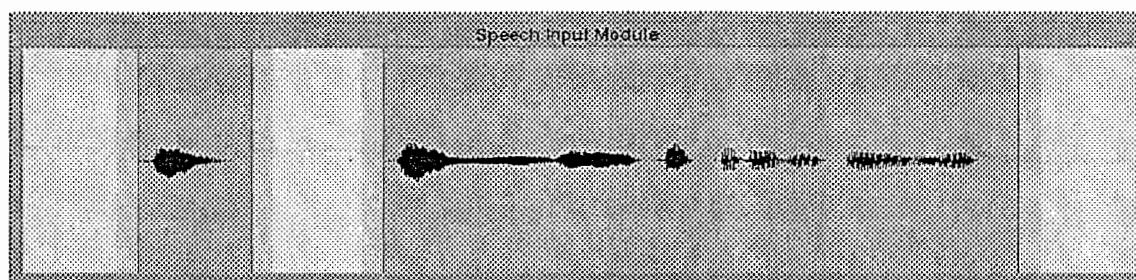


図 2.1: SpeechPress という音声入力窓の動作例。2 文節を発話した状態で、陰影の付いた部分が切り出された音声区間を表している。陰影の付いた部分は X 画面上では淡い青色。

発話している間、音声波形信号は音声波形窓で左から右に表示が進んでいきます。12.5 秒で、音声波形信号は画面の右端に到達し、画面が消去されます。音声波形信号は画面の左端に巻き取られ、再び左端から表示が進んでいきます。全体の音声の最大長は 50 秒に設定されています。

国際共同実験用 ASURA では、2 段階に陰影が施されています。濃い陰影は抽出アルゴリズムによって検出された音声区間を示しています。薄い陰影は安全のために追加された約 50 ms のマージン<sup>1</sup>を示しています。

誤った音声区間境界(例えば、発話の最初の弱い /shi/ など)を修正するために、音声区間境界の移動(シフト+マウス左ボタン)や、削除(シフト+マウス右ボタン)ができます。新しい音声区間境界の追加(シフト+マウス中ボタン)もできます。キーボードから h を押すことにより、いつでもヘルプ画面が表示できます(図 2.2 参照)。プログラムを終了するには、キーボードから q を入力してください。

<sup>1</sup>認識過程で、予測される文字列の最初と最後に HMM 無音モデルが連結されるため。

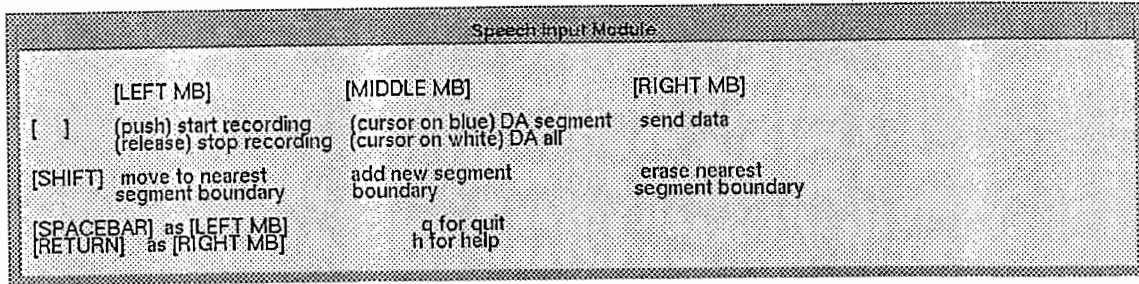


図 2.2: SpeechPress における開始およびヘルプ画面

### 2.2.3 文節音声認識窓とその操作方法

このプロセスウィンドウに表示される情報は、文節音声認識の途中経過から結果に至る流れである。それは、文節毎のウィンドウの中で、文字列の変化として表される。その変化は、以前の文字列と変わった部分の表示色を変えることで表している。また、文節認識の終了後（表示変化が止まった後）、文節窓をマウスでダブルクリックしてやると、各文節の認識候補を、上位5位まで表示したウィンドウが現れる。そして、このウィンドウを閉じるには、再び、現れたウィンドウをダブルクリックすると良い。

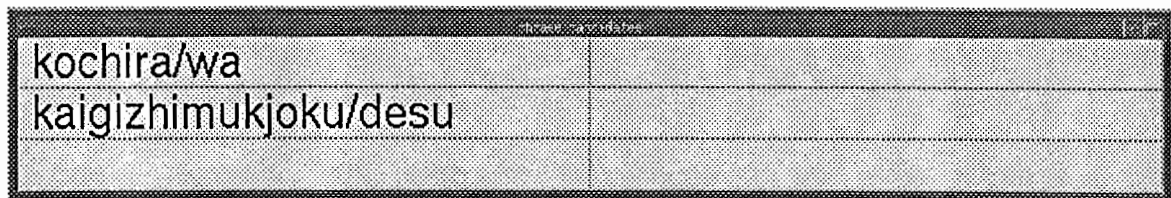


図 2.2.3-1 : 文節音声認識ウィンドウ

|| いずれかのウィンドウで  
|| マウスをダブルクリック  
V

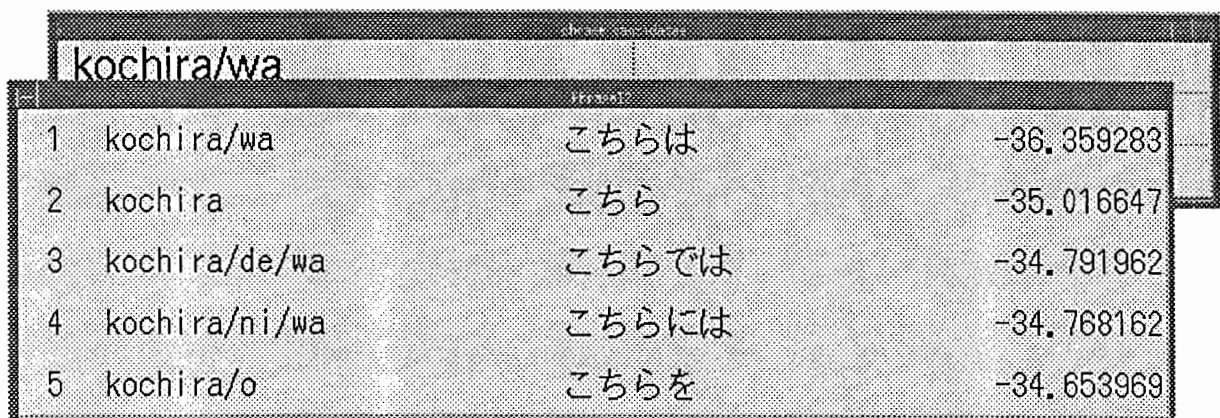


図 2.2.3-2 : 文節認識候補ウィンドウ

### 2.2.4 音声認識結果の文候補表示窓とそこでの文候補選択方法

音声認識処理から出力された文候補 (最高10個) をウインドウ上に表示し、その中から妥当な文候補を選択して、それを言語翻訳処理へ渡すプログラムである (図 2.3 参照)。1 度に表示出来る文候補は最高3候補であり、ウインドウ上でマウス右ボタンをクリックすることにより、ページが進んで次の文候補が見られる。マウス操作は、次の通りである。

- マウスクリック方法
  - － 左ボタン 文候補の選択 (選択された候補のウインドウの色が変わる)
  - － 中ボタン 前ページ
  - － 右ボタン 次ページ

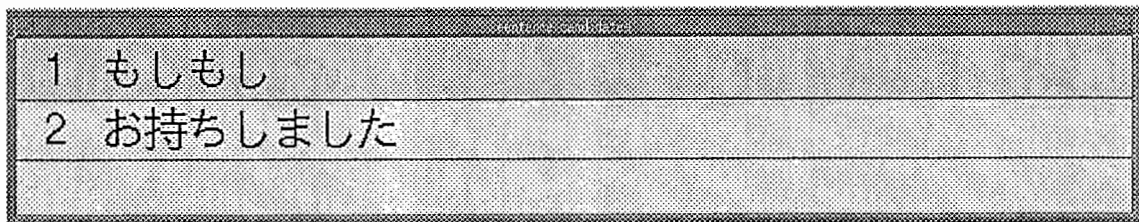


図 2.3: 音声認識結果の文候補表示選択ウインドウ

### 2.2.5 翻訳結果表示窓とその操作方法

言語翻訳処理から出力された結果 (英語、または、ドイツ語) をウインドウ上に表示するプログラムである (図 2.4 参照)。万一、ウインドウの幅を越えるような結果が入ってきた場合のために、横スクロール機能を備えている。マウス操作は、次の通りである。

- マウスクリック方法
  - － 中ボタン 右へスクロール
  - － 右ボタン 左へスクロール



図 2.4: 翻訳結果表示ウインドウ

## 2.3 話者適応

このシステムを始めて使用する際には、使用する方の声の特徴をシステムに登録し、その特徴に基づいて音声認識のためのモデルのパラメータを変更する必要があります。この一連の処理は「話者適応」と呼ばれています。

話者適応を行なうための第一ステップとして、まず使用者があらかじめ決められた内容の言葉(単語、あるいは文節など)を発声し、システムに登録します。

話者適応をするにはまず、次の一連の手順を行なってください。

1. “cstar” というアカウントでログインします。
2. 次に、チェンジディレクトリをします。  
> cd ~/DEMO\_R1.0/RECOG
3. このディレクトリで次のようにしてシェルを起動します。  
> Exe/menu.csh

ここまでの操作が終ると、画面上には図 2.5 のような話者選択画面が表示されます。

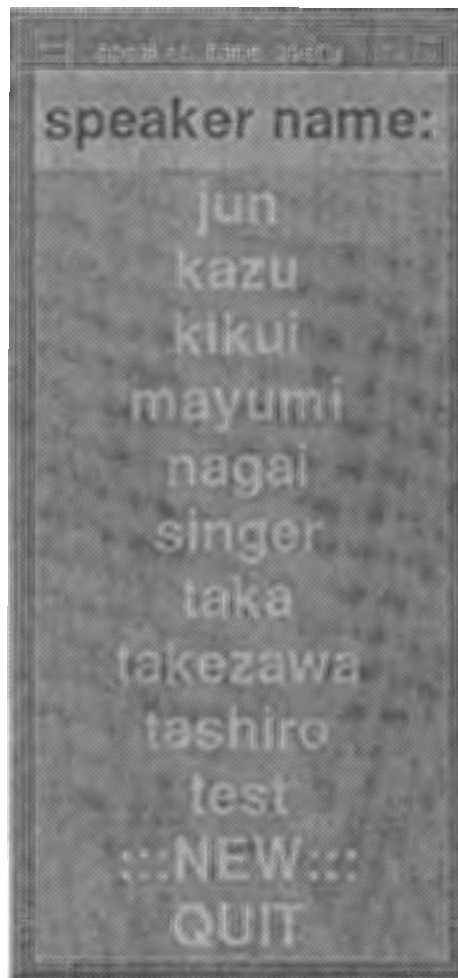


図 2.5: 話者選択画面



始めて登録する場合には、ここで **[::NEW:]** を選択します。すると、図 2.6 のような話者名入力画面が表示されます。話者適応をする人の名前を入力し、**リターンキー** を押します。その際、話者選択画面に既に存在する名前と重複しないように注意してください。



図 2.6: 話者名入力画面

動作環境が変化したりして、話者適応をし直す場合には、図 2.5 の話者選択画面に自分の名前が存在していますから、それをそのままマウスで選択してください。

ここまでの手順が終了すると、画面上には、図 2.7 のようなメニューが表示されています。このメニューで **Voice\_Registration** を選択します。これによって、図 2.8 のようなメニューが表示されます。



図 2.7: システムメニュー画面



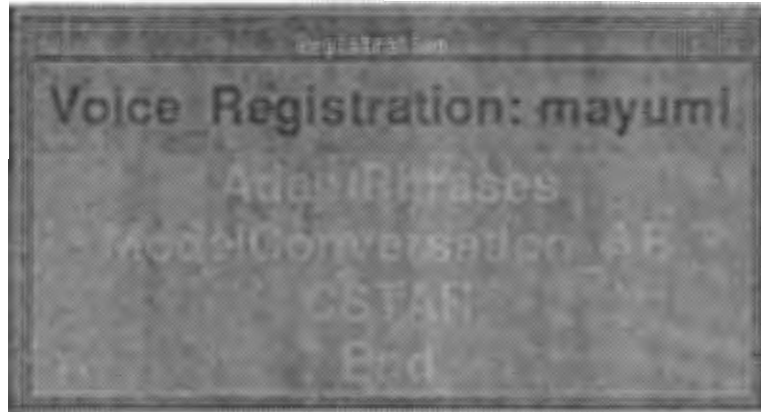


図 2.8: 話者適応メニュー画面

この時、図 2.7や図 2.8のメニューウインドウ上部に表示されている“Speaker:.....”の部分の話者名が自分の名前と一致していることを確認してください<sup>2</sup>(他人の名前になっている時に以下の操作を行なうと、既に登録されている音声データが消えてしまいます)。

図 2.8に表示されている“AdaptPhrase”、“ModelConversation\_AB”、“CSTAR”は、それぞれ登録の際に使用する発声内容を表しています。通常は最も発声量の少ない“AdaptPhrase”を選択すればよいと思いますが、より高い認識率を要求する場合には、“ModelConversation\_AB”(発声量: 中)や“CSTAR”(発声量: 多)を選択して下さい。一般に多く発声する程高い認識率が期待できます(以降は **AdaptPhrase** を選択したもとして話を進めます<sup>3</sup>)。

図 2.9のような画面が表示されたら、いよいよ音声登録の開始です。左上の画面に表示されている言葉を、一行ごとに、間に1秒程度のポーズを入れながら丁寧に発声してシステムに入力して下さい。この入力方法は、2.2.2節に示した通りです。また、波形の切り出し(2.2.2節参照)にも十分に注意して下さい(波形の切り出しが誤っている場合には、うまく話者適応ができないことがあります)。

あらかじめ設定されたすべての言葉の登録が終了すると、図 2.8のメニューに戻ります。特に問題がなければ、**End** を選択して下さい。

これで音声の登録が終了しました。画面には再び図 2.7が表示されるはずですが。次の処理はモデルの変更です。

<sup>2</sup>ここでは、mayumi (鈴木真由美さん) を選択した例を示しています。

<sup>3</sup>CSTAR デモの場合は **CSTAR** を選択することをお勧めします。

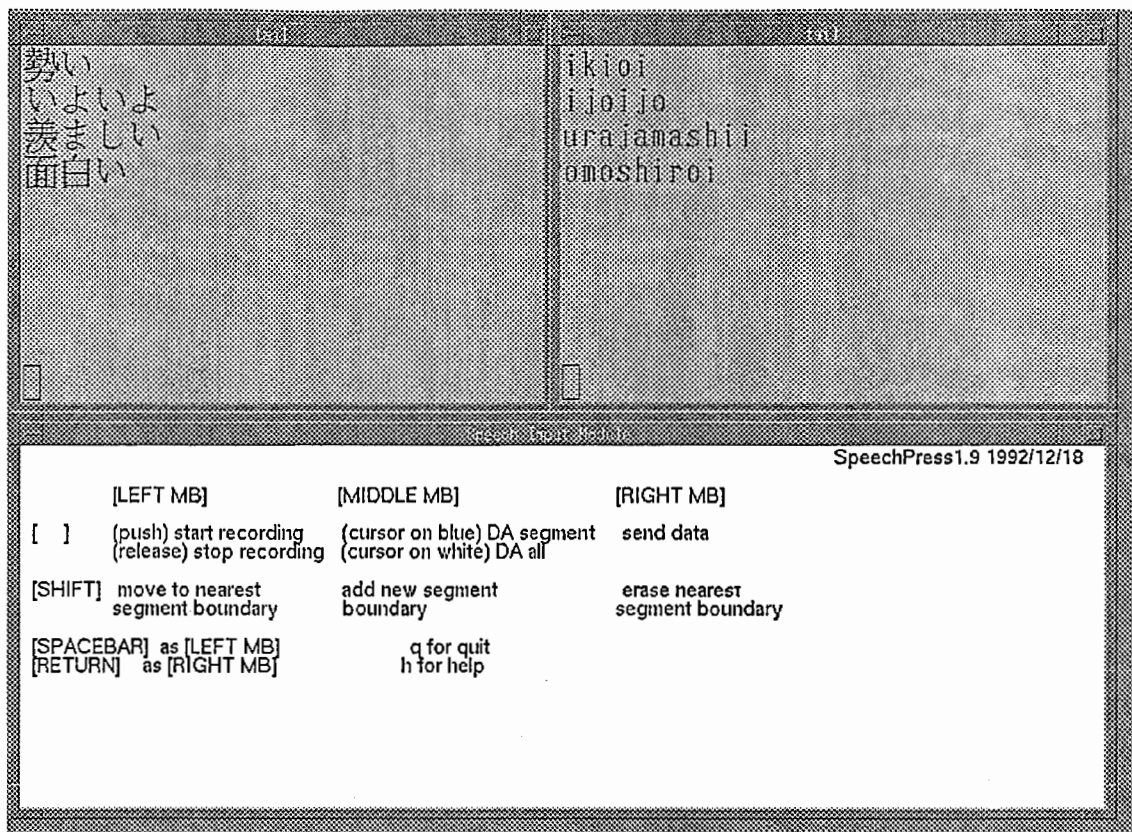


図 2.9: 話者適応入力画面

このためには、メニューの中から **Speaker\_Adaptation** を選択してください。

メニュー選択後、しばらくは何も表示されません。これは先ほど登録した音声に対する分析処理を行なっているためです。

やがて分析処理が終ると、上部に“Standard Speaker Selection”と表示されたウィンドウが表示されます。ここでは、話者適応を行なう際の標準話者の選択が行なわれます。現在のシステムでは、標準話者の候補として、男性3名、女性3名の計6名が登録されています。ここではそれらの中から、使用者の音声の特徴と最も近いと推測される話者のモデルが一つ選択されます。どの話者が標準話者として選択されたかを確認した後、**リターンキー**を押してください。

この後、続いて上部に“Speaker Adaptation”と表示されたウィンドウが表示されます。ここでは先ほど選択された標準話者のモデルパラメータの変更が行なわれます。所要時間は1分弱です。

以上で話者適応に関するすべての処理が終了します<sup>4</sup>。これ以降、このシステムを使用する場合には、ここで作られた使用者に専用のモデルが使用されます。

<sup>4</sup>CSTAR デモの場合、ASURA 本体は図 2.7 からでは起動できないので、**Exit** を選択して話者適応プログラムを終了してください。daily demo 用 ASURA は図 2.7 で **Speech\_Translation** を選択すれば起動できます。

## 2.4 ローカルテスト用ツールの基本操作方法

ASURA システムの動作試験などのためにローカルテスト用ツールを開発した。このツールは、相手側システムの代わりに、音声入力・音声認識・言語翻訳の各モジュールの模擬をすることができる。本節では、このツールの基本的な操作方法を紹介する。

まず、このローカルテスト用ツールを動作させるにあたって必要な環境あるいは設定を述べる。

- X ウィンドウの環境が整っていること。
- モデムが接続されていること。
- shell の中で実行ファイルの設定がなされていること。

ここでは、上記3点を満足しているマシン as26 を使用した例を示して、基本的な操作のみを説明する<sup>5</sup>。

1. as26 に “cstar” というアカウントでログインする。
2. “xinit” と入力して、X ウィンドウを起動する。
3. “cd /home/cstar/DEMO\_R1.0/TOOL/P-JANUS” と入力して、ディレクトリを移動する。
4. 起動方法  
次のコマンドによりツールを起動する。
  - (a) 米国 CMU の JANUS システムを模擬する場合 (「疑似 JANUS」モード)
    - JANUS.as26.sh
  - (b) ドイツ Siemens 社のシステムを模擬する場合 (「疑似 Siemens システム」モード)
    - SIEMENS.as26.sh

### 5. 回線の設定

メインウィンドウ (図 2.10 参照) 上端メニュー中の **Option** をクリックして、プルダウンメニュー中の **Option Panel** を選択し、オプションパネルウィンドウ (図 2.11 参照) を開く。ASURA 側に電話の着呼側か、発呼側かの確認を取ってから、右側 “MODEM” 中の着呼側 (“RECEIVER”) か、発呼側 (“CALLER”) のトグルボタンをクリックする (トグルボタンが IN 側に入っている方が選択されている状態)。同じウィンドウ右下の **OK** ボタンをクリックして、ウィンドウを閉じる (あるいは、**CANCEL** ボタンをクリックすることで、キャンセルしてウィンドウを閉じることもできる)。

### 6. 回線の接続

---

<sup>5</sup>詳しくは、3.12 節で説明する。

## (a) 電話の着呼側の場合

メインウインドウ上端メニュー中の **Command** をクリックし、プルダウンメニュー中の **Connect Line** を選択する。メインウインドウの“JAPANESE”の下に“Ready”というメッセージが表示されたら接続完了。

## (b) 電話の発呼側の場合

相手側の準備が完了してから **Connect Line** を選択する。自動的に電話がかけられて、回線の接続に成功すると、メインウインドウの“JAPANESE”の下に、“Ready”というメッセージが表示される。

## 7. シナリオ選択の準備

シナリオ選択のためには、まず、メインウインドウ上端メニュー中の **File** をクリックして、プルダウンメニュー中の **Select Scenario** を選択し、シナリオセレクションウインドウ (図 2.12参照) を開く。そのウインドウ右側の“Files”中のファイル名をダブルクリックすると、シナリオウインドウ (図 2.13参照) が開かれる。

## 8. シナリオ番号の送信

シナリオウインドウが表示されたら、まず最初に上のコマンドボタンの **Send Scenario No** を押して、相手側にシナリオ番号を送信する。

## 9. センテンス等の送信

その後は、コーパスにそって対話を進めていくことになる。メインウインドウの“ENGLISH”の下に、“Ready”というメッセージが表示されたら、発話サイトがこちら側になる。センテンスを送信する方法は、シナリオウインドウのセンテンスをダブルクリックするか、シングルクリックして上のコマンドボタンの **Send Sentence** を選択するか、の2つの方法がある。また、相手側にサイトを切替える時は、上のコマンドボタンの **Send Over** を選択する。

## 10. 終了方法

終了するには、まず、メインウインドウ上端メニュー中の **Command** をクリックして、プルダウンメニュー中の **Disconnect Line** を選択し、回線を切断する。次に、メインウインドウ上端メニュー中の **File** をクリックして、プルダウンメニュー中の **Exit** を選択すると、プロセスが終了する。

## 11. メッセージウインドウについて

一番下のメッセージウインドウ (図 2.14参照) では、上のウインドウがこちら側の、下のウインドウが相手側のそれぞれ認識結果と翻訳結果を表示する。



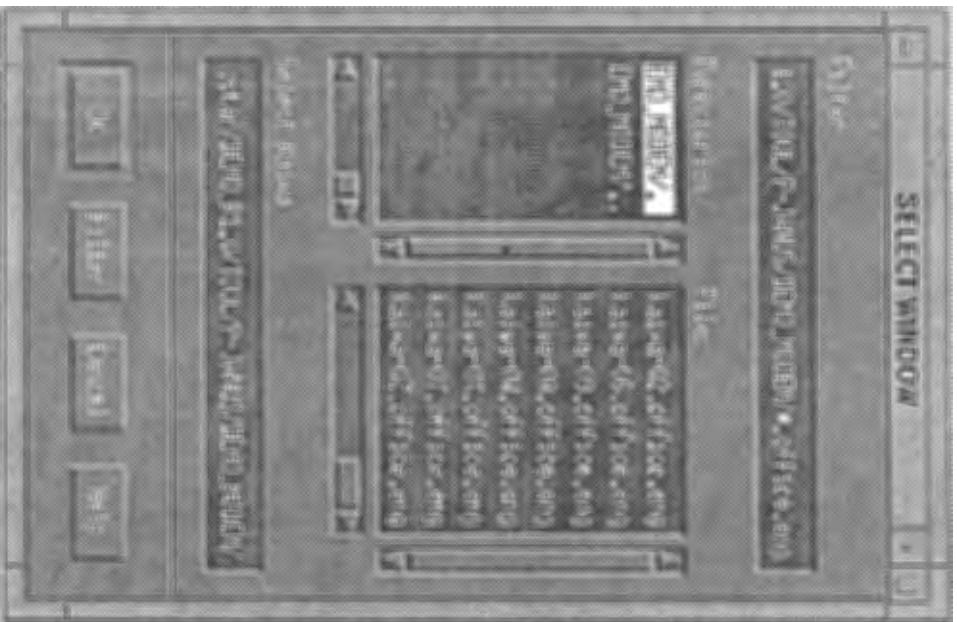


図2.1.2：セレクションウインドウ

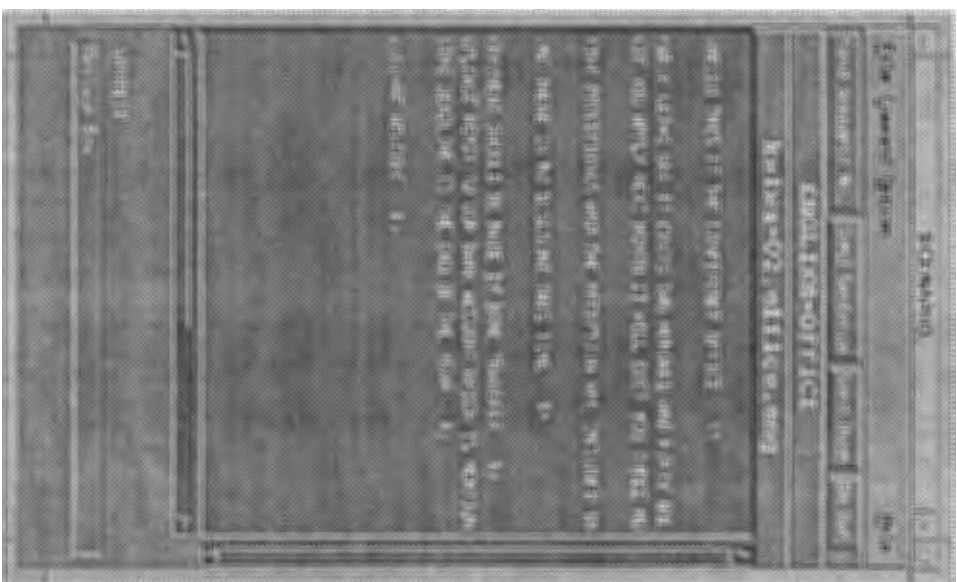


図2.1.3：シナリオウインドウ

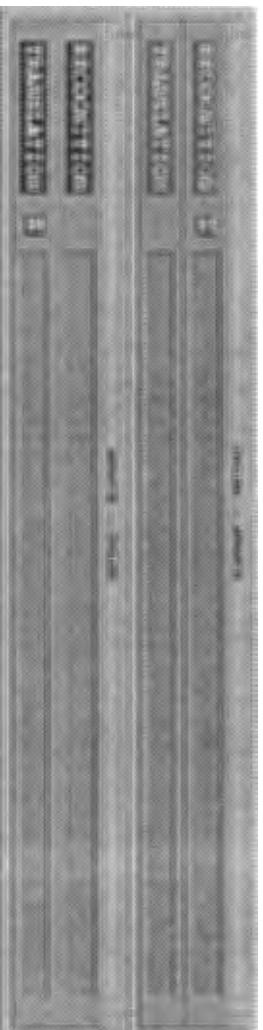


図2.1.4：メッセージウインドウ

## 第 3 章

### 自動翻訳電話国際共同実験用 ASURA の実装

第3章は自動翻訳電話国際共同実験用 ASURA のプログラマ用マニュアルを目指している。実際のプログラムの実装方法が知りたい人やプログラムに改良を加える必要がある人は第2章に加えてさらにここも読むとよい。

#### 3.1 動作環境

CSTAR (自動翻訳国際共同実験) に使用した環境を記す。

##### 主要マシン

###### 通信制御

マシン名 : as22 ( Sun SPARC2 )  
メモリー : 32Mbytes  
OS : SunOS Release 4.1.1  
window : X11R4 + OFS/Motif

###### 音声認識

マシン名 : atrp11 ( HP 9000/750 )  
メモリー : 304Mbytes  
OS : HP-UX 8.07  
window : X11R4 + OFS/Motif

###### 音声合成

マシン名 : atrp17 ( HP 9000/730 )  
メモリー : 96Mbytes  
OS : HP-UX 8.07

###### 言語解析および変換生成

マシン名 : atrp19 ( HP 9000/750 )  
メモリー : 256Mbytes  
OS : HP-UX 8.07

###### DDX-Pへの接続

マシン名 : atr-sp ( VAX 8820 )  
メモリー : 80Mbytes  
OS : ULTRIX 3.1

###### 音声分析装置

マシン名 : IOPBOX  
社名 : システムデザインサービス株式会社

## 通信用機材

## 公衆電話回線モデム

社名 : モトローラ社  
品名 : CODEX 3266 DATA MODEM

※ 接続対象となるマシンは as22

## パケット回線 (DDX-P)

社名 : 日本ダイナテック株式会社  
品名 : Multi-PAD X.25

※ 接続対象となるマシンは atr-sp

## プログラミング環境

## L I S P

バージョン : HP Common Lisp B.04.0.1

## C

各マシン、備え付けのものを使用。

ウィンドウは、Xlib並びにMotifツールキットを使用。

## シェルスクリプト

基本的には、/bin/csh を使用。



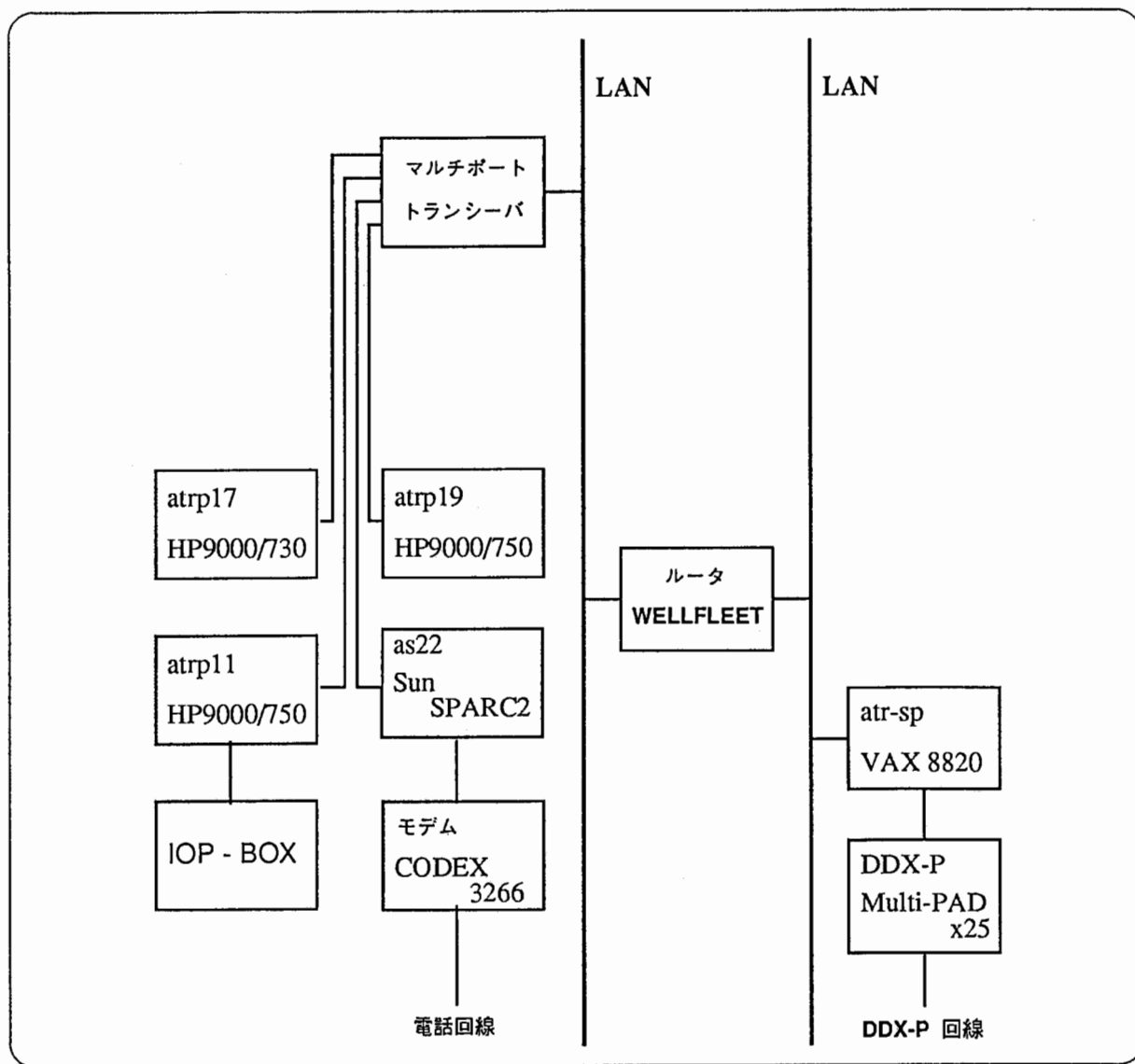


図 3.1: マシン環境図

## 3.2 プログラムの構成

### 3.2.1 プロセス構成について

CSTAR 版 ASURA システム全体をプロセスレベルでながめてみると大きく三つに分割することができる。それらは、認識、翻訳、通信という集まりである。その三つの、それぞれの集まりは、独立したシステムとしてプロセス群を構成している。また、その三つのシステムは、ソケット通信を使うことによってお互いの同期とデータのやりとりを行ない、このような大きなシステム“CSTAR 版 ASURA”として成り立っている。

以下、それぞれ認識、翻訳、通信のシステム構成について述べる。

### 3.2.2 認識系システム構成

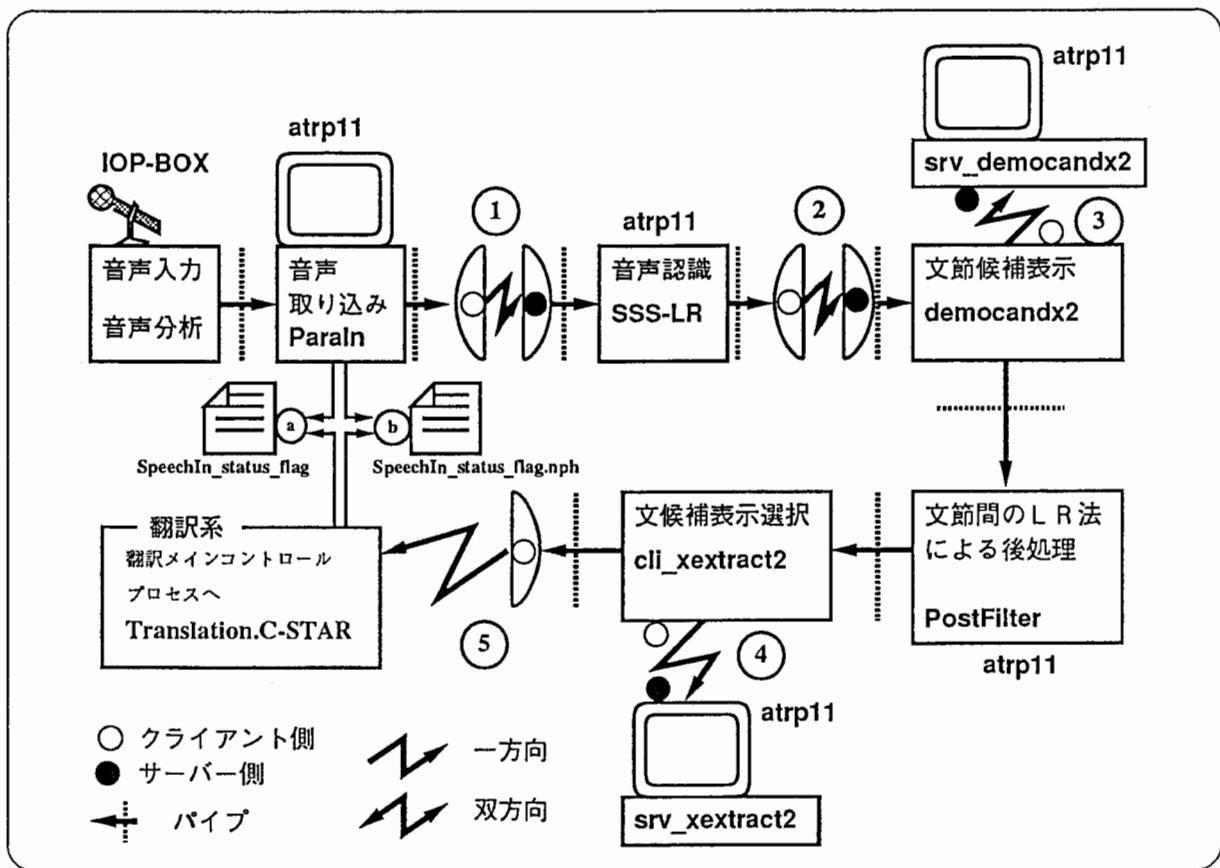


図 3.2: 認識系プロセス構成図

図 3.2を見ると、次のような点が特徴として挙げられる。

- 一つのプロセスは、ツールとして成り立ち、それは最小限の独立した機能を持つ。
- UNIX のパイプラインを使用し、小さなプロセスツールを接続している。
- 関連する小さなプロセスのグループの集まりである。

- パイプラインで作られたそれぞれのプロセスグループは、ソケットを使って接続されている。そして、それは、ソケットツールというものを使うことで標準入出力データでの通信を可能としている。

ここで、このソケットツールについての説明を行なう。

#### ○ソケットツールについて

プロセス名	: socket
ソースディレクトリ	: ~/DEMO_R1.0/RECOG/SRC/SOCKET (atrp11)
概要	: プロセス間通信にソケットフィルターとして使用。 シェルレベルでの tcp-ip 仕様のソケット接続が可能。 オプション指定によりクライアントとサーバの使い分けができる。
オプション	: -s ... サーバ指定 (default : クライアント) -r ... 接続されたソケットから読み込みのみ。 -w ... 接続されたソケットから書き込みのみ。 -q ... stdin から EOF を受け取るとクローズする。 -l ... ループモードになる。 ※ここで紹介する以外にもオプションは存在するが、 他のオプションに関しては、上で記述したディレクトリの README を参照されたいし。
使用方法	: パイプラインの前後にこのプロセスをパイプでつなげる。 その時、接続したいパイプラインどうしのソケット番号指定。

#### 1. 接続形態

図 3.2 に示したものが、認識系システム全体のプロセスの構成状態である。

①, ②, ⑤ は、上で説明を行なったソケットツールによる接続状態を示したものである。③, ④ は、メインとなるプロセスと画面表示のみに使用するプロセスとの間を結ぶソケットとなっている。ここで使用しているソケットはすべて INET ドメイン (tcp-ip) 仕様である。また、ポート番号は直接指定 (/etc/services は使用しない)。以降、詳細は統括制御、または各プロセス説明を参照されたいし。

#### 2. ソケットポート番号

以下に認識系で使用しているソケットのポート番号を記す。

- ① ..... 10022 : ParaIn — SSS-LR
- ② ..... 10021 : SSS-LR — democandx
- ③ ..... 10024 : democandx2 — srv\_democandx2
- ④ ..... 10025 : cli\_xextract2 — srv\_xextract2
- ⑤ ..... 10023 : cli\_xextract2 — 翻訳系へ

#### 3. 認識系の状態の把握方法

ここでは翻訳系がどのようにして認識系の状態を把握しているかその方法を記す。まず、主な状態として以下のものが挙げられる。

- 音声入力開始
- 音声入力終了
- 音声認識開始
- 音声入力準備 OK
- 発話権を相手に渡す

これらの状態を知るため ParaIn (音声取り込み) と翻訳メインコントロールプロセス間で共有するファイルを持つ。これは図中の ④, ⑤ で表す。次に、共有するファイルについての説明を行なう。

SpeechIn_status_flag	……	このファイルはファイルモードの変化を利用している。
		読み込みのみ (-r--r--r--)
	…	音声入力準備 OK
		実行可能のみ (---x--x--x)
	…	音声入力開始
		書き込みのみ (--w--w--w-)
	…	音声入力終了 および 音声認識開始
SpeechIn_status_flag.nph	…	このファイルは ParaIn で得た 文節数が書き込まれる。
		n = 文節数
	n = 0	… 発話権を相手に渡す
	n > 0	… 音声入力および音声認識開始

以上の状態を翻訳側のコントロールプロセスが覗く形となる。

#### 4. ソケットで流されるデータ内容

- ① … HMM のために切り出された尤度値
- ② … 認識途中経過、認識結果、並びに候補、および全認識時間
- ③ … 認識途中経過、認識結果、並びに候補
- ④ … 認識文結果並びに候補
- ⑤ … 最適認識結果一文

以上がソケット間で、受渡しされているデータである。これら以外にパイプにより、接続された間のデータのやりとりはそれぞれのパートにおいて説明がなされている。

## 3.2.3 翻訳系システム構成

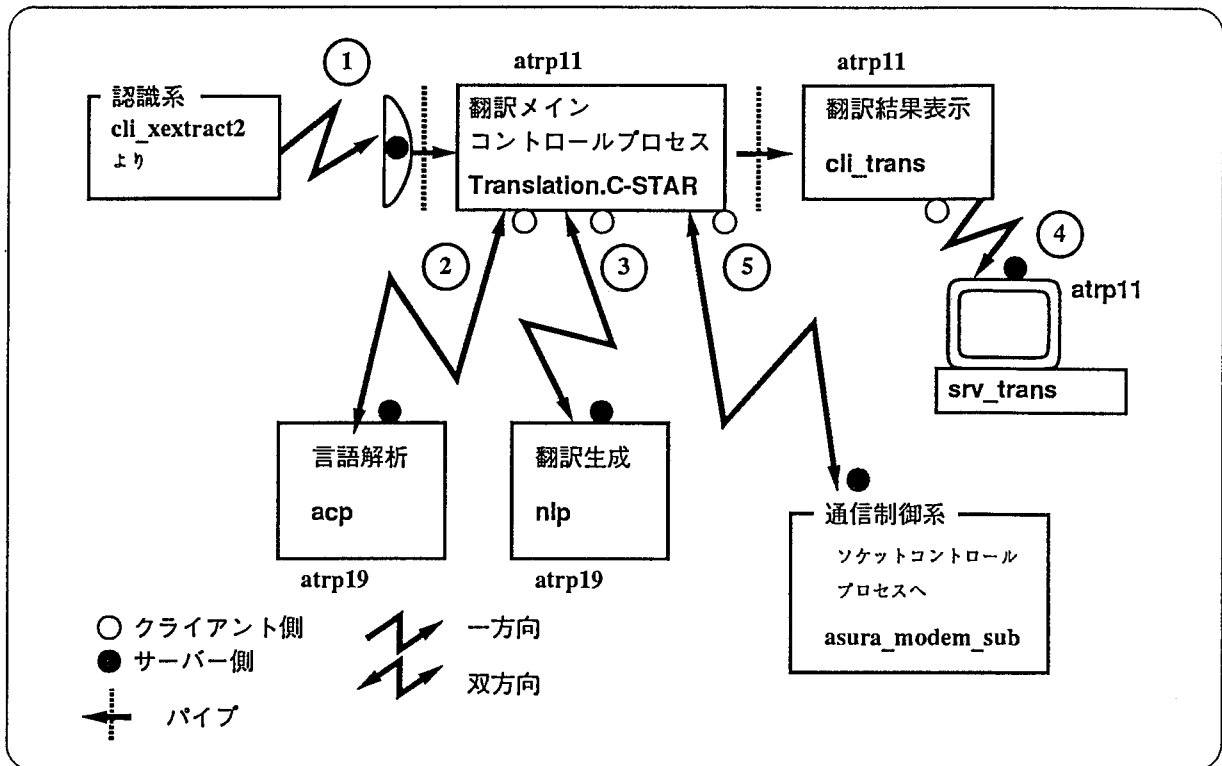


図 3.3: 翻訳系プロセス構成図

## 1. 接続状態

ここで使用されているソケットは、すべて INET ドメイン (tcp-ip) 仕様となっている。

①, ④, ⑤ は、ポート番号直接指定。① は認識系側との socket プロセスとの接続となっている。②, ③ は、サービスファイル (/etc/services) を読む形となっている。

## 2. ソケットポート番号

- ① …… 10023 : 認識系 — Translation.C-STAR
- ② …… 10050 : Translation.C-STAR — acp (サービス名 : cstar-acp)
- ③ …… 10051 : Translation.C-STAR — nlp (サービス名 : cstar-nlp)
- ③ …… 10052 : Translation.C-STAR — nlp2 (サービス名 : cstar-nlp2)
- ④ …… 10026 : cli\_trans — srv\_trans
- ⑤ …… 10001 : Translation.C-STAR — 通信制御へ

※ 10051 のソケットは対アメリカ、10052 のソケットは対ドイツ。

## 3. システム状態の把握方法

システム全体の状態は、翻訳システムコントロールプロセスがすべてを担っている。またこのプロセスによって ASURA システムの同期が保たれている。図 3.3 中での ① ~ ④ のソケットに関しては、その処理ごとの結果のやりとりのため接続がされている。唯一、多少使用目的が違っているのが ⑤ のソケットである。このソケットでは、

音声入力から翻訳までの状態を決められたメッセージとして送受信している。よって、このソケットでは随時メッセージのやりとりが行なわれている。この作業によって相手側に ASURA システムの処理状態を知らすことができる。以降、詳細は統括制御を参照のこと。

#### 4. ソケットで流されているデータの内容

- ① … 最適認識結果文一文
  - ② … コントロールプロセスからは ① で結果文、言語解析からは解析結果
  - ③ … コントロールプロセスからは、解析結果、変換生成からは翻訳結果
  - ④ … 翻訳結果またはエラーメッセージ
  - ⑤ … ASURA システムの各状態
- それぞれのデータの詳細は各パートにおいて説明がなされる。

### 3.2.4 通信制御系システム構成

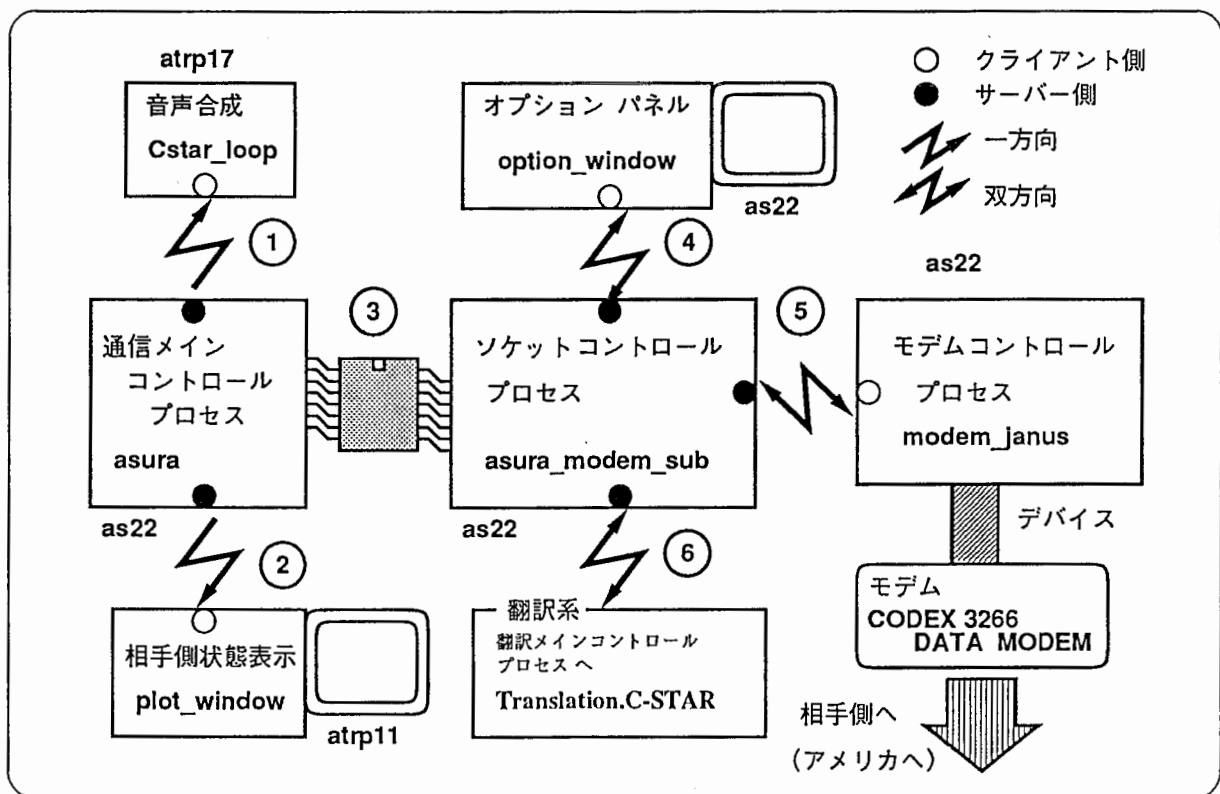


図 3.4: 通信制御系プロセス構成図 (対アメリカ)

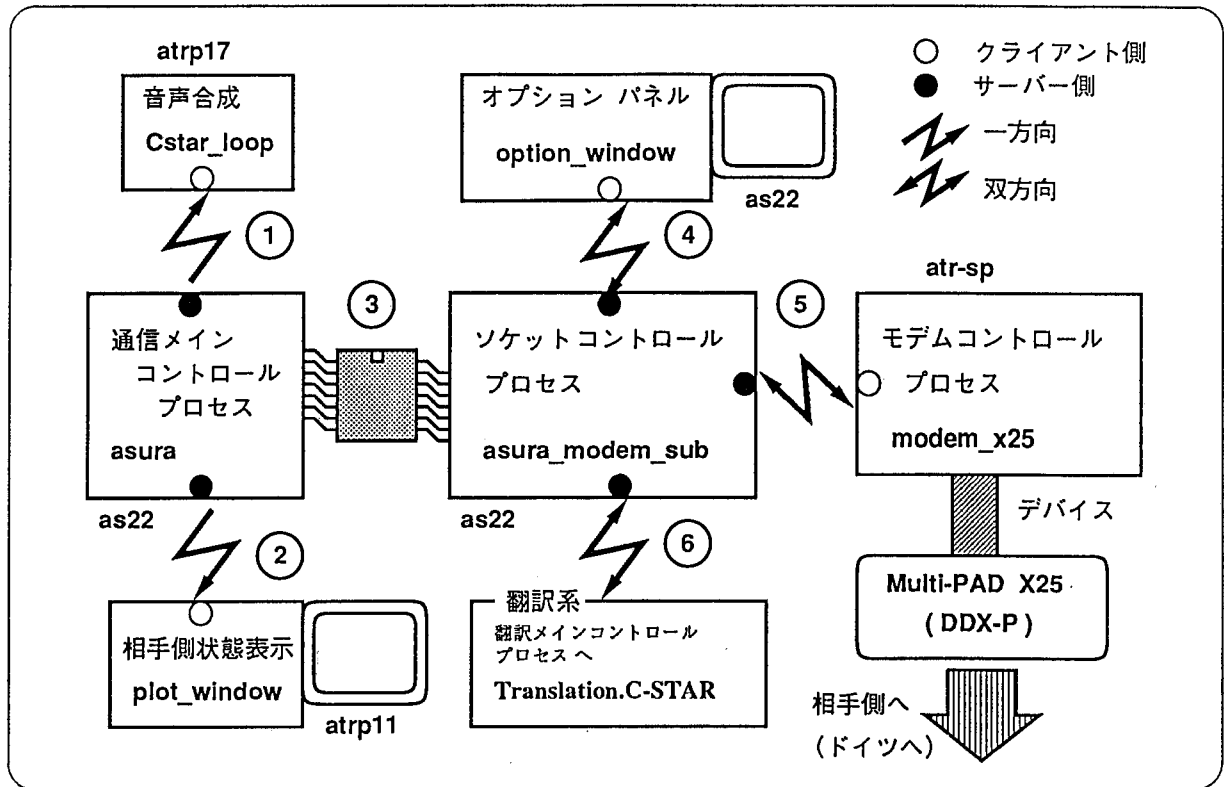


図 3.5: 通信制御系プロセス構成図 (対ドイツ)

## 1. 接続状態

ここでのソケットは ①, ②, ④, ⑤, ⑥ である。これらはすべて INET ドメイン仕様である。ポート番号はすべて直接指定。また、ここではアメリカとドイツとでソケットコントロールプロセスが接続する相手が違う。

アメリカの場合… 接続相手は modem\_janus

ドイツの場合… 接続相手は modem\_x25

となっている。③ に関しては、これは共有メモリを使用している。この共有メモリに両プロセスが互いにアクセスすることでデータをチェックしあっている。

## 2. ソケットポート番号および共有メモリ

- ① …… 10001 : asura — Cstar-loop
- ② …… 10002 : asura — plot\_window
- ④ …… 10004 : asura\_modem\_sub — option\_window
- ⑤ …… 10011 : asura\_modem\_sub — modem\_janus か modem\_x25
- ③ において、共有メモリの番号は 6809 を使用。

## 3. 通信制御系の構造

ここで、メインのプロセスとなるのが asura である。asura は asura\_modem\_sub が得たデータを共有メモリに格納したものを読み、そのデータに対応した処理をそれぞれ

れのプロセスへ送る。また、翻訳系、オプションパネルまたは相手側システムに対しては再び共有メモリに処理データを格納する。そのデータを `asura_modem_sub` が読み込み再びソケットを通して送り出す。全体の通信の制御を行なうインタフェースは、`option_window` である。オペレータは、この画面により通信の制御を行なう。`plot_window` に関しては、相手側の結果および途中経過を表示する画面であり、そして、その表示された結果を音声として出力するのが音声合成となる。次に、海外との通信回線だが、これは、アメリカ側とドイツ側で使用されている回線が異なる。これはドイツ側とのモデムでの接続がスムーズにできなかったため、公衆回線を使用せず、パケット回線により接続が行なわれた。以降、詳細は通信制御を参照されたし。

#### 4. ソケットで流されるデータ内容

- ① … 相手側から送られてきた合成用のフォーマットを持った文字列
- ② … 途中経過および認識、翻訳それぞれの結果
- ③ … コントロール用のメッセージ文字列
- ④ … 通信プロトコルに対応した文字列
- ⑤ … ASURAシステムの各状態

以上がデータ内容である。以降、詳細は各パートの説明を参照されたし。



## 3.3 統括制御

### 3.3.1 概要

このシステムは三カ国間(日本、ドイツ、アメリカ)で行なわれた自動翻訳電話の実験システムである。ここでは、起動シェルスクリプトおよび翻訳システムコントロールプロセス(Translation.C-STAR)についての説明を行なう。

ここで説明するものは、このシステムで使用された音声認識および翻訳システムの同期をとるためのコントロールプロセス、および、これらを起動するシェルスクリプト等である。

### 3.3.2 ディレクトリ構成

- 起動シェルスクリプトディレクトリ
  - /users/cstar/DEMO\_R1.0/RECOG/
  - Exe\_C-STAR/
    - +- SSS\_C-STAR.csh
    - | SSS\_C-STAR.csh.options
    - | boot\_SSS.csh
    - | honyakuserv\_C-STAR.csh
    - | reco\_C-STAR.csh
    - | site\_change\_C-STAR.csh
    - | ATR\_LOGO.bmap
    - |
    - +-- LOGS/
  
- 翻訳メインコントロールプロセスディレクトリ
  - /users/cstar/DEMO\_R1.0/TRANS/
  - filterdemo/
    - +- hserv2\_C-STAR.csh
    - | kill\_SSS.csh
    - | kill\_TRANS.csh
    - | kill\_TRANS\_SSS.csh
    - | socket
    - |
    - +-- translation.C-STAR/
      - +- Translation.C-STAR
      - |
      - | +-- TRANS\_SRC/
      - |
      - | +-- TRANS\_SRC.930119/
      - |
      - | +-- TEST/
      - |
      - +-- resultfile/

### 3.3.3 ディレクトリ説明

- ◎ Exe\_C-STAR/
  - ◇ ATR\_LOGO.bmap ... ターンテーブルウィンドウに使用している ATR のビットマップファイル

※ このディレクトリにある他のシェルスクリプトについては

3.3.4 節『主なシェルスクリプト』で説明を行なう

○ LOGS ... ログファイルを置いておくディレクトリ

◎ filterdemo/

◇ hserv2\_C-STAR.csh ... コントロールプロセス (Translation.C-STAR) に起動をかけるシェルスクリプト

◇ kill\_TRANS\_SSS.csh, kill\_SSS.csh, kill\_TRANS.csh ... システムを終了させるシェルスクリプト  
kill\_TRANS\_SSS.csh だけを実行すれば良い

◇ socket ... SSS-LRとコントロールプロセスをソケットにより接続する実行イメージ  
このプロセスについては、3.2.2 節『認識系システム構成』で説明を行なっている

○ translation.C-STAR/

◇ Translation.C-STAR ... コントロールプロセスの実行イメージ本体

◇ TRANS\_SRC/ ... コントロールプロセス (Translation.C-STAR) のソース  
このディレクトリで make を行ないイメージを一つ上のディレクトリに移動する

◇ TRANS\_SRC.930119/ ... バックアップ

◇ TEST/ ... テストのためのディレクトリ

○ resultfile/ ... ログファイルを置いておくディレクトリ

### 3.3.4 主なシェルスクリプト

~cstar/DEMO\_R1.0/RECOG/Exec\_C-STAR/.

boot\_SSS.csh :

CSTAR版ASURAシステムの中核となる、音声認識、言語翻訳に関するプロセス群を起動するシェルスクリプトを実行する。

SSS\_C-STAR.csh :

音声認識、言語翻訳に関するプロセス群のための環境設定、並びにオプション設定シェルスクリプトの実行。その他、話者名選択要求、各プロセス起動シェルスクリプトの実行をする。

SSS\_C-STAR.csh.options :

各プロセス起動時に必要なオプションをシェル変数として設定する。また、そのオプションの意味については、3章10節の『システムパラメータの初期設定と変更』の項を参照されたい。

reco\_C-STAR.csh :

音声認識に関するプロセスの起動。その中には、メインディスプレイ関係のプロセス、並びに、プロセス間通信のためのソケットフィルターなども含まれる。

音声取り込み装置であるIOPBOXの初期化プロセスの実行。

言語翻訳に関するシェルスクリプトの実行。



```
[-german]
[-both]
[-socket]
[-display machine]
```

- -acp machine  
 言語解析システムを立ち上げるマシン指定。現在、CSTAR用とdaily ASURA用のものがある。  
 cstartp20 : CSTAR用が atrp20 で起動  
 cstartp19 : CSTAR用が atrp19 で起動  
 asurap20 : daily用が atrp20 で起動  
 asurap19 : daily用が atrp19 で起動  
 as26 : daily用が as26 で起動  
 < default : cstartp20 >
- -enlp machine  
 英語用変換生成システムを立ち上げるマシン指定。  
 オプションは言語解析システム時と同様。  
 < default : cstartp20 >
- -gnlp machine  
 ドイツ語用変換生成システムを立ち上げるマシン指定。  
 オプションは言語解析システム時と同様。  
 < default : cstartp20 >
- -asura  
 SpeechIn との同期をとる。  
 これは常に指定すること。
- -v  
 言語解析、変換生成時のメッセージを標準出力に出す。  
 翻訳結果表示のため。  
 これは常に指定すること。
- -english  
 英語用システムの指定。
- -german  
 ドイツ語用システムの指定。
- -both  
 英語用、ドイツ語用の両システムの指定。  
 CSTAR時には使用しない。
- -socket  
 通信制御との接続の方法。  
 これは常に指定すること。
- -display machine  
 通信制御システムの起動マシン指定。

この指定は 3.10 節『システムパラメータの初期設定と変更』を参照されたし。

```
< default : atrp20 >
```

### 3. 言語解析、変換生成の起動方法

言語解析、変換生成の二つのシステムは、このコントロールプロセスから起動がかけられている。言語解析システム起動を例に挙げると、

```
char *acp_command[] = {
"remsh atrp20 'cd /users/project/asura/demo/cstar/english;com-interface/acp >
/users/project/asura/demo/cstar/eval/outputfile/ACP.LOG' &",
"remsh atrp20 'cd /users/project/asura/demo/english; com-interface/acp >
/users/project/asura/demo/filterdemo/resultfile/ACP.LOG' &",
"remsh atrp19 'cd /users/project/asura/demo/cstar/english;com-interface/acp >
/users/project/asura/demo/cstar/eval/outputfile/ACP.LOG' &",
"remsh atrp19 'cd /users/project/asura/demo/english;com-interface/acp >
/users/project/asura/demo/filterdemo/resultfile/ACP.LOG' &",
"remsh as26 'cd /usr/project/asura/demo/english;com-interface/acp > /dev/null' &",
" "
};
```

というような文字配列<sup>1</sup>を持ち、これらをオプション (-acp, -enlp, -gnlp) により指定し、選択された文字列を直接 system 関数によりこのプログラム内から起動している。これは、変換生成システムも同じ方法である。

### 4. システム全体の同期の取りかた

---

<sup>1</sup>フォントを小さくしても入り切らないので、この例では、文字列の途中で折り返しているものがある。

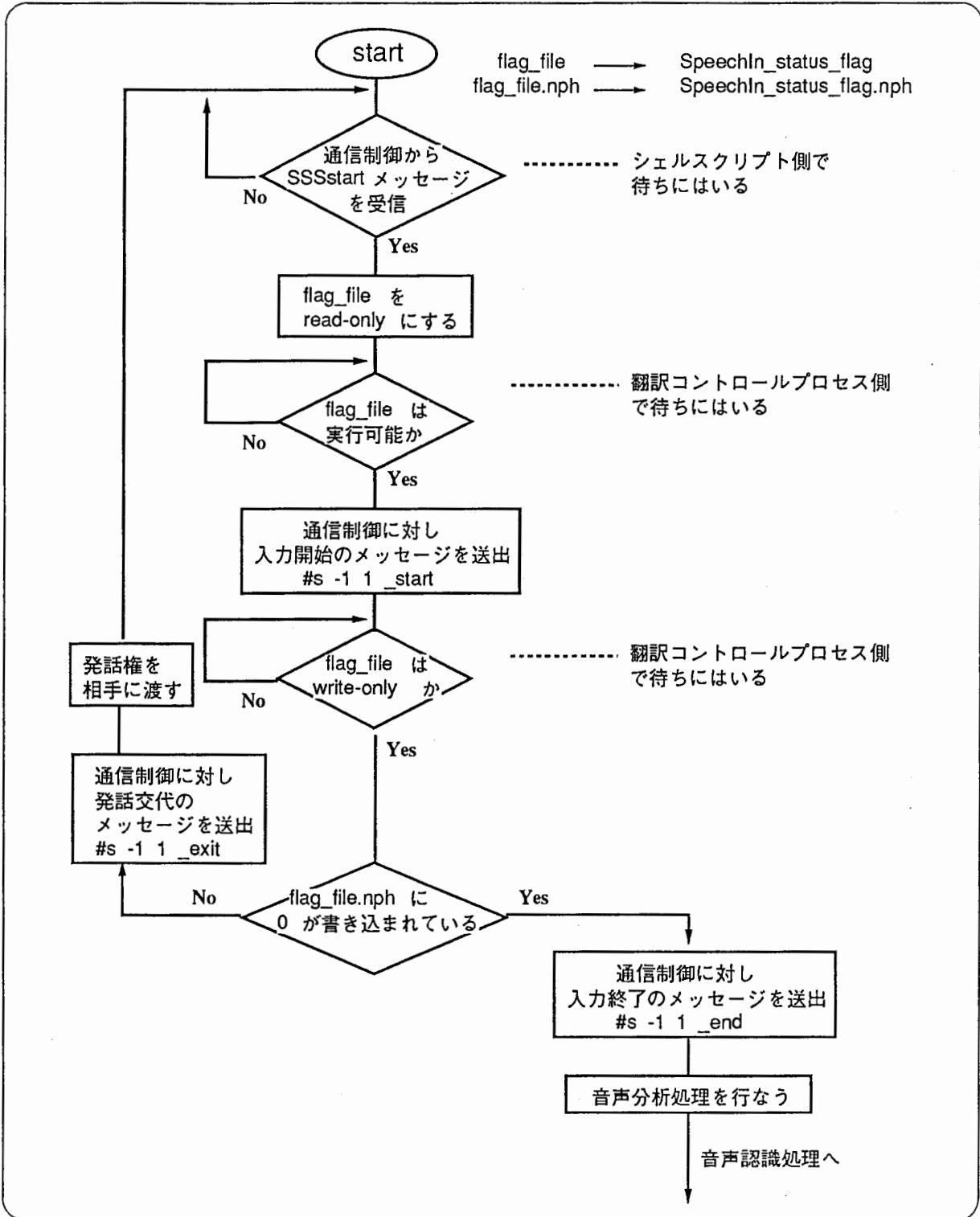


図 3.6: システムの同期の取り方

図 3.6は翻訳システムコントロールプロセス、音声入力そして通信制御との間のやりとりである。これらのやりとりの状態を常に通信制御に知らせている。

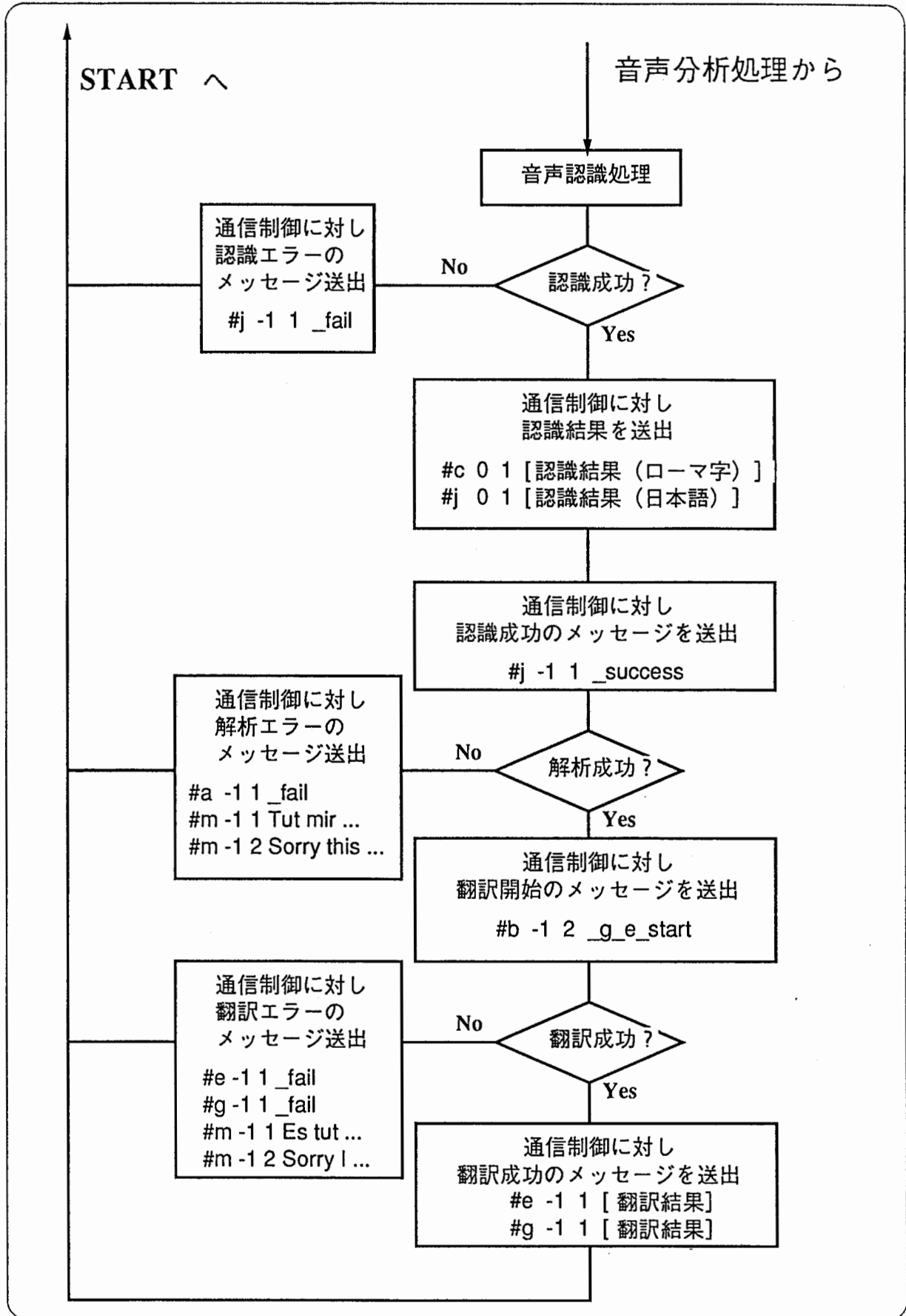


図 3.7: 音声認識、言語解析、変換生成の間のやりとり

図3.7が音声認識、言語解析、変換生成とのやりとりである。言語解析の処理は

```
『#j -1 1 _success』
```

のメッセージを受けた後、変換生成の処理は

```
『#e(g) -1 1 _g_e_start』
```

のメッセージを受けた後、コントロールプロセスからそれぞれ認識結果、解析結果をもらう。そして、それぞれの処理が終了した後、再びコントロールプロセスに対しこれらの処理結果（解析結果、翻訳結果）を返す。

すべての処理は一旦、翻訳システムコントロールプロセスに対して結果を渡し、その結果をコントロールプロセスがチェックをし、次の処理へと進めていくのである。この間もコントロールプロセスからの通信制御への情報は常に渡されている。

### 3.3.6 プロセス間のメッセージ書式

a. s SSSstart

1. #s -1 1 \_start

2. #s -1 1 \_end

3. #c 0 1 kaigino aidani shinaikaNkooga arusoodesuga

4. #j 0 1 会議の間に市内観光があるようですが

5. #j -1 1 \_success

6. #b -1 2 \_g\_e\_start

7. #e -1 1 I've heard that there's city sightseeing during the conference.

(英語)

7. #g -1 1 ich habe gehoert dass es eine stadtrundfahrt waehrend der konferenz gibt.

(ドイツ語)

8. #s -1 1 \_exit

以上が基本的な動作メッセージの流れである。

このほかに、処理エラー時のメッセージがある。

1'. #j -1 1 \_fail

2'. #a -1 1 \_fail

3'. #e -1 1 \_fail

4'. #g -1 1 \_fail

5'. #m -1 2 Sorry this is difficult. Could you change the wording?

(英語)

5'. #m -1 1 Tut mir leid, das ist schwierig. K nten Sie einen anderen Aus druck benutzen?

(ドイツ語)

6'. #m -1 2 Sorry I couldn't handle that sentence.

(英語)

6'. #m -1 1 Es tut mir leid, diesen Satz konnte ich nicht verarbeiten.

(ドイツ語)

ここまでが主なメッセージ群である。

これ以降に各メッセージの意味を記述する。



## ○ 基本メッセージ

- a … モデム処理との処理プロセスから送出されてくるメッセージ。  
このメッセージにより音声入力がOKとなる。
- 1 … 音声入力開始時に送出するメッセージ。
- 2 … 音声入力終了時に送出するメッセージ。  
このメッセージを音声認識開始のメッセージとする。
- 3,4 … 音声認識結果。3 は ローマ字結果  
4 は 日本語結果
- 5 … 音声認識成功。認識終了メッセージ
- 6 … 変換生成開始メッセージ。
- 7 … 翻訳結果。
- 8 … 発話権を相手側に渡す時に送出するメッセージ。  
これは音声入力ウィンドウの右側に表示されている  
ウィンドウをマウスクリック（左クリック）する  
ことで送られる。

## ○ 処理エラーメッセージ

- 1' … 音声認識エラー時に送出されるメッセージ。
- 2' … 言語解析エラー時に送出されるメッセージ。
- 3' ~ 6' … 変換生成エラー時に送出されるメッセージ。

## 3.4 音声認識

### 3.4.1 音声入力窓

IOPBOX がデモシステムに接続される場合は、SpeechPress のハードウェア版が使用される。このハードウェア版は ParaIn と呼ばれ、SpeechPress と正確に同じユーザインタフェースを提供するが、切り出された音声波形データの代わりに、HMM のための切り出された尤度値を標準出力に書き出す。これを実現するためには、まず DSP と呼ばれる別のプログラムで HMM モデルと DSP プログラムをハードウェアにダウンロードしてから、音声入力プログラム ParaIn を起動しなければならない。

SpeechPress に関するもっと詳しい説明は文献 [6] を参照のこと。

### 3.4.2 文節音声認識窓

#### 概要

このプロセスは、文節音声認識における途中経過および結果を、Motif ウィンドウ上に表示するものである。そして、これは標準的な UNIX コマンドフィルタと同じように、UNIX のパイプで接続し使用するものである。

#### 作成にあたり

音声認識には、現在のコンピュータの性能ではある程度の時間を要してしまう。これは、どうしようもないことだが、それならば、その結果の出るまでの時間をただ待つのではなく、何か途中経過のようなものを見せることで、音声認識をより感じていただく、ということとなった。また、そのために、以前 HMM-LR などのデモで使われていた parse\_tree<sup>2</sup> と呼ばれている音声認識窓を利用すると、ディスプレイにおけるスペースの問題などがあり、今回、新たに、このようなツールを作成することとなった。

#### 仕様

- UNIX コマンドフィルタとして、標準入力から受け取った、音声認識プロセスからの文節認識データを、ディスプレイ上に逐次表示する。
- 標準出力へは、入力データから認識経過を表しているデータを削除したものを渡す。
- 画面構成については、X11 上に、Motif のウィンドウツールキットを使って、実現するものとする。
- プロセスは、大きく 2 つに分け、1 つは標準入出力への読み書きを担当し、もう一方は、X ウィンドウの表示窓への描画を担当するものとする。互いのプロセス間のやり取りは、ソケット通信によるものとする (図 3.8 参照)。
- 文節認識の経過および結果は、文節毎に与えられた表示窓に対して、文字列の変化として表す。また、その文字列の更新部分の色を変えることで、その変化を明示する。

<sup>2</sup>音声認識の途中経過を、木の枝の分岐のような文字列のツリーによって、ディスプレイ上に表示するツール。

- 一度に表示できる文節の数は、表示窓の数と等しい6つまでとする。
- 各文節認識表示窓を、ディスプレイのマウスでダブルクリックすると、その文節の認識候補について、第5位まで表示した窓を新たに出現させる。この窓は、いなくなれば、マウスで再びそれ自身をダブルクリックすることで消える。

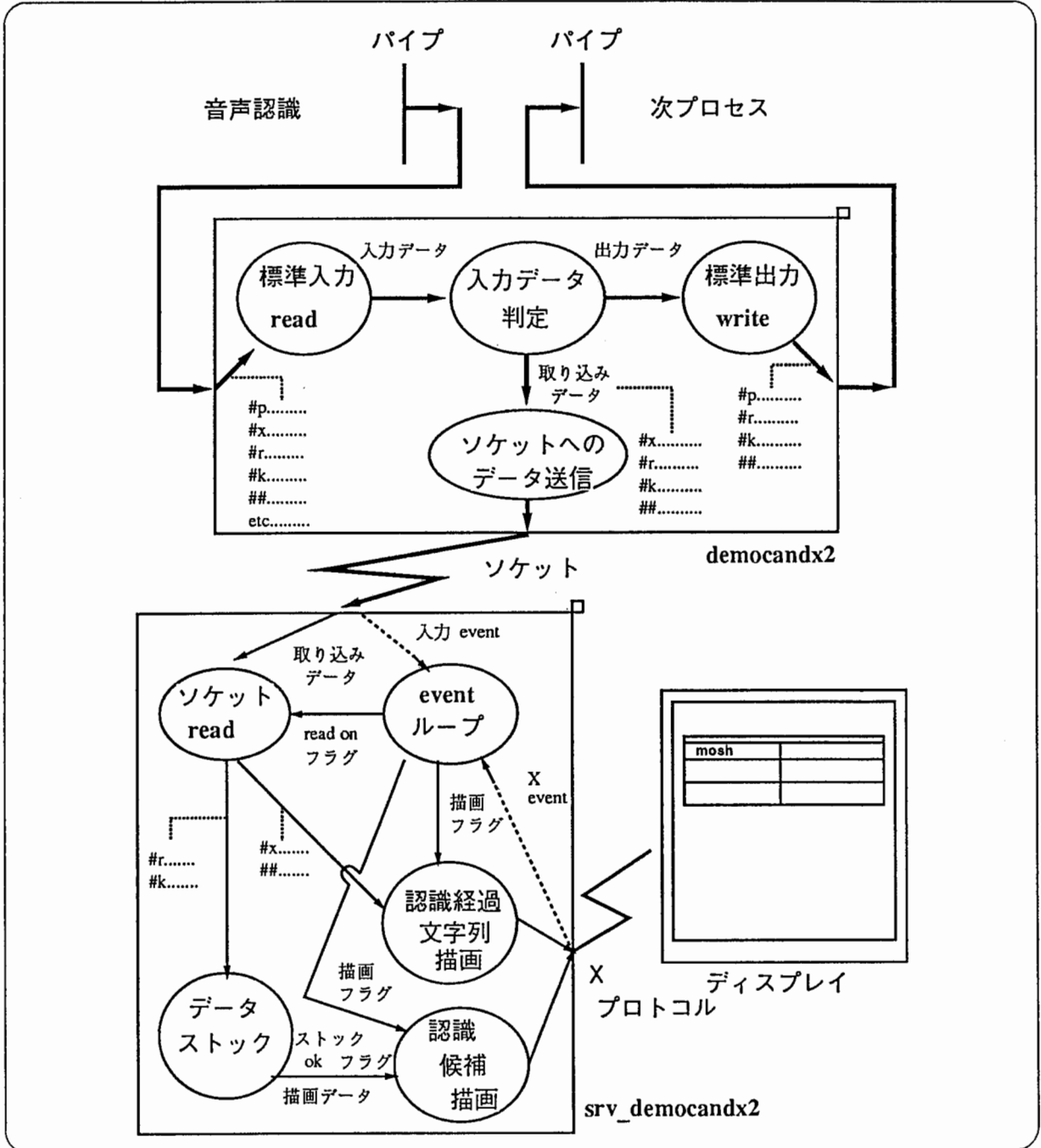


図 3.8: 文節音声認識窓プログラム構成図

## 標準入力からのデータサンプル

```

#p 0 1 AllBunsetsuTime= 4
#p 0 1 TimeStamp= 720374046
#x 0 1 m
#x 0 1 mo
#x 0 1 mosh
#x 0 1 moshi
#x 0 1 moshim
#x 0 1 moshimo
#x 0 1 moshimosh
#x 0 1 moshimoshi
#x 0 1 moshimoshi
#x 0 1 moshimoshi
#x 0 1 moshimoshi
#x 0 1 moshimoshi
#x 0 1 moshimoshi
#x 0 1 moshimoshi
#p 0 1 CPU= 1430 msec Elapsed= 2 sec TotalVerify= 858 Acc= 26 Depth= 16
#r 0 1 1 moshimoshi (-35.680016)
#k 0 1 もしもし <interj> -35.680016
#r 0 1 2 mooshiwake/e (-31.975001)
#k 0 1 申し訳へ <np-e> -31.975001
#r 0 1 3 mooshiwake (-31.874506)
#k 0 1 申し訳 <np> -31.874506
#r 0 1 4 mooshiwake/o (-31.293635)
#k 0 1 申し訳を <np-o> -31.293635
#r 0 1 5 moo/shi/mashi/ta (-31.246238)
#k 0 1 申しました <vaux> -31.246238
## 0 1

```

## 標準出力へのデータサンプル

入力データより、認識経過データを削除したものの。

```

#p 0 1 AllBunsetsuTime= 4
#p 0 1 TimeStamp= 720374046
#p 0 1 CPU= 1430 msec Elapsed= 2 sec TotalVerify= 858 Acc= 26 Depth= 16
#r 0 1 1 moshimoshi (-35.680016)
#k 0 1 もしもし <interj> -35.680016
#r 0 1 2 mooshiwake/e (-31.975001)
#k 0 1 申し訳へ <np-e> -31.975001
#r 0 1 3 mooshiwake (-31.874506)
#k 0 1 申し訳 <np> -31.874506
#r 0 1 4 mooshiwake/o (-31.293635)
#k 0 1 申し訳を <np-o> -31.293635
#r 0 1 5 moo/shi/mashi/ta (-31.246238)
#k 0 1 申しました <vaux> -31.246238
## 0 1

```

## ソースファイルの所在

```
~cstar/DEMO_R1.0/RECOG/SRC/DEMOCANDX2/.
```

## コンパイル

```
make -f makefile.{hp, sun, dec, alpha} install
```

を、実行することで、

```
../BIN{HP, SUN, DEC, ALPHA}/.
```

それぞれの場合の、ディレクトリーに、実行イメージを作製してくれる。

## 動作環境

X ウィンドウシステム バージョン 11 (X11) と共に、Motif X ウィンドウシステムの走っている、HP9000 シリーズ、SUN SPARC ステーション、DEC ステーション、DEC Alpha AXP で、動作確認している。また、X11 については、R4 もしくは R5 を望み、一部ロードできないフォントがある場合など、代用フォントで動作は保証される。

## 使用方法

文節音声認識プロセス | 予

```
democandx2 ... 予
```

| 次プロセス (翻訳系プロセス等..)

Usage :

democandx2 予 : クライアント実行ファイル名

options 予 : オプション

Socket Number 予 : ソケットポート番号

SaverMachineName 予 : サーバ実行マシン名

srv\_democandx2 : サーバ実行ファイル名

options :

-title CandX : ウィンドウタイトル名

-geometry 200x1260+0+180 : Xウィンドウジオメトリ

-display atrp11 : X表示ディスプレイ名

## 使用しているフォント

・日本語フォント :

```
"-typebank-gothic-medium-r-normal--36-360-75-75-c-360-jisx0208.1983-0"
```

```
"-typebank-gothic-medium-r-normal--36-360-75-75-c-180-jisx0201.1976-0"
```

```
"j16.32x32"
```

```
"jeuc.16x32"
```

```
"-typebank-gothic-medium-r-normal--24-240-75-75-c-240-jisx0208.1983-0"
```

```
"-typebank-gothic-medium-r-normal--24-240-75-75-c-120-jisx0201.1976-0"
```

```
"kanji24"
```

```
"rk24"
```

```
"k24"
```

```
"r24"
```

```
"k14"
```

"r14"

上記、いずれかのフォント。

・アスキーフォント：

"-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1"

"-adobe-helvetica-bold-r-normal--14-140-75-75-p-82-iso8859-1"

"-adobe-helvetica-bold-r-normal--18-180-75-75-p-103-iso8859-1"

"-adobe-helvetica-bold-r-normal--24-240-75-75-p-138-iso8859-1"

"-adobe-helvetica-medium-r-normal--34-240-100-100-p-176-iso8859-1"

"-adobe-helvetica-medium-r-normal--40-280-100-100-p-200-iso8859-1"

"-adobe-helvetica-medium-r-normal--48-480-75-75-p-260-iso8859-1"

フォントはウインドウの高さによって選択される。

- なお、フォントサイズ、36, 40, 48のものについては、大画面表示のために急ぎで作製したものであり、標準的な X11 の環境では、存在しないと思われる。また、作成にあたっては、そのサイズの  $\frac{1}{2}$  の大きさのフォントのデータを単純に2倍に拡大したものである。

### 3.4.3 音声認識結果の文候補表示選択窓

#### 仕様

音声認識処理から出力された文候補 (最高10個) をウインドウ上に表示し、その中から妥当な文候補を選択して、それを言語翻訳処理へ渡すプログラムである。このプログラムは、認識文候補データを標準入力を受け取り、選択された文候補を標準出力に出力するクライアントプロセスと、クライアントから受け取った認識文候補データをウインドウに表示し、選択された文候補をクライアントへ渡すサーバープロセスとに分けられている (図 3.9 参照)。1 度に表示出来る文候補は最高3候補であり、ウインドウ上でマウス右ボタンをクリックすることにより、ページが進んで次の文候補が見られる。マウス操作は、次の通りである。

#### • マウスクリック方法

1. 左ボタン 文候補の選択 (選択された候補のウインドウの色が変わる)
2. 中ボタン 前ページ
3. 右ボタン 次ページ

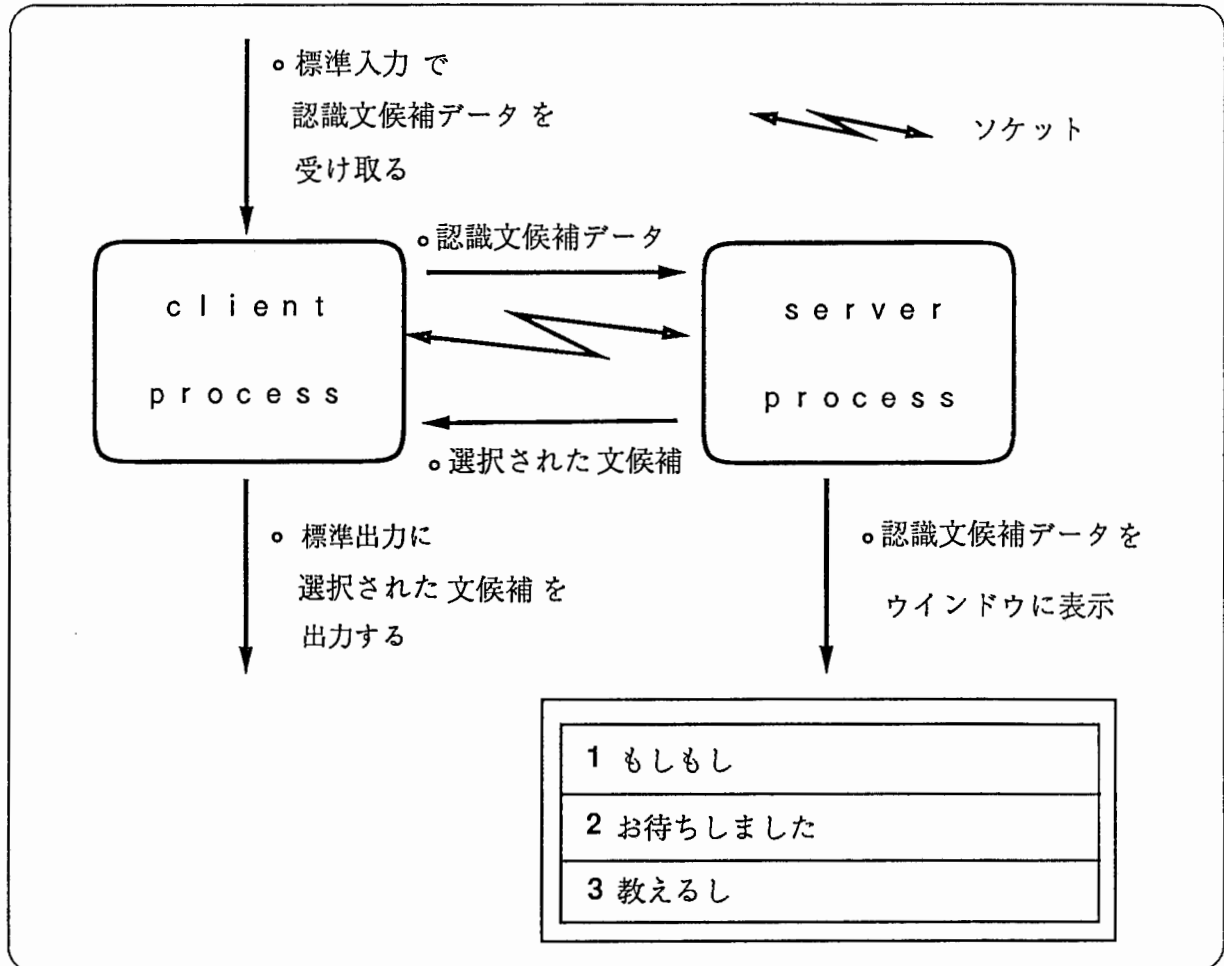


図 3.9: 音声認識結果の文候補表示選択窓プロセスのしくみ

#### 標準入力から入力されるデータ例

```
#o 0 1 kochira/wa#kaigizhimukjoku/desu (score = -39.158642)
#o 0 1 hoteru/wa#kaigizhimukjoku/desu (score = -38.133793)
#o 0 1 kochira#kaigizhimukjoku/desu (score = -38.034500)
#j 0 1 こちらは # 会議事務局です (score = -39.158642)
#j 0 1 ホテルは # 会議事務局です (score = -38.133793)
#j 0 1 こちらは # 会議事務局です (score = -38.034500)
## 0 1
```

#### 標準出力へ出力されるデータ例 (選択された認識候補)

```
#o 0 1 kochira/wa#kaigizhimukjoku/desu (score = -39.158642)
## 0 1
#j 0 1 こちらは # 会議事務局です (score = -39.158642)
## 0 1
```

## ソースファイルの所在地

- クライアント → /home/cstar/DEMO\_R1.0/RECOG/SRC/XEXTRACT2/Ver1.2.930205/PARENT
- サーバ → /home/cstar/DEMO\_R1.0/RECOG/SRC/XEXTRACT2/Ver1.2.930205/DISP

## コンパイル

- HP 仕様: make installで../../../../BINHPの中に実行ファイルができる。
- SUN 仕様: make -f makefile.sun installで../../../../BINSUNの中に実行ファイルができる。

## 実行方法

“標準入力から入力されるデータ例”で述べたようなデータをパイプを通じて標準入力を受け取り、実行させる。

```

client_process_name      : クライアント実行ファイル名
  [-t <titlebar> ]      : ウィンドウタイトル名
  [-display <display_name:0.0> ] : 表示するマシン名
  [-fg <foreground color> ] : フォアグラウンド名
  [-bg <background color> ] : バックグラウンド名
port_number              : ソケットポート番号
hostname                 : 起動するマシン名
server_process_name     : サーバ実行ファイル名

```

## 使用しているフォント

## • 漢字:

```

"k14"
"j16.16x18"
"-typebank-gothic-medium-r-normal--18-180-75-75-c-180-jisx0208.1983-0"
"-typebank-gothic-medium-r-normal--24-240-75-75-c-240-jisx0208.1983-0"
"j16.24x24"
"*-fixed-medium-r-normal--24-*-*-*-*-*--jisx0208.1983-*"
"-typebank-gothic-medium-r-normal--32-*-*-*-*-*--jis*-*"
"tbgm36k"
"tbgm48k"

```

## • アスキー:

```

"-adobe-helvetica-bold-o-normal--8-80-75-75-p-50-iso8859-1"
"-adobe-helvetica-bold-r-normal--10-100-75-75-p-60-iso8859-1"
"-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1"
"-adobe-helvetica-bold-r-normal--14-140-75-75-p-82-iso8859-1"
"helvR36"
"helvR48"

```

- フォントはウィンドウの高さによって変わるように設定している。
- なお "tbgm36k", "tbgm48k", "helvR36", "helvR48"は、大画面表示のために特別に作成した。
- "tbgm36k", "tbgm48k", "helvR36", "helvR48"は、それぞれ
 

```

"-typebank-gothic-medium-r-normal--18-180-75-75-c-180-jisx0208.1983-0"
"-typebank-gothic-medium-r-normal--24-240-75-75-c-240-jisx0208.1983-0"

```



```
"-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1"
```

```
"-adobe-helvetica-medium-r-normal--24-240-75-75-p-130-iso8859-1"
```

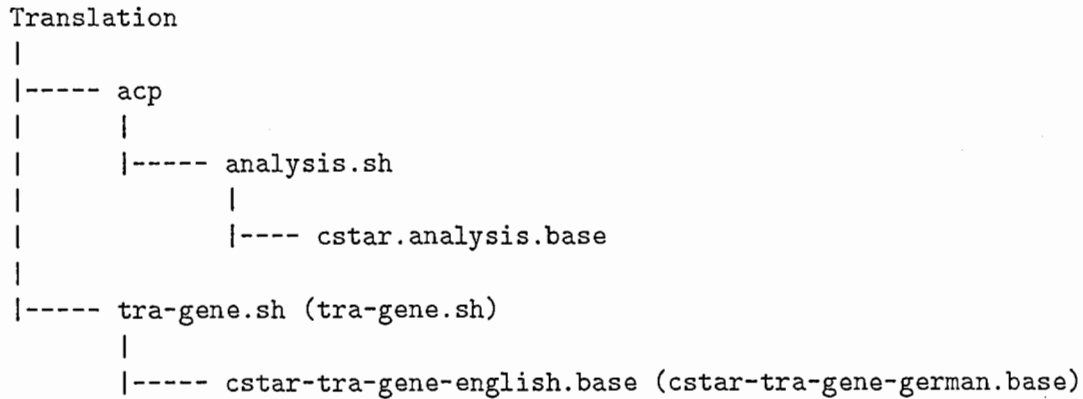
を2倍に拡大した。

- フォントを変更する場合は サーバー側ソースプログラムの font\_data.h の中で宣言している  
フォント名を変更する。

## 3.5 言語翻訳

### 3.5.1 翻訳部各プロセスの概要

翻訳を実行するプロセスの概要を説明する。



( ) 内はドイツ語翻訳時のプロセス

図 3.10: 翻訳部各プロセス

- Translation: 翻訳部の各プロセスをコントロールするメインプロセス。デモシステム起動中は、常時動作している。プロセス起動後、`acp` と `tra-gene.sh` を起動し、ソケットを用いてコミュニケーションを確立する。デモ実行中は、ポストフィルタが出力する認識結果を受け取り、翻訳部の各プロセスと通信しながら翻訳を実行する。
- `acp`: 構文解析部をコントロールするプロセス。デモシステム起動中は常時動作。Translation からの音声認識候補をソケットで読み取り、`analysis.sh` を起動して解析を実行する。`analysis.sh` が終了すると、ソケットを利用して終了メッセージ (解析成功・失敗の情報のみ) を Translation に渡す。
- `analysis.sh`: 解析部の実体である `cstar.analysis.base` (Lisp の実行可能イメージ) を起動するシェル。解析を実行するたびに起動される。認識候補は引数として渡される。Lisp への認識候補の伝達には、シェルのリダイレクト機能とコマンド置換機能を利用している。解析部と変換部のインタフェースのために、出力はリダイレクトされている。なお、このプロセスは本質的な作業を一切行っていない。Lisp 側を工夫すれば (disksave 時に `restart-function` を指定したり、`lisp-init.lisp` を利用するなどして)、省略可能なプロセスである。ただし、開発途中には、Lisp の実行イメージを単独で起動したり、デバッグ用の情報を出力させる必要があるため、このシェルを使うことにも意味はある。
- `cstar.analysis.base`: 構文解析部の実行イメージ。`analysis.sh` に書かれた S 式を評価することにより解析を行なう。入力となる文字列は、`analysis.sh` に変数として記述されており、シェルの変数置換の機能により、具体的な文字列に置き替わる。出力である日本語の意味表現は標準出力に出力されるが、`analysis.sh` がリダイレクトされてるた

めに、結果としてファイルに出力される。解析出力は一般に複数出力される (曖昧性がある) ので、出力は意味表現のリストとなっている。

- tra-gene.sh: 変換・生成部の実体である cstar-tra-gene-english.base を起動するシェル。デモシステム起動中常時動作する。このプロセスも analysis.sh と同じく、省略可能なプロセスである。
- cstar-tra-gene-english.base (cstar-tra-gene-german.base): 変換・生成部の実行イメージ。Translation から構文解析成功のメッセージを受け取ると、ファイルから入力となる日本語の意味表現を読み取り、変換・生成を行ない、出力である英語 (ドイツ語) の文字列をソケットを利用して Translation に渡す。入力である意味表現は複数あるが、現在は最初の候補しかみていない。出力である英語 (ドイツ語) も複数解存在するが、生成内部の機構により、一つの解が選択され Translation に渡される。

### 3.5.2 翻訳部各プロセスの実行ファイルの所在地

すべてのファイルは atrp20 上にある。また実験は atrp19 上で行なったため、atrp19 にも同一のファイルが同じディレクトリに存在する。

- Translation  
:/users/project/asura/demo/cstar/eval/translation/Translation
- acp  
:/users/project/asura/demo/cstar/english/com-interface/acp
- analysis.sh  
:/users/project/asura/demo/cstar/english/com-interface/analysis.sh
- cstar.analysis.base  
:/users/project/asura/demo/cstar/english/world/cstar.analysis.base
- tra-gene.sh (English)  
:/users/project/asura/demo/cstar/english/tra-gene.sh
- tra-gene.sh (German)  
:/users/project/asura/demo/cstar/german/tra-gene.sh
- cstar-tra-gene-english.base  
:/users/project/asura/demo/cstar/english/world/cstar-tra-gene-english.base
- cstar-tra-gene-german.base  
:/users/project/asura/demo/cstar/german/world/cstar-tra-gene-german.base

### 3.5.3 翻訳部各プロセスのプロセス間通信

Translation と acp 間はサービス名 cstar-acp を利用して通信を行なっている。Translation 側からは音声認識候補の文字列を acp に渡し、acp 側からは解析終了のメッセージを Translation に渡している。解析結果自体は一時ファイルを利用している (ソケットを利用して単なる文字列として扱うには大きすぎるため)。

Translation と cstar-tra-gene-english.base 間は サービス名 cstar-nlp を利用して通信を行なっている。Translation 側からは解析成功のメッセージを cstar-tra-gene-english.base に渡している。cstar-tra-gene-english.base からは生成された英語文字列を Translation に渡している。

ドイツ語版ではサービス名が異なるだけで機能は英語と同様である。

[プロセス名]	[サービス名(ソケット番号)]	[プロセス名]
Translation	<----- <cstar-acp> ----->	acp
Translation	<----- <cstar-nlp> ----->	cstar-tra-gene-english.base
Translation	<----- <cstar-nlp2> ----->	cstar-tra-gene-german.base
Translation	<----- 10001 ----->	<display control program>

図 3.11: 翻訳部各プロセスのプロセス間通信

### 3.5.4 データの流れ

音声認識部からの認識候補が英語(ドイツ語)に翻訳されるまでのデータの流れを説明する。

音声認識候補は標準出力に書き出され、UNIXのパイプ機構により Translation プログラムに読み込まれる。Translation プログラムから acp にはソケットを通じて認識候補を渡している。acp のプログラム内で、認識候補文字列はシェルプログラムの引数に変換され、子プロセスである analysis.csh を実行する。analysis.csh 内では、引数はシェル変数として扱われ、解析本体である cstar.analysis.base 内の関数の引数となる。

解析結果(日本語の意味構造)は標準出力に出力されるが、analysis.csh (およびその子プロセスである cstar.analysis.base) がリダイレクトされているため、最終的には一時ファイルに出力されることになる。なお一時ファイルは

```
atrp20(atrp19):/users/project/asura/demo/cstar/tmp
```

に TEMP-FILE.ana という名前で作成される。

解析終了のメッセージがソケットを通じて Translation に送られると、Translation は cstar-tra-gene-english.base に変換・生成を開始させるためのメッセージをソケットで送る。cstar-tra-gene-english.base は日本語意味構造を一時ファイル(TEMP-FILE.ana)から読み取り、変換・生成を行なった後、出力である英語の文字列をソケットを介して Translation に送る。

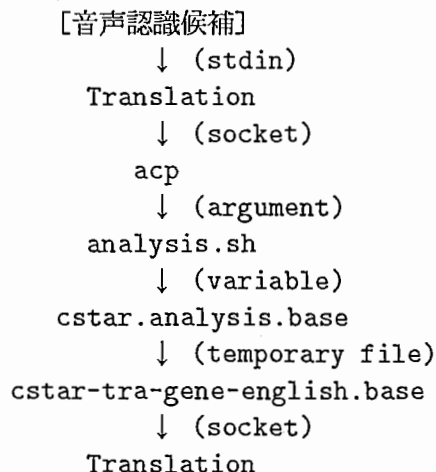


図 3.12: 翻訳部におけるデータの流れ

### 3.5.5 ドイツ語のキャラクターセットについて

ドイツ語にはウムラウト文字などの非アスキー文字が含まれている。これらの文字を表すためのコードについて説明する。

#### 1. 非アスキー文字の種類

ドイツ語を表示するために扱うべき非標準アスキー文字は次の7つである。

Ä Ö Ü ä ö ü ß

ドイツ語ではこれら非アスキー文字を2つのアスキー文字で表す記法(たとえば、äを“ae”で表す)が存在しているが、この記法は用いない。これは、ドイツ語音声合成装置などがウムラウトの一文字表記を前提としていることと、2文字表記を一文字表記に戻す手続きが単純でないことによる。

#### 2. ASURA での扱い

ASURA ではこれらの文字を目的に応じて次の3種類のコード体系で扱っている。

##### (a) 2バイトコード

ATR で作成したコード体系。“.”をエスケープ文字としてこれに後続する文字を特殊文字と解釈する。ウムラウト系の表示できない端末で利用でき、一文字コードへもこの2バイトのみの情報で復元できる。文法、辞書の開発用に用いている。ドイツ語への翻訳を行なうリスププロセス内ではこのコードを用いている。

##### (b) 7ビットコード (DEC-Multinational-Kode fuer das VT220-Sichtgeraet)

[,]などの記号をドイツ語特殊文字に割り当てている。したがって、これらの記号は使えない。音声合成装置、国際通信に使っている。

- (c) 8ビットコード (DEC-Multinational-Kode fuer das VT100-Sichtgeraet, or ISO Draft International Standard 8859/1<sup>3</sup>)

JISでいうと半角カナ文字の所にドイツ語等の特殊文字が入る。デモシステムの画面表示に使っている体系。

ドイツ語への翻訳処理を行なうリスププロセス (cstar-tra-gene-german.base) はこのコードをソケット (ストリーム) に出力する。

最後に、文字ごとのコードを表 3.1に、プロセスとコードの関係を図 3.13に示す。

表 3.1: ドイツ語非標準アスキー文字のコード

非アスキー文字	2 バイトコード	7ビットコード (アスキー文字)	8ビットコード
Ä	:A	5B (I)	C4
Ö	:O	5C (\)	D6
Ü	:U	5D (J)	DC
ä	:a	7B ({)	E4
ö	:o	7C (l)	F6
ü	:u	7D (j)	FC
ß	:s	7E (O)	DF

```

cstar-tra-gene-german.base
  (2bite)
    -----> translation
socket
  (8bit)
      -----> display
      (8bit)
      -----> communication
      (7bit)
      ( -----> Benz-Talk      )
      ( (7bit)                  )

```

図 3.13: 翻訳処理プロセスと文字コードの関係

<sup>3</sup>Latin-1 (?)

## 3.5.6 翻訳結果表示窓

## 仕様

言語翻訳結果データを標準入力で受け取るクライアントプロセスと、クライアントから受け取った翻訳結果データをウインドウに表示するサーバープロセスとに分けられている(図 3.14参照)。万一、ウインドウの幅を越えるような結果が入ってきた時のために、横スクロール機能を備えている。マウス操作は、次の通りである。

- マウスクリック方法
  1. 中ボタン 右へスクロール
  2. 右ボタン 左へスクロール

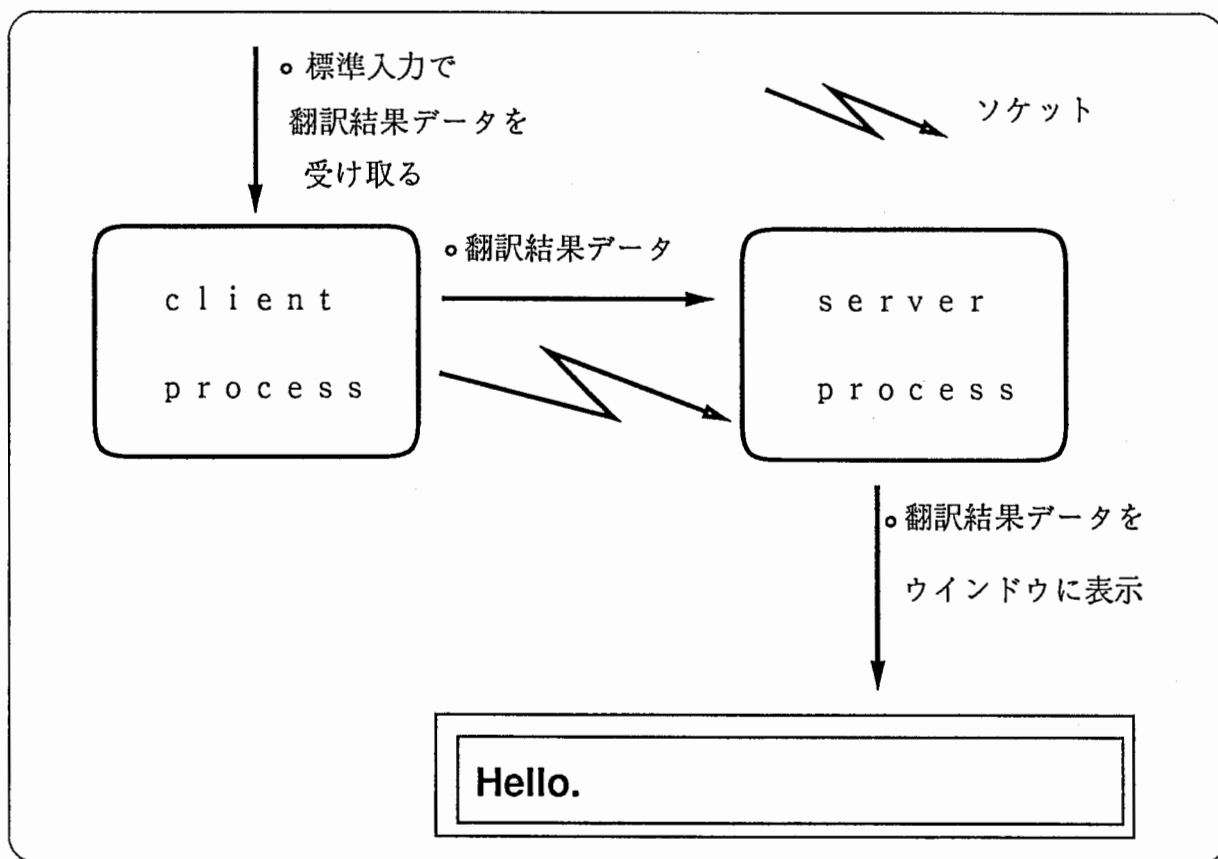


図 3.14: 翻訳結果表示窓プロセスのしくみ

## 標準入力から入力されるデータ例

## 1. 【Siemens 相手の時】

```
#g 0 1
## 0 1
#g -2 1 ... Analyse 1. Kandidat ...
```

```
#g -1 1 Das Thema dieser Konferenz ist Telefondolmetschen.
#g 0 1 Das Thema dieser Konferenz ist Telefondolmetschen.
## 0 1
```

## 2. 【CMU 相手の時】

```
#e 0 1
## 0 1
#e -2 1 ... analysis 1. candidate ...
#e -1 1 Can I still attend?
#e 0 1 Can I still attend?
## 0 1
```

### ソースファイルの所在地

- クライアント → /home/cstar/DEMO\_R1.0/RECOG/SRC/XEXTRACT2/Ver1.2.930205/TRANS\_PARENT
- サーバ → /home/cstar/DEMO\_R1.0/RECOG/SRC/XEXTRACT2/Ver1.2.930205/TRANS

### コンパイル

- HP 仕様: make install で ../../../../../../BINHP の中に実行ファイルができる。
- SUN 仕様: make -f makefile.sun install で ../../../../../../BINSUN の中に実行ファイルができる。

### 実行方法

“標準入力から入力されるデータ例”で述べたようなデータをパイプを通じて標準入力を受け取り、実行させる。

```
client_process_name      : クライアント実行ファイル名
[ -t <titlebar> ]       : ウィンドウタイトル名
[ -display <display_name:0.0> ] : 表示するマシン名
[ -fg <foreground color> ] : フォアグラウンド名
[ -bg <background color> ] : バックグラウンド名
port_number              : ソケットポート番号
hostname                 : 起動するマシン名
server_process_name     : サーバ実行ファイル名
```

### 使用しているフォント

#### ・アスキー:

```
"-adobe-helvetica-bold-o-normal--8-80-75-75-p-50-iso8859-1"
"-adobe-helvetica-bold-r-normal--10-100-75-75-p-60-iso8859-1"
"-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1"
"-adobe-helvetica-bold-r-normal--14-140-75-75-p-82-iso8859-1"
"helvR36"
"helvR48"
```

- フォントはウィンドウの高さによって変わるように設定している。
- なお "tbgm36k", "tbgm48k", "helvR36", "helvR48" は、大画面表示のために特別に作成した。
- "tbgm36k", "tbgm48k", "helvR36", "helvR48" は、それぞれ



```
"-typebank-gothic-medium-r-normal--18-180-75-75-c-180-jisx0208.1983-0"  
"-typebank-gothic-medium-r-normal--24-240-75-75-c-240-jisx0208.1983-0"  
"-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1"  
"-adobe-helvetica-medium-r-normal--24-240-75-75-p-130-iso8859-1"
```

を2倍に拡大した。

- フォントを変更する場合は サーバー側ソースプログラムの font\_data.h の中で宣言している  
フォント名を変更する。

## 3.6 通信制御

ここでは ASURA システムと、CMU, SIEMENS の各システムとの接続時に使用したプログラムについて説明します。

### 3.6.1 公衆電話回線モデム接続

#### 1. 機能

今回、モデムの接続は CMU との接続においてのみ使用しました。モデムの制御に使用したプログラムは“modem\_janus”です。これは、“talk\_modem”プログラム (CMU の Arthur E. McNair 氏作成) をもとにして、ASURA 用に改造したものです。これには下記の機能があります。

- (a) asura\_modem\_sub とのソケットの接続。
- (b) asura\_modem\_sub からの指令により “initfile” を読み込み (AT コマンドの並び) モデムの初期設定を行なう。この時、モデムとの接続に使用するデバイスをオープンする (デバイスはオプションにより指定する)。
- (c) (b) と同じ指令により、相手との接続を行なう。発呼側の場合、こちらから電話をかける。着呼側の場合相手の呼び出しの待ちに入る。
- (d) 接続失敗時リトライする。
- (e) 接続後、相手とのデータの受渡しを行なう。これはプログラム上では、ファイルの入出力を行なうだけです。
- (f) asura\_modem\_sub からの指令によりモデムとの切断を行なう。この時、モデムのリセットも行なう。

#### 2. ソースファイルの所在地

- /home/cstar/DEMO\_R1.0/COMMU/MODEM

#### 3. コンパイル

- “make modem\_janus” で make されるので “cp modem\_janus ../EXE/.” でインストールする。

#### 4. オプション指定

```
modem_janus init ファイル名 [-d number] [-h hostname] [-p port number] [-a]
```

- init ファイル名 モデムの初期化コマンド (AT コマンド)。
- [-d] 相手の電話番号を指定します。
- [-h] asura\_modem\_sub の動作しているマシン名を指定します (今回は “as22” を指定します)。
- [-p] モデムのポート番号を指定します (as22 では “/dev/ttyb” を指定します)。

- [-a] ASURA モードを指定します。modem\_janus は「疑似 JANUS」でも使っていますので、ASURA で使用する場合、このオプションは必須です。

#### 5. 今回使用した init ファイル (pset6y.init)

```
# This file should contain two strings per line.
# The first string is the initialization string that will be sent to the modem.
# The second string is a description of what the first string does.
# By the end of the initializations in this file, the mode must be set to
# no echo of input commands, and a numerical response from the modem.
# Minimal should be ATZ, ATE0, and ATV0.
AT&F "Reinitializing Memory (factory-default)"
AT*MM4 "Set to V.22bis only (2400/1200)"
AT*CD0 "Modem will not disconnect due to loss carrier"
ATX=4 "Setting modem dial up enable"
ATE0 "Setting echo to off"
ATV0 "Setting numerical reply mode"
ATS12=0 "Setting escape code guard time 0"
```

(注) AT コマンドの詳細は、モデムの説明書を参照して下さい。また、AT コマンドの設定は谷戸文廣氏によって行なわれました。

### 3.6.2 パケット回線 (DDX-P) 接続

#### 1. 機能

今回、DDX-P の接続は SIEMENS との接続においてのみ使用しました。DDX-P の制御に使用したプログラムは “modem\_x25” です。これは、“talk\_modem” プログラム (CMU の Arthur E. McNair 氏作成) をもとにして、ASURA 用に改造したものです (DDX-P への対応は、株式会社ノヴァの大西雅晴氏によって行なわれました)。これには下記の機能があります。

- asura\_modem\_sub とのソケットの接続。
- asura\_modem\_sub からの指令により DDX-P との接続に使用するデバイスをオープンする (デバイスはオプションにより指定する)。
- (b) と同じ指令により、相手との接続を行なう。発呼側の場合、こちらから電話をかける。着呼側の場合相手の呼び出しの待ちに入る。
- 接続失敗時リトライする。
- 接続後、相手とのデータの受渡しを行なう。これはプログラム上では、ファイルの入出力を行なうだけです。
- asura\_modem\_sub からの指令により DDX-P との切断を行なう。

#### 2. ソースファイルの所在地

- atr-sp:~cstar/x25

#### 3. コンパイル

- “make” で make されます。

## 4. オプション指定

```
modem_x25 [ -d number ] [ -h hostname ] [ -p port number ] [ -a ]
```

- [-d] 相手の電話番号を指定します。
- [-h] asura\_modem\_sub の動作しているマシン名を指定します (今回は “as22” を指定します)。
- [-p] DDX-P のポート番号を指定します (atr-sp では発呼側時 “/dev/tty00”、着呼側時 “/dev/tty01” を指定します)。
- [-a] ASURA モードを指定します。modem\_x25 は「疑似 JANUS」でも使用していますので、ASURA で使用する場合、このオプションは必須です。

## 3.6.3 通信メインコントロールプロセス

## 1. 機能

asura は付録 C に書かれたプロトコルを実際に実現している処理です。音声合成、相手側状態表示画面へは直接ソケットを介して指示を送ります。オプションパネル、翻訳系、モデムコントロールプロセスへはソケットコントロールプロセスを介してコントロールを行ないます (3.2 節『プログラムの構成』を参照して下さい)。これには下記の機能があります。

- (a) 共有メモリによる通信ソケットコントロールプロセスとの接続。
- (b) 音声合成とのソケットの接続。
- (c) 相手側状態表示画面とのソケットの接続。
- (d) オプションパネルからの設定データを取り込む。
- (e) オプションパネルからの指示により回線オープン指示を出す。
- (f) オプションパネルからの指示により翻訳処理開始指示を出す (音声認識処理を起動)。
- (g) 音声認識結果、翻訳結果など翻訳系から送られてくるデータに対応してプロトコルに基づき相手側へデータを送る。
- (h) 相手側へ送ったデータに対しての応答チェックを行なう。
- (i) 相手側のターンになったら、相手から受け取ったデータをチェックし、応答を返す。
- (j) 相手側のデータを音声合成と相手側状態表示画面に送る。
- (k) オプションパネルからの指示により回線クローズ指示を出す。
- (l) オプションパネルからの指示により全プロセスを終了する。

## 2. ソースファイルの所在地

- /home/cstar/DEMO\_R1.0/COMMU/ASURA\_DEMO\_MODEM

## 3. コンパイル

- “make” で make されるので “cp asura ../EXE/.” でインストールする。

#### 4. オプション指定

asura [-t]

- [-t] dectalk 用の RS232C のポート番号を指定します。指定なしの時は “/dev/null” です。

### 3.6.4 ソケットコントロールプロセス

#### 1. 機能入出力のあるソケットの処理と共有メモリとのデータの受渡しを行なっています。これには下記の機能があります。

- (a) 共有メモリによる通信メインコントロールプロセスとの接続。
- (b) モデムコントロールプロセスとのソケットの接続。
- (c) モデムコントロールプロセスと通信メインコントロールプロセスの間のデータの受渡しをソケットと共有メモリを用いて行なう。
- (d) オプションパネルとのソケットの接続。
- (e) オプションパネルと通信メインコントロールプロセスの間のデータの受渡しをソケットと共有メモリを用いて行なう。
- (f) 翻訳系とのソケットの接続。
- (g) 翻訳系と通信メインコントロールプロセスの間のデータの受渡しをソケットと共有メモリを用いて行なう。

#### 2. ソースファイルの所在地

- /home/cstar/DEMO\_R1.0/COMMU/ASURA\_DEMO\_MODEM

#### 3. コンパイル

- “make -f Makefile.asura\_modem\_sub” で make されるので “cp asura ../EXE/.” でインストールする。

#### 4. オプション指定

asura\_modem\_sub [-h]

- [-h] modem\_janus (モデム使用時) または modem\_x25 (DDX-P 使用時) の動作しているマシン名を指定します (今回は “as22” を指定します)。

### 3.7 日本語音声合成

表3.2に、日本語音声合成モジュールの所在、オブジェクト名、参照ファイル名、作成ファイル名を示す。

表 3.2: 日本語音声合成モジュールの構成

項目	名前	コメント
ホスト・マシン名	atrp17	HP9000/730
ホーム・ディレクトリ	/users/cster/C_STAR	
メイン・モジュール	Cstar_loop	
作成ファイル	*.file *.word *.phra *.CP *.WV	入力合成データファイル 単語情報ファイル アクセント句情報ファイル ケプストラムデータファイル 合成音声
参照ファイル	~/WV/*.WV  ~/CHK_list	CSTAR 用会話文の合成音声 合成音声出力時に参照される * <sup>1</sup> 。 CSTAR 用会話文の合成音声のリスト マッチング時に参照される * <sup>2</sup> 。
ソースファイル	~/job_loop/C_star.h ~/job_loop/Cstar_main.c ~/job_loop/gousei.c ~/job_loop/word_file_check.c ~/job_loop/data_job.c ~/job_loop/synthe_job.c ~/job_loop/make_CSTAR	CSTAR 用ヘッダ CSTAR 用メインルーチン ソケット通信ルーチン (read_data) 入力データフォーマット変換ルーチン 入力データフォーマット変換サブルーチン 文字列チェックおよび音声合成ルーチン make files
バックアップ ディレクトリ	atrf2:/Spara/speech/CSTAR	atrp17:/users/cstar/C_STAR 以下のコピー

- ‘~/’ は、ホーム・ディレクトリを示す。
- 入力データフォーマット変換ルーチンは、CMU/Siemens 側入力データ (4.5.1節、4.5.2節) をへボン式ローマ字列 (accent 区切りなし、4.5.5節) に変換する。
- \*<sup>1</sup>,\*<sup>2</sup> 新たに会話文を加える場合は、ATR 用 Talk 合成システム (atrf2:/Spara/speech/ATR\_SYN/\*、文献 [5] 参照) を用いて追加する文の合成音声を作成し、~/WV にコピーする。次にその文を~/CHK\_list に追加登録する。

日本語音声合成モジュールは、

1. ソケット通信ルーチン
2. 文字列チェックルーチン
3. 音声合成ルーチン
4. 合成音声出力ルーチン

から成る。基本機能は、通信制御モジュールから、ステータス・フラグと合成文データを受け取り、日本語合成音声を出力することである。ターン・アラウンド・タイムの縮小のため、CSTAR デモで使用する会話文の合成音声(入力合成文データより ATR  $\mu$ -Talk 合成システムで作成)を用意し、実行時の入力合成文データとマッチングした文の合成音声が出力されるようになっている。用意していない文が入力された場合は、入力データが音声合成ルーチンに渡され、合成音声は生成される。

### 3.7.1 ソケット通信ルーチン

ここでは、CSTAR デモシステム本体から送られてくるステータス・フラグおよび合成用文字列の受け取りを行なう。通信制御モジュールとのインタフェース部分である。ソケット番号は、10005 である。

#### ● ステータス・フラグの種類

No.	意味
1	CMU 側入力による合成および音声出力
2	Siemens 側入力による合成および音声出力
10	合成音声の再出力
99	プログラム終了

#### ● インタフェース・パケットのフォーマット

0	1	4	5	5+length (byte)
ステータスフラグ	length	NULL	合成用文字列	

### 3.7.2 文字列チェックルーチン

ここでは、入力された文字列により作成される合成音声(~/CHK\_list)の中に、含まれているかどうかのチェックを行なう。含まれている場合は、用意されている合成音声のデータ(~/WV/\*.WV)を合成音声出力ルーチン(3.7.4節)に渡す。含まれていない場合は、入力された文字列を音声合成ルーチン(3.7.3節)に渡す。CSTAR デモ用に約 70 文(付録 B に一覧を示す)を用意した。

### 3.7.3 音声合成ルーチン

入力された文字列により、合成音声の作成を行なう。作成した合成音声のデータを合成音声出力ルーチン(3.7.4節)に渡す(ATR  $\mu$ -Talk 合成システムを使用)。

日本語音声合成ルーチンは、

1. 入力ファイル解析プログラム
  - (a) CMU 用入力解析プログラム (情報ファイル作成)
  - (b) Siemens 用入力解析プログラム (情報ファイル作成)
2. 合成音声作成プログラム (ケプストラムデータ作成)
3. LMA フィルタプログラム (合成音声作成)

の三つに、大別される。

#### 1. 入力ファイル解析プログラム

入力されるファイルフォーマット(4.5.1節、4.5.2節)により、1. (a) もしくは1. (b) のプログラムを用い、単語情報ファイル(4.5.3節)およびアクセント句情報ファイル(4.5.4節)を作成する。

#### 2. 合成音声作成プログラム(ケプストラムデータ作成)

1. で作成された二つのファイルから、ケプストラムデータファイル(4 byte float, \*.CP)を作成する。

#### 3. LMA フィルタプログラム(合成音声作成)

LMA フィルタを使い、ケプストラムデータファイルから合成音声ファイル(2 byte short, \*.WV)を作成する。

### 3.7.4 合成音声出力ルーチン

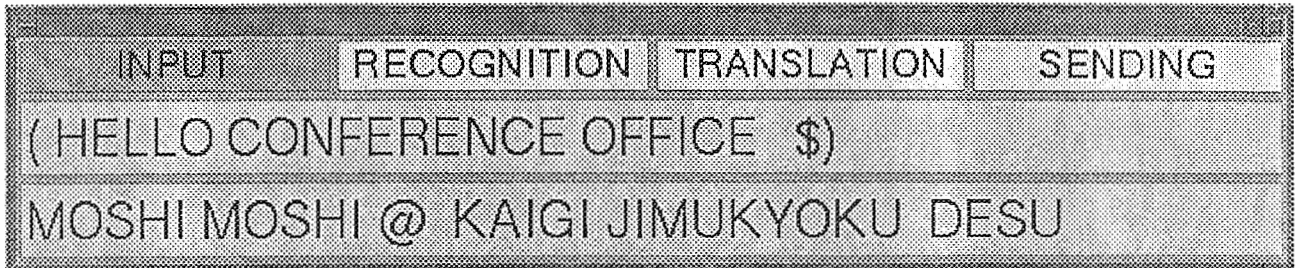
DASBOX に、作成された合成音声(\*.WV)を出力する。



## 3.8 相手側の状態表示窓

## 3.8.1 画面構成の説明

それぞれのウインドウに名称をつける。以下、その名称をもって表すこととする。



総称	名称	位置	役割
ブリンク欄	INPUT	一段目左側	音声入力処理、実行中
	RECOGNITION	一段目中左側	音声認識処理、実行中
	TRANSLATION	一段目中右側	構文解析, 変換生成処理、実行中
	SENDING	一段目右側	処理結果転送処理、実行中
認識, 翻訳結果表示欄	認識結果表示欄	二段目	認識結果の表示
	翻訳結果表示欄	三段目	翻訳結果の表示

ブリンク欄のブリンクとは一般に点滅を意味するが、この画面では実際には点灯しかしない。この画面は以前に作成したデモの流用であり、以前のバージョンは点滅していた。以下、ブリンクという用語を点灯という意味で使っている。

## 3.8.2 機能説明

[概要] このプロセスは、相手側 (CMU/Siemens) からの情報を表示するためのものである。表示する情報には次のようなものがある。

- 音声入力中 [INPUTの点灯]
- 音声認識実行中 [RECOGNITIONの点灯]
- 構文解析実行中 [TRANSLATIONの点灯]
- 変換生成実行中 [TRANSLATIONの点灯]
- 処理結果転送中 (相手側から送られるものではない) [SENDINGの点灯]
- 音声認識結果
- 翻訳結果

[実行時のオプション] 実行時には、以下のオプションが指定できる。ただし、通常はデフォルトの値を持っておりオプションの指定は必須ではない。

```
% plot_window [-fn1 (font name)][-fn2 (font name)][-fnd (font name)]
               [-d (display name)]
               [-s (server host name)]
```

```
-fn1 : 認識結果および翻訳結果表示欄のフォントの指定
-fn2 : ブリンク欄のフォントの指定
-fnd : 表示欄全てのフォントの指定 (-fn1, -fn2 よりも優先)
```

```
-d    : ウィンドウを表示する画面の指定 (format display_name:0.0)
```

```
-s    : ソケットをつなぐサーバープロセスのマシン名
```

また、現在は必要としないが、使用できるオプションが他にもあるのでそれを示す。

```
-p    : 0 か 1 を指定することによりソケットのポート番号を変える。
        0 : 10002
        1 : 10003
```

これは以前に作成したデモの流用であるため、その名残りである。

```
-m    : hp か sun を指定する。hp のマシンと sun のマシンでは、
        ディスプレイの大きさが違うためウィンドウの大きさを、
        それに合わせるためのものである。これは、使用する
        ディスプレイを何にでも対応できるようにしたものである。
```

### 3.8.3 通信について

通信のフォーマットに基づいて、相手側 (CMU/Siemens) からモデムを通して送られてきた情報は、通信制御プロセスからソケットを通して、必要なものだけが相手側状態表示窓プロセスにわたされる。さらに、相手側からの情報に関係なく、内部的な取り決めにより通信制御プロセスからわたされるものもある。以下に、通信制御プロセス、相手側状態表示窓プロセス間で、やり取りされるメッセージの書式を示す。

・メッセージ書式

I n n n \_

I            識別子 (メッセージの種別を示す)  
 n n n        メッセージの文字数 (000-999)  
 \_            ヌル文字 ( \0 ) セパレータ

・メッセージ種別

[ ] の中は画面上の動き

B 0 0 5 \_        音声入力開始 [INPUT のブリンク]  
 N 0 0 5 \_        音声認識実行中 [RECOGNITION のブリンク]  
 K 0 0 5 \_        構文解析実行中 [TRANSLATION のブリンク]  
 T 0 0 5 \_        変換生成実行中 [TRANSLATION のブリンク]

P 0 1 3 \_ B y e \_ B y e \_    音声認識結果文字列 [認識結果の表示]  
 S 0 1 1 \_ H i \_ H i \_        翻訳結果文字列 [翻訳結果の表示]

H 0 0 5 \_        回線の接続終了

e 0 0 5 \_        相手側が English(CMU) のとき  
 g 0 0 5 \_        相手側が German(Siemens) のとき

q 0 0 5 \_        終了 (システムの Quit)

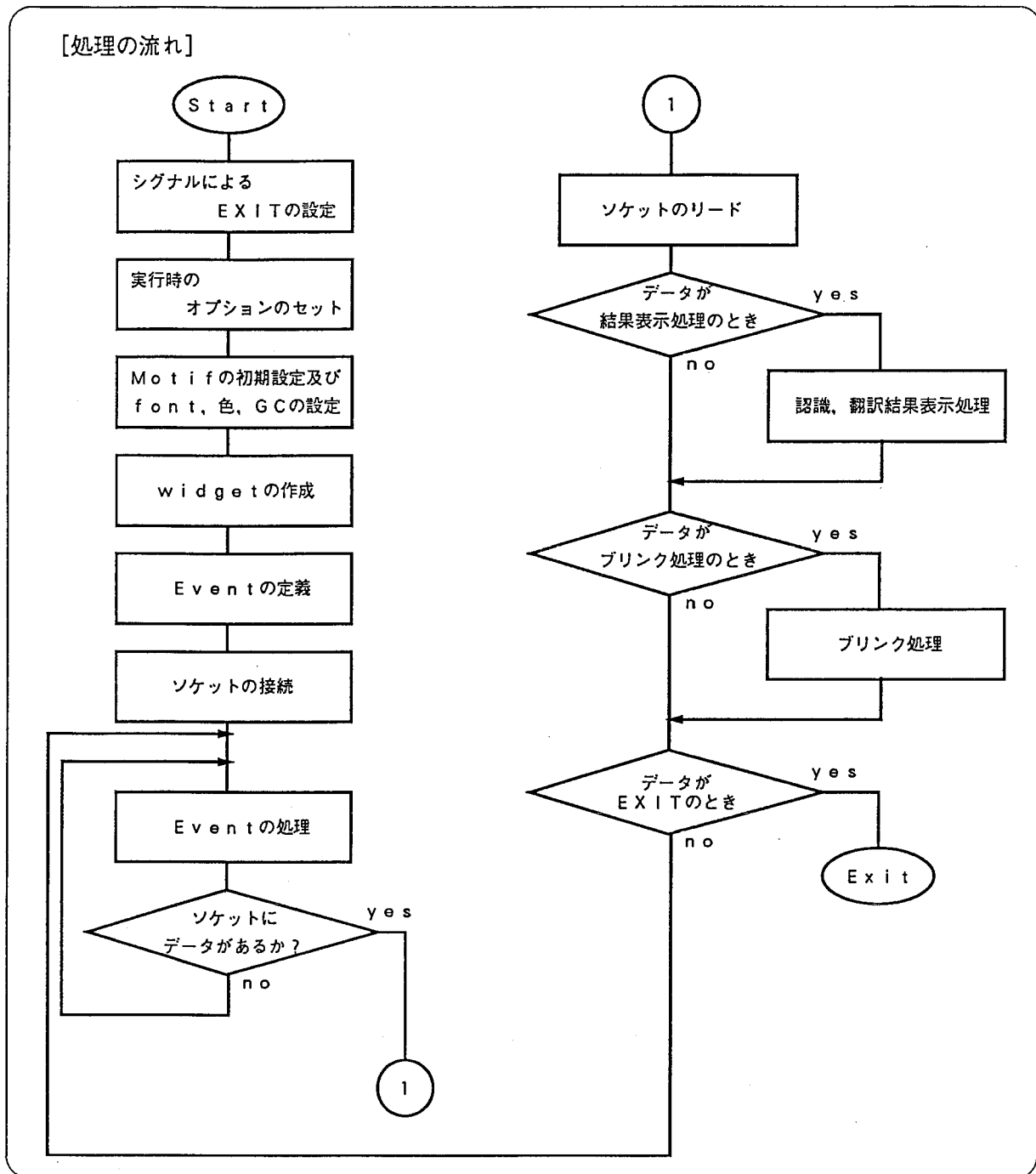
### 3.8.4 CMU/Siemens から受け取る日本語文字列の書式の違い

CMU から送られる音声認識結果文字列と翻訳結果文字列は、普通のローマ字文字列 (ただし @ などの特殊文字がはいっている) なのでそのまま表示する。

Siemens から送られる音声認識結果文字列は普通のローマ字文字列である。しかし、翻訳結果文字列は、音声合成に使用する文字列であるため、特殊な記号や数字などが含まれたローマ字文字列であり (4.5節参照)、表示する前に専用のフィルターを通さなければならない。

また、フィルターを通す通さないの切替え (CMU/Siemens) は、通信制御プロセスから送られる e005 g005 によって行なう。

## 3.8.5 処理説明



[シグナルによる EXIT の設定] ハングアップのとき、または異常時にシグナルを送ることにより EXIT させるためのものである。また、シグナルを受けとった時点でソケットがオープンされているときはソケットにヌル文字を書き出し(そうすることによりソケットをつなぎ直すときにつながりやすくなる)クローズする。対応するシグナルの種類を示す。

- SIGINT (割り込み)
- SIGQUIT (中断)
- SIGKILL (強制終了)

- SIGTERM (ソフトウェア終了)

[実行時のオプションのセット] 実行時のオプションの種類は 3.8.2節 [実行時のオプション] を参照。ここでは、デフォルトの値、オプションの値を変数にセットする。

[Motifの初期設定および、font、色、GCの設定]

1. Motifのデフォルトでもっているオプション (-foreground, -background, -geometry, -display) をプログラムの中から設定する。
2. 使用する色の Color Pixel (unsigned long) を求めて、変数にセットする。
3. フォントをロードする。ロードする際に、エラーとなった場合は、数種類のフォントのロードを試みる。最後に `fixed` がロードできなかった場合はプログラムを終了する。
4. フォントおよび色を GC に設定する。

[widgetの作成]

1. ブリンク欄の widget を作成する。widget の構成はラベルウィジェットをフレームウィジェットで囲んだもの。
2. 認識、翻訳結果表示欄の widget を作成する。widget の構成はドロ잉エリアウィジェットをフレームウィジェットで囲んだもの。

[Eventの設定] 認識、翻訳結果表示欄の Expose と ButtonRelease のイベントを設定。

[ソケットの接続] 通信制御プロセスとの間のソケットの接続を行なう。ソケットがつかない場合、2秒間隔でソケットの接続を30回までリトライする。

<ソケット番号> 10002

[Eventの処理] Motifの XtMainLoop 関数を使わずに、次のような関数によって行なっている。この関数をソケットのリードのときにループの中で呼び出す。

```
void event_manage()
{
    XEvent event;

    XSync(dispatch, 0);
    while(XCheckMaskEvent(dispatch, -1, &event) != 0){
        XtDispatchEvent(&event);
    }
}
```

[ソケットのリード] ソケットのリードは一行単位 (`\n`, `\0` まで) で行なう。ソケットからリードしたデータが通信のフォーマットに合っているかどうかをチェックする。間違ったデータが送られてきたときは、エラーメッセージを `stderr` に出力する。さらに、次のような処理を行なう。

- q005 のとき終了 (EXIT) する
- e005, g005 のとき CMU か Siemens かを区別するフラグを設定する
- H005 のときブリンク欄と認識、翻訳結果表示欄をクリアする

これら以外のデータはバイト数 (005 等) を @ に変えて認識、翻訳結果表示処理とブリンク処理へ、わたされる。

[認識、翻訳結果表示処理] ソケットのリード処理からのデータが次の時、以下の処理を行なう。

- P @ 認識結果文字列のとき、認識結果を表示する。
- S @ 翻訳結果文字列のとき、翻訳結果を表示する。またこのとき、相手側が Siemens である場合は、翻訳結果文字列を専用のフィルターをとうしてから表示する。

さらに、認識、翻訳結果表示処理を行なう関数は表示文字列のスクロール制御も行なう。スクロールは、ウィジェット上で、マウスの真中のボタンクリックで左へ、右のボタンクリックで右へ動く。ボタンイベントの定義の際に、この関数を定義しておく。

さらに、翻訳結果を表示するときに、`SENDING` ブリンクウィジェットを 1/3 秒間点灯させる。1/3 秒のカウントには、`select` 関数を使用する。

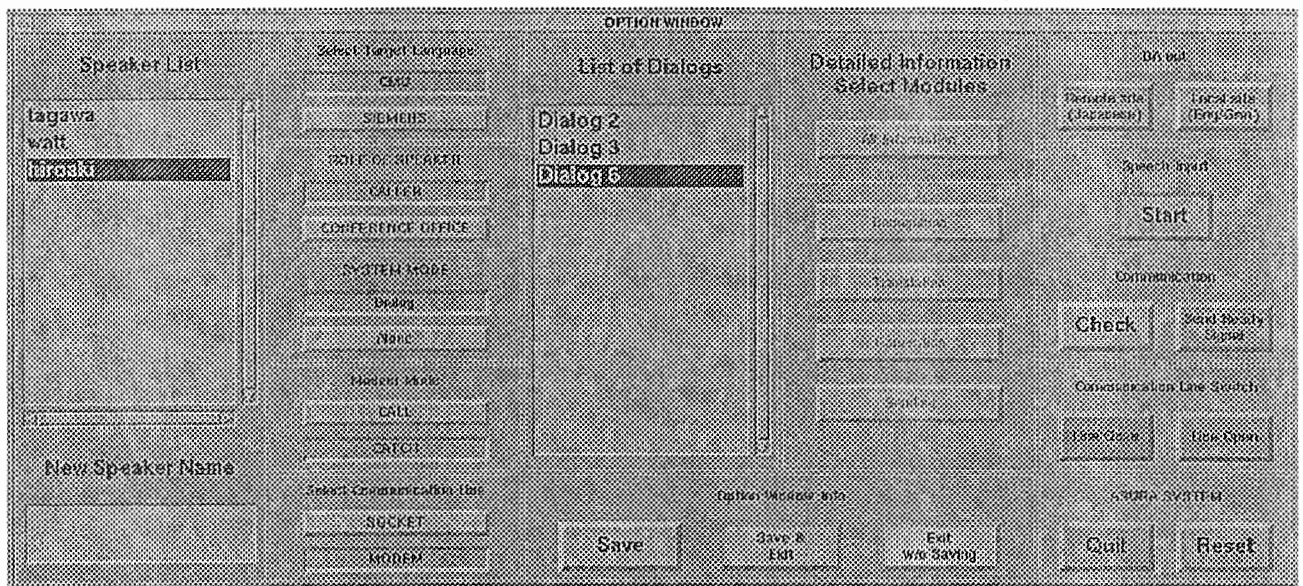
[ブリンク処理] ソケットのリード処理からのデータが次の時、以下の処理を行なう。

- B @ のとき、`INPUT` ブリンクウィジェットのバックグラウンドを点灯させる。
- N @ のとき、`RECOGNITION` ブリンクウィジェットのバックグラウンドを点灯させる。
- K @ のとき、`TRANSLATION` ブリンクウィジェットのバックグラウンドを点灯させる。
- T @ のとき、`TRANSLATION` ブリンクウィジェットのバックグラウンドを点灯させる。
- f @ のときに、`SENDING` ブリンクウィジェットのバックグラウンドが点灯する。  
(これはソケットからのデータではない)

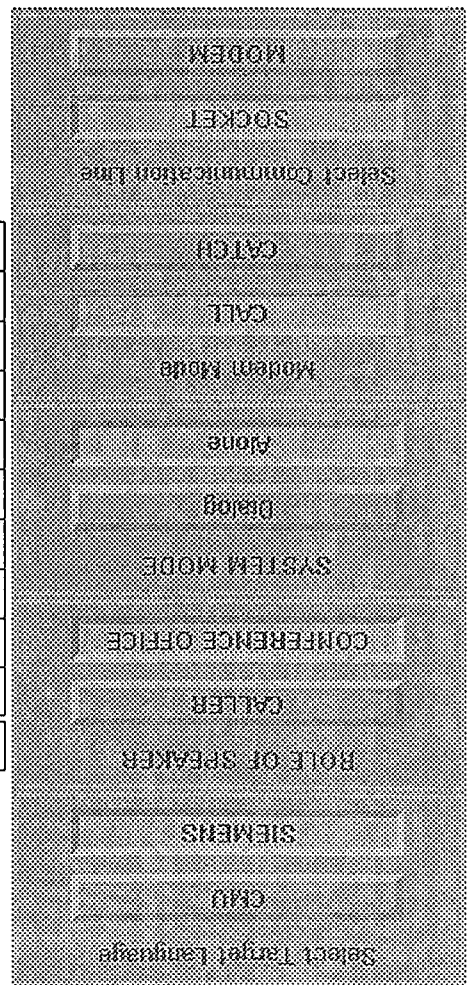
## 3.9 オプション操作画面

## 3.9.1 画面構成の説明

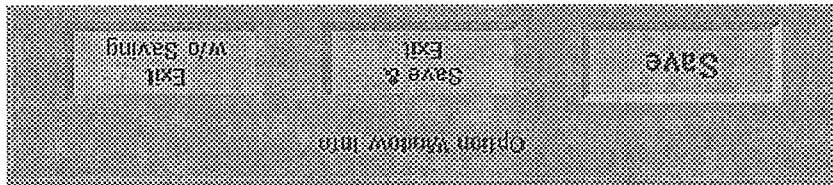
それぞれのボタン、および、リストに名称をつける。以下、その名称をもって表すこととする。



ボタンの名称	ボタンの位置
Speaker List	上図、左端のリストウィジェット
Dialogue List	上図、中央のリストウィジェット

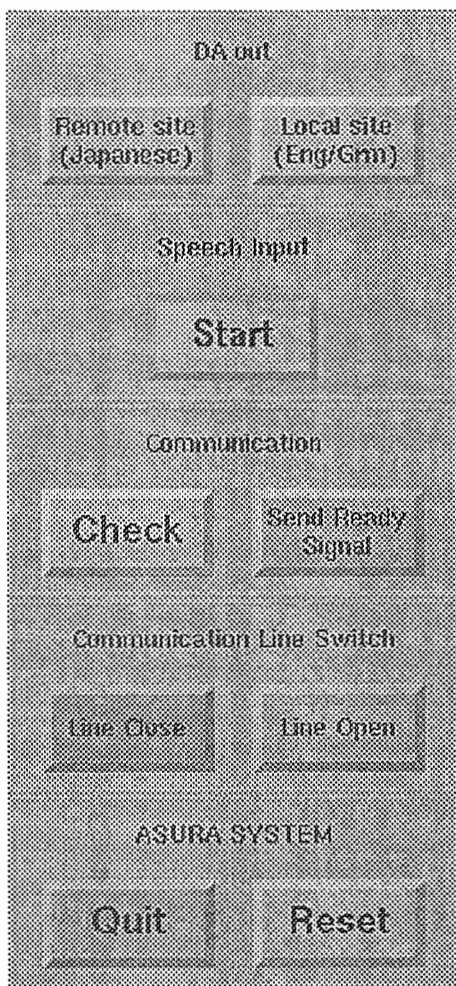


ボタンの名称	ボタンの位置
CMU	1 段目、上段のトグルボタン
SIEMENS	1 段目、下段のトグルボタン
CALLER	2 段目、上段のトグルボタン
OFFICE	2 段目、下段のトグルボタン
Dialog	3 段目、上段のトグルボタン
Alone	3 段目、下段のトグルボタン
CALL	4 段目、上段のトグルボタン
CATCH	4 段目、下段のトグルボタン
SOCKET	5 段目、上段のトグルボタン
MODEM	5 段目、下段のトグルボタン



ボタンの名称	ボタンの位置
Save	左側のボタン
Save&Exit	中央のボタン
Exit	右側のボタン





ボタンの名称	ボタンの位置
DAout Jpn	1 段目、左のボタン
DAout EG	1 段目、右のボタン
Start	2 段目のボタン
Check	3 段目、左のボタン
Send Ready Signal	3 段目、右のボタン
Line Close	4 段目、左のボタン
Line Open	4 段目、右のボタン
Quit	5 段目、左のボタン
Reset	5 段目、右のボタン

### 3.9.2 機能説明

[概要] このプロセスは、通信制御を操作するための画面を表示するものである。画面上のボタンをマウスで操作すると通信メッセージがソケットをとうして通信制御プロセスに送られ、相手側 (CMU/Siemens) との通信が制御される。背景がブルーのボタンおよびリストは **Save** ボタンまたは **Save&Exit** ボタンを押すことにより選択されている情報が送信される。背景がグリーンのボタンはそれを押すだけで情報が送信される。操作する内容には次のようなものがある。

- 背景がブルーのボタンおよびリストで選択され、**Save** ボタンまたは **Save&Exit** ボタンで送信されるもの
  - － 話者名の選択および入力 (使用しない)
  - － 相手側の切替え CMU/Siemens
  - － モデル会話の CALLER/CONFERENCE OFFICE の切替え
  - － システムモードの切り変え Dialog(対話)/Alone(単独) (使用しない)
  - － 相手側との接続をこちらからするか向こうからするか切替え CALL/CATCH
  - － 相手側との接続を SOCKET で行なうか MODEM で行なうかの切替え
  - － モデル会話番号の選択

- 選択したデータのセーブおよび送信
- 背景がグリーンで送信されるもの
  - 相手側の翻訳結果の合成音声の出力
  - こちら側の翻訳結果の合成音声の出力
  - 音声入力開始
  - 接続回線のチェック
  - 相手側にこちらが準備 OK の信号を送る
  - 接続回線のオープン
  - 接続回線のクローズ
  - システムのリセット
  - システムの終了 (Quit)

[実行時のオプション] 実行時には、以下のオプションが指定できる。ただし、通常はデフォルトの値をもっておりオプションの指定は必須ではない。

```
% option_window [-fl (font name)][-ft (font name)]  
                [-d (display name)]  
                [-s (server host name)]
```

- fl : ボタンやラベルに使うフォントの指定
- ft : ボタンやラベルに使う小さなフォントの指定
  
- d : ウィンドウを表示する画面の指定 (format display\_name:0.0)  
実行するマシン以外のディスプレイに表示したい時
  
- s : ソケットをつなぐサーバープロセスのマシン名の指定  
実行するマシン以外でサーバープロセスが動いている時

## 3.9.3 通信について

通信制御プロセスとソケットを通してやり取りされる通信メッセージの種別を以下に示す。

・メッセージ書式

o p t i @ 頭文字列、これによりオプション操作画面からの通信メッセージであることを示す。  
 I n n n I (識別子)、n n n (バイト数)  
 \_ n l 文字セパレータ

<注意>

D@、A@の2つは通信制御プロセスでは受けていない。オプション操作画面プロセスからの送り捨てとなっている。

● オプション操作画面プロセスから通信制御プロセスへ送る通信メッセージ

○  ボタンまたは、 ボタンにより送信されるもの

メッセージ	対応するボタン	メッセージの内容
opti@USERS\$????_	Speakers List	選択されている話者を指す(???? は話者名)
M008_??_	Dialogue List	モデル会話のダイアログ番号を表す(?? はダイアログ番号)

(トグルボタンの選択によりどちらかが送信されるもの)

opti@CMU_	CMU	通信の相手がCMUであることを表す
opti@SIEMENS_	SIEMENS	通信の相手がS i e m e n sであることを表す
opti@CALL_	CALL	モデムのつなぎ方はこちらが発呼側であることを表す
opti@CATCH_	CATCH	モデムのつなぎ方はこちらが着呼側であることを表す
L005_	CALLER	モデル会話における役割がC a l l e rであることを表す
O005_	OFFICE	モデル会話における役割がO f f i c eであることを表す
D@_	Dialog	システムの動作モードがD i a l o g (対話)形式であることを表す
A@_	Alone	システムの動作モードがA l o n e (単独)形式であることを表す

○ 背景がグリーンのボタンにより送信されるもの

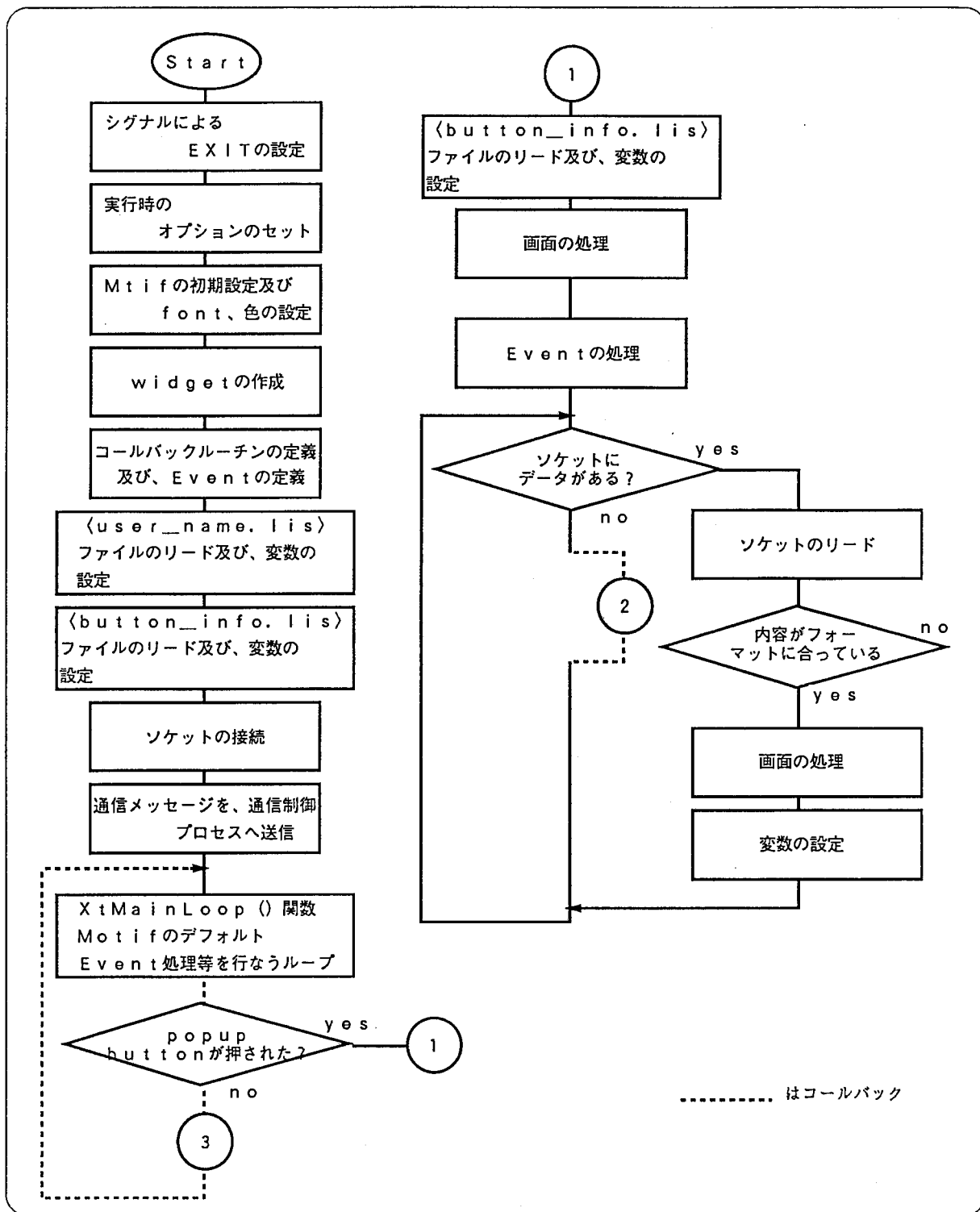
メッセージ	対応するボタン	メッセージの内容
opti@DAOUT_	DAout Jpn	相手側 (CMU/Siemens) の翻訳結果 (日本語) の合成音声出力
opti@DAOUT_EG_	DAout EG	こちら側の翻訳結果 (英語またはドイツ語) の合成音声出力
opti@HMMSATART_	Start	音声入力の開始
opti@LINECHECK_	Check	接続回線のチェック
opti@READY_	Send Ready Signal	準備OK信号の送信 (相手側に)
opti@LINEON_	Line Close	接続回線のオープン (モデムをつなぐ)
opti@LINEOFF_	Line Open	接続回線のクローズ (モデムを切る)
opti@QUIT_	Quit	ASURAシステムの終了
opti@RESET_	Reset	ASURAシステムのリセット

● 通信制御プロセスからオプション操作画面プロセスへ送られる通信メッセージ

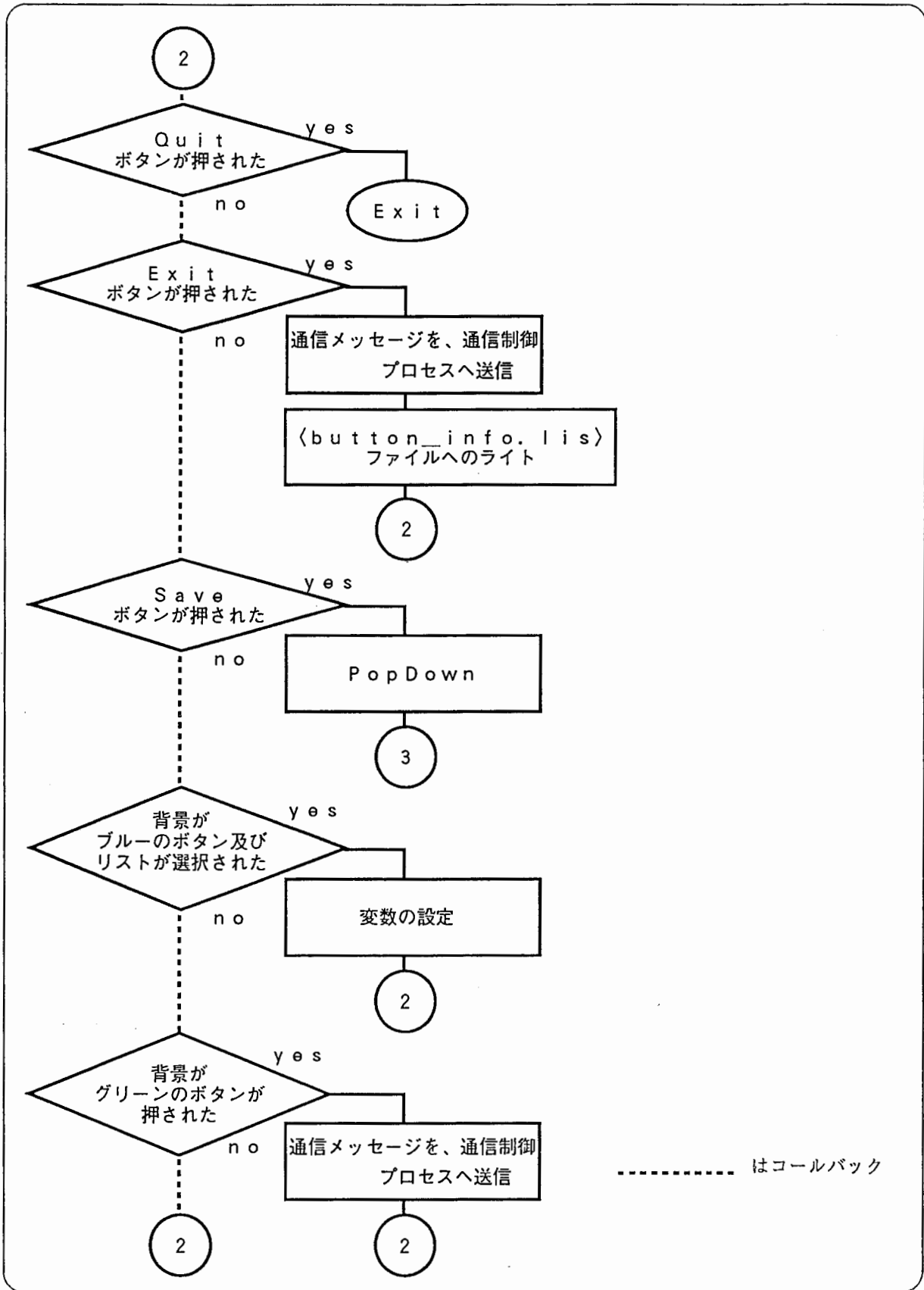
メッセージ	対応するボタン	メッセージの内容
M008_??_	Dialogue List	モデル会話のダイアログ番号を表す (?? はダイアログ番号) <input type="text" value="Dialogue List"/> で選択されているダイアログ番号が送られてきたダイアログ番号にかわる
L005_	CALLER	モデル会話における役割が <i>Caller</i> であることを表す。 <input type="text" value="CALLER"/> ボタンが選択される
O005_	OFFICE	モデル会話における役割が <i>Office</i> であることを表す。 <input type="text" value="OFFICE"/> ボタンが選択される

3.9.4 処理説明

[ 処理の流れ 1 ]



[ 処理の流れ 2 ]



## [シグナルによる EXIT の設定]

- ハングアップのとき、または異常時にシグナルを送ることにより EXIT させるためのものである。また、シグナルを受けとった時点でソケットがオープンされているときはソケットにヌル文字を書き出し(そうすることによりソケットをつなぎ直すときにつながりやすくなる)クローズする。対応するシグナルの種類を示す。
  - SIGINT (割り込み)
  - SIGQUIT (中断)
  - SIGKILL (強制終了)
  - SIGTERM (ソフトウェア終了)

## [実行時のオプションのセット]

- 実行時のオプションの種類は 3.9.2 節 [実行時のオプション] を参照。ここでは、デフォルトの値、オプションの値を変数にセットする。

## [Motif の初期設定および、font、色の設定]

1. Motif のデフォルトでもっているオプション (-foreground, -background, -geometry, -display) をプログラムの中から設定する。
2. 使用する色の Color Pixel (unsigned long) を求めて、変数にセットする。
3. フォントをロードする。ロードする際に、エラーとなった場合は、数種類のフォントのロードを試みる。最後に `fixed` がロードできなかった場合はプログラムを終了する。

## [WIDGET の作成]

1. ポップアップボタンを作成する。widget の構成はフォームウィジェットにプッシュボタンをのせたもの。
2. ポップアップウィンドウを作成する (ポップアップボタンを押すとポップアップするウィンドウのこと)。widget の構成はフォームウィジェット。このウィンドウの上にこれから説明するウィンドウ、および、ボタン、リスト等をのせる。
3. 背景がグリーンボタンの作成および背景がブルーのプッシュボタンの作成。
  - (a) ボタンをのせるフォームウィジェットを作成する。
  - (b) ボタンのタイトルを書くためのラベルウィジェットを作成する。
  - (c) ボタンを作成する (プッシュボタン)。
4. 背景がブルーのトグルボタンの作成
  - (a) トグルボタンをのせるフォームウィジェットを作成する。
  - (b) トグルボタンのタイトルを書くためのラベルウィジェットを作成する。

- (c) トグルボタンを作成する。ただし、このトグルボタンは Motif の default のトグルボタンではない。プッシュボタンをプログラムでコントロールすることによりトグルボタンのように見せかけたものである。

〈フォームウィジェット、ラベルウィジェット、プッシュボタン、トグルボタンの作成に関する注意〉  
 フォーム、ラベル、プッシュボタン、トグルボタンを作成する関数はそれぞれひとつある。作成する時は、その関数にウィジェットの情報が入った構造体を引数として渡す。その構造体の書式を次に示す。

```
typedef struct{
  XFontStruct *font;
  char *label;          /* label string */
  Widget *wd;          /* widget (pointer) */
  Widget *p_wd;        /* parent widget (pointer) */
  char *wd_name;       /* widget name */
  int sensitive;      /* use or unused (set True or False) */
  int x;
  int y;
  int width;
  int height;
  char *fore;          /* foreground color (RGB "#xxxxxx") */
  char *back;          /* background color */
  char *top;           /* top shadow color */
  char *bottom;        /* bottom shadow color */
  char *arm;           /* button arm color */
} Struct_Arg;
```

5. module information window を作成する。CSTAR デモでは、sensitive に設定してあり、実際には利用できない。
6. **Dialog List** を作成する。作成する時にリストの内容を〈model\_talk.lis〉ファイルから読み込む。また、リストの内容を変更する時は、〈model\_talk.lis〉ファイルを書き換えればよい。
7. **Speaker List** を作成する。

#### [コールバックルーチンの定義および、Event の定義]

- 各ボタンが押された時、および、リストが選択されたときに呼び出されるコールバックルーチンを定義する。
- ポップアップボタンが他のウインドウの下に隠れないようにする Event の定義をする。

#### [〈user\_name.lis〉ファイルのリード、および、変数の設定]

- 〈user\_name.lis〉ファイルを読み込みユーザー名を配列に格納する。この配列の内容から、**Speaker List** の内容が設定される。また、〈user\_name.lis〉ファイルの内容を書き換えることにより、立ち上げ時の **Speaker List** の内容を変更できる。  
 〈user\_name.lis〉ファイルの書式を次に示す。

```
tagawa
watt
hiroaki
```



## [ &lt;button\_info.lis&gt; ファイルのリード、および、変数の設定]

- <button\_info.lis> ファイルを読み込む。<button\_info.lis> ファイルには、各ボタン、および、リストの情報が書き込まれている。その情報をプログラム内部の変数にセットする。また、<button\_info.lis> ファイルの内容を書き換えることにより立ち上げ時の設定を変更できる。<button\_info.lis> ファイルの書式を次に示す。

```
USER : tagawa
SYSTEM : CMU
SPEAKER : CALLER
MODE : DIALOG
INFORMATION : NONE
MODEL TALK : 6
LINE : SOCKET
MODEM : CALL
```

## [ソケットの接続]

- 通信制御プロセスとの間のソケットの接続を行なう。ソケットがつながらない場合、2秒間隔でソケットの接続を30回までリトライする。

```
<ソケット番号>    10004
```

## [通信メッセージを通信制御プロセスへ送信する]

- 立ち上げ時の設定を通信メッセージの書式にしたがって送信する。(  ボタンを押した時と同じ処理の内容)

## [ポップアップボタンが押されたとき]

- ポップアップボタンが押されるまで(ポップダウンしている時)の間は、Motifのもっている XtMainLoop 関数によって Event の処理などを自動的に行なっている。
- ポップアップボタンが押された時(ポップアップしている時)は、Event の処理とソケットリードのループに入る。

## [Event の処理]

- Motif の XtMainLoop 関数を使わずに、次のような関数によって行なっている。この関数をソケットのリードのときにループの中で呼び出す。

```
void event_manage()
{
    XEvent event;

    XSync(dispatch, 0);
    while(XCheckMaskEvent(dispatch, -1, &event) != 0){
        XtDispatchEvent(&event);
    }
}
```

#### [画面の処理]

- 各ボタン、および、リストの設定値が格納されている変数の値によって、画面上 (見ための) の設定を変更する。

#### [〈button\_info.lis〉ファイルへのライト]

- **Save** ボタン、および、**Save&Exit** ボタンが押された時、その時点での各ボタンの設定値を、〈button\_info.lis〉ファイルへセーブする。よって、プロセスを終了した時点での〈button\_info.lis〉ファイルの内容は、最後に **Save** **Save&Exit** ボタンが押されたときの設定値となっている。

### 3.10 システムパラメータの初期設定と変更

ここでは、システム起動前の初期設定、および、それら設定の変更方法を記す。

#### 1. 変更すべきファイルについての説明。

- ファイル名: SSS\_C-STAR.csh.options
- ディレクトリ: atrp11:/users/cstar/DEMO\_R1.0/RECOG/Exe\_C-STAR
- 使用目的: このシステムのシェルスクリプト内で使用している環境変数をこのファイル内で起動前の設定を行なう。

#### 2. このファイル内で設定されている環境変数の説明。

- # connect x server
  - X\_HOST  
DISPLAY  
ASURA 画面の表示場所の設定。
- # honyaku server
  - HONYAKUHOST  
翻訳システムコントロールプロセスを起動するマシンを設定。
- # communicate server (Modem)
  - COMMU\_HOST  
翻訳システムコントロールプロセスが通信する相手。  
動作状態 (状態認知メッセージ) を送受信する通信制御プロセスが起動しているマシン設定。
- # SSS RECOG\_TRANS start shell script
  - RECO\_PROC\_CSH  
SSS-LR および翻訳システムの起動を行なう。システム立ち上げのもととなるシェルスクリプトのファイル名の設定。
- # socket port base number
  - SSS\_PORT  
SSS-LR 内のプロセス群の接続に使用しているソケットのポート番号で最小のもの。
- # SSS options
  - RECOG\_TRANS  
翻訳処理まで行なうか、認識処理まで行なうか。
  - HARDSOFT  
音声解析をソフトバージョンで起動するか、ハードバージョンで起動するか。
  - HMnet\_Size  
HMnet の全状態数。

- GRAMMAR  
文節文法。
- BUN\_GRA  
文文法。
- GLOBAL\_BEAM  
パーズング中に生成される解析木の各深さにおいて残すべき仮説数。
- MARGIN  
波形データの前後に入れる無音領域の時間 [単位: msec]。

### 3. 環境変数変更方法または、変更可能変数の説明と操作方法

これより以下に変更可能な変数とその変更方法を記す。

- 画面表示ウィンドウ群の表示マシンの変更  

```
setenv X_HOST [マシン名]
```

```
setenv DISPLAY [マシン名:0.0]
```

 ここでのマシン名は常に同じものであること
- 音声認識のみ起動する  

```
setenv RECOG_TRANS TRANSLATION
```

 ↓  

```
setenv RECOG_TRANS RECOGNITION
```

 このとき、通信制御との接続は行なわない。よって、CSTAR 版 ASURA システム起動時には、この変数は変更しないこと。この状態で起動させたい時は、  

```
> cd /users/cstar/DEMO_R1.0/RECOG
```

 をしてから  

```
> /Exe_C-STAR/SSS_C-STAR.csh [-ENGLISH or -GERMAN]
```

 をすると話者メニューが表示される。そこで話者を選択すると SSS-LR (認識のみ) が起動する。
- 音声解析をソフトバージョンで起動する  

```
setenv HARDSOFT HARD
```

 ↓  

```
setenv HARDSOFT SOFT
```

 ハードウェア (IOPBOX) のトラブル等でソフトウェア (SpeechIn ~) をかわりに起動させたい時に変更をする。
- 翻訳システムコントロールプロセスの起動マシンを変える  

```
setenv HONYAKUHOST [マシン名]
```

 これは基本的には初期の設定マシンのみ起動可能である。
- 通信制御プロセスが起動するマシン設定  
 この対象マシンは初期設定マシンのみである。  
 ただし、翻訳システムのみ (通信制御と接続しない) ならばダミーとしてソケット (サーバー側) を接続すれば起動する。  
 その方法としては、まず  

```
setenv COMMU_HOST atrp11
```

として他のターミナルで

```
> cd /users/cstar/DEMO_R1.0/TRANS/local_test_socket  
> local_soc_server
```

を起動して、音声認識のみ起動する手順 (3 で記述した手順) を行なう。  
(ただし RECOG\_TRANS は TRANSLATION のまま)。

- 文法の変更文法はこのシステム専用の文法を使用している。この文法を変更するには

```
setenv GRAMMAR [文節内文法]
```

```
setenv BUN_GRA [文節間文法]
```

を変更すればよい。文法作成についての詳細は 4.1節を参照のこと。

- SSS options のその他の変数について

```
HMnet_Size · GLOBAL_BEAM · MARGIN
```

に関してはこのシステムにおいて最適と思われる値を設定しているので、変更の必要は特にないと思われる。

### 3.11 話者適応

話者適応を行なうためのトップレベルのシェルは“Speaker\_select.sh”と“Adapt.sh”である。“Speaker\_select.sh”からは、さらに、“Speaker\_select0.sh”という別のシェルが呼ばれている。CSTAR 版 ASURA で利用したプログラムは、次の場所にある。

```
atrp11:~cstar/DEMO_R1.0/Adapt/Adapt.sh
      Sp_select/Speaker_select.sh
      Sp_select/Speaker_select0.sh
      HMnet/
      Data/
```

“Speaker\_select.sh”は標準話者の選択のために使用されている。これによって、話者適応用として登録された音声の先頭から3単語を使用して、その音声に最もふさわしい標準話者モデルが選択される。この際に標準話者モデルの候補となるものは、ディレクトリ“HMnet/”の下にある話者のモデルとなる(現時点では、MAU, MHT, MXM, FMS, FTK, FYMの6名が登録されている)。

“Adapt.sh”では、“Speaker\_select.sh”によって選択された標準話者のモデルに対するパラメータの変更が行なわれる。これによって新たに作成された、使用者専用のモデルは、ディレクトリ“Data/「使用者名」/Adapted\_HMnet”の下に保管される。

なお、既に登録されている話者を削除するには、このディレクトリ“Data/”の下に作られた「話者名」のディレクトリを削除すればよい。

## 3.12 ローカルテスト用ツール

### 3.12.1 プログラムの概要

#### 1. ウィンドウプログラム (janus)

これはローカルテスト用ツール (通称: 疑似 JANUS) のメインモジュールで、次の機能を持つ。

- (a) 相手の状態表示
- (b) こちらの状態表示
- (c) オプションパネルでの各種設定 (接続方法の指定、ブリンク時間の指定など)
- (d) 送信シナリオの指定
- (e) シナリオの送信

#### 2. モデムとウィンドウのインタフェースプログラム (modem\_sub)

これは相手との接続処理を行なうプログラムである。ソケットによる接続時には、このプロセスから直接相手と接続を行なう。モデムによる接続時には、モデム通信プログラムとソケットによる接続を行ない、モデムに対してスタート、ストップの指令を行なう。ウィンドウプログラムとの接続は、共有メモリを用いて行なっており、相手からのデータ、こちらからのデータの受渡しを行なっている。

#### 3. モデム通信プログラム (modem\_janus)<sup>4</sup>

これはモデム接続時にのみ利用するプログラムで、ソケット接続時には必要ない。モデム接続時には、モデムとウィンドウのインタフェースプログラムからの指令により、モデムのリセット、相手との接続、データの送受信、交信の切断などの処理を行なう。

### 3.12.2 プログラムで有効なオプション指定

オプション指定によっては、「疑似 ASURA」としても動作可能。

#### 1. ウィンドウプログラム (janus [-a ][-s ][-t ])

- デフォルト ([ -a ][ -s ]) のどちらも指定しないモードでは、ASURA に対する「疑似 JANUS」として動作。
- [-a ] のみオプション指定した時、JANUS に対する「疑似 ASURA」として動作。
- [-s ] のみオプション指定した時、ASURA に対する「疑似 Siemens システム」として動作。
- [-a ][-s ] の両方をオプション指定した時は、Siemens システムに対する「疑似 ASURA」として動作。
- [-t]: DECTALK の出力ポートを指定する (例: /dev/null, /dev/ttya, /dev/ttyb etc.)。

---

<sup>4</sup>このプログラムは ASURA システムでのモデム通信にも利用している。

## 2. モデムとウィンドウのインタフェースプログラム (modem\_sub [-h] [-a])

- [-h]: モデムが接続されているホストマシン名を指定できる。
- [-a]: 「疑似 ASURA」モードとして動作する。

## 3. モデム通信プログラム (modem\_janus\_initializefile [-d] [-h] [-p] [-a])

- [-d]: 相手側の電話番号。
- [-h]: モデムとウィンドウのインタフェースプログラムが動作しているマシン名。
- [-p]: モデムのポート番号 (as22 では “/dev/ttya” を指定)。
- [-a]: ASURA システムでこのプログラムを利用する時に指定するオプション。

## 3.12.3 メインウィンドウ

このツールのメインウィンドウであり、ASURA との回線接続や現在状況を示す役割をするウィンドウである (図 2.10 参照)。上から順に説明する。

## 1. メニューウィンドウ

## (a) “File” のプルダウンメニュー

- i. “Select Scenario”  
選択すると、シナリオセレクションウィンドウが表示される。
- ii. “Close”  
プロセスの終了。

## (b) “Command” のプルダウンメニュー

- i. “Line Check”  
ASURA との回線をチェックする。
- ii. “Connect Line”  
ASURA との回線を接続する。
- iii. “Disconnect Line”  
ASURA との回線を切断する。

## (c) “Option” のプルダウンメニュー

- i. “Option Panel”  
オプションパネルウィンドウが表示される。

## 2. 役割

“CALLER”, “CONFERENCE OFFICE” がある。

## 3. 状態メッセージ

次のものが用意されている。



- (a) “Not Ready” モデム、または、ASURA 側の準備ができていない。
- (b) “Connect” ASURA 側と回線接続完了。
- (c) “Ready” ASURA 側、または、こちら側の準備完了。
- (d) “Input Ready” 入力準備完了。
- (e) “Waiting” 待機中。
- (f) “Processing” 処理中。

#### 4. 状況の表示

現在、どのモジュールが処理されているかをブリンクで示す。

- (a) “INPUT” 音声入力
- (b) “RECOGNITION” 音声認識
- (c) “TRANSLATION” 言語翻訳
- (d) “SENDING” 送信

### 3.12.4 オプションパネルウインドウ

メインウインドウ上端メニュー中の **Option** をクリックして、プルダウンメニュー中の **Option Panel** を選択すると、オプションパネルウインドウが表示される (図 2.11 参照)。

#### 1. POSITION

問合せ者側なのか会議事務局側なのかを設定 (トグルボタンを押すと変更できる)。

- CALLER
- OFFICE

#### 2. PART

クライアント側かサーバ側かを設定 (トグルボタンを押すと変更できる)。

- CLIENT
- SERVER

#### 3. ACCESS

モデムかソケット<sup>5</sup>かどちらを使用するかを設定 (トグルボタンを押すと変更できる)。

- MODEM
- SOCKET

#### 4. MODEM

MODEM の発呼側か着呼側かを設定 (トグルボタンを押すと変更できる)。

- CALLER
- REVEIVER

---

<sup>5</sup>ソケットについては動作は確認していない。

## 5. DELAY TIME

それぞれのモジュールの模倣的な処理時間を設定する(単位は秒)。スケールをクリックし移動させて設定する。を押すと、すべてのモジュールを5秒に設定する。

- |     |               |      |
|-----|---------------|------|
| (a) | “INPUT”       | 音声入力 |
| (b) | “RECOGNITION” | 音声認識 |
| (c) | “TRANSLATION” | 言語翻訳 |
| (d) | “SENDING”     | 送信   |

## 6. BLINK COLOR

ブリンクするカラーを設定(既に登録してあるカラーをクリックするか、キーボードから入力する)。カラーを入力した時、そのマシンで使用できるか判定し、使用不可能だったら警告メッセージを表示する。

## 7. HOSTNAME

相手側のマシンの名前を設定(既に登録してあるホスト名をクリックするか、キーボードから入力する)。

- ボタンを押すと、その時の状態でオプションパネルを閉じる(ただし、シナリオウインドウを開いた状態で、POSITION を変更し、 ボタンを押して終了した場合は、“status change position really” という警告メッセージが表示され、 か  かを要求する)。
- ボタンを押すと、オプションパネルを開く前の状態で終了する。
- このプロセスが終了しても前の状態はファイルに残される(option-data.file)。

## 3.12.5 シナリオセレクトションウインドウ

- シナリオファイルを設定する。
- メインウインドウ、あるいは、シナリオウインドウの  をクリックして、プルダウンメニュー中の  を選択した時に表示される。
- シナリオセレクトションウインドウは、おおよそ次の5つのウインドウで成り立っている(図 2.12参照)。
  1. 現在のフィルタ<sup>6</sup>が表示される編集可能なテキストウインドウ(上辺)
  2. ディレクトリが表示されるリストウインドウ(中上左)
  3. シナリオファイル名が表示されるリストウインドウ(中上右)
  4. 現在のファイル名が表示される編集可能なテキストウインドウ(中下)
  5. OK, FILTER, CANCEL, および, HELP の PushButton ウインドウ(下辺)
- ファイルの選択には2通りの方法がある。1つは、ファイル名リスト中のファイル名を選択する方法である。もう1つは、現在のファイル名が表示されているテキストウインドウに直接キーボードから入力する方法である。

<sup>6</sup>\* や? のようなワイルドカードの指定をするもの。

- 他のディレクトリのファイル名を参照したい場合は、ディレクトリが表示されているリストウインドウで、参照したいディレクトリをダブルクリックすると、ファイル名を表示することができる。

### 3.12.6 シナリオウインドウ

このウインドウは、シナリオセレクションウインドウによって開くことができる。選択されたシナリオを表示している (図 2.13参照)。シナリオセンテンスを選択することができる。上から順に説明する。

#### 1. メニューウインドウ

##### (a) “File” のプルダウンメニュー

###### i. “Select Scenario”

選択すると、シナリオセレクションウインドウが表示される。

###### ii. “Close”

シナリオウインドウは閉じられる。

##### (b) “Command” のプルダウンメニュー

コマンドボタンウインドウの処理と同じ。

#### 2. コマンドボタンウインドウ

##### (a) Send Scenario No.

相手にシナリオ番号を送る。

##### (b) Send Sentence

相手にセンテンスを送る。この機能を使わなくても、シナリオウインドウで送りたいセンテンスをダブルクリックすることにより、相手にそのセンテンスを送ることもできる。

##### (c) Send Over

発話サイトを相手方に切り替える。

##### (d) DA Out

翻訳結果が音声出力される。

#### 3. こちらの役割

ENGLISH-CALLER, ENGLISH-OFFICE がある。

#### 4. シナリオファイル名

(例) kaiwa-D2.caller.eng

#### 5. シナリオ表示ウインドウ

シナリオが表示される。

#### 6. メッセージボックス

こちらの翻訳結果が表示される。

### 3.12.7 メッセージウインドウ

このツールが起動された時に表示されるウインドウである(図 2.14 参照)。上のウインドウはこちら側の、下のウインドウは相手側のそれぞれ認識結果と翻訳結果を表示する。上のウインドウの RECOGNITION の横に付いている正方形の小さいウインドウをクリックすると、こちら側の音声認識結果を音声出力する。下のウインドウの TRANSLATION の横に付いている小さいウインドウをクリックすると、相手側の翻訳結果を音声出力する。

## 第 4 章

### 自動翻訳電話国際共同実験用 ASURA で必要なファイル等

第 4 章は自動翻訳電話国際共同実験用 ASURA の管理者・保守担当者用の手引である。音声認識や言語翻訳の文法・辞書ファイルの差し替えをするような場合はプログラムの内容を詳しく知る必要はなく、第 2 章と第 4 章を読めばその作業ができるはずである。

#### 4.1 音声認識用辞書ファイル

ここではオリジナル文法から、SSS-LR 用文法に変換して、SSS-LR 用の LR テーブルを作成する手順および文法のバージョンなどの説明を行なう。

##### 4.1.1 文法ファイルの説明

###### 1. 文法のバージョン管理

オリジナル文法(日本 IR から納品された文法)のバージョン管理は、

```
atr-dp:/data3/HMM-LR/REF-GRA
```

で行なっている(管理人: 谷戸 文廣氏)。

###### 2. 文法の種類・名前

文法の名前は、以下の二種類の名前の組合せになっている。

###### (1) タスク別の分類

```
cstar      : CSTAR 用文法
eset       : 1500 語彙、機能評価文 600 文
mset       : 744 語彙、モデル会話 A、B、1-10
```

###### (2) 文法の適用対象別の分類

```
phrase     : 文節内
sentence   : 文節間
twolevel   : 2 段 LR パーザ
```

拡張子は、以下の種類がある。

```
.lex       : 文法規則 + かな漢字 (オリジナル文法)
.gra       : 文法規則
.dic       : かな漢字辞書
.tab       : LR テーブル
.cat       : 2 段 LR 用品詞表
```

現在(1993年2月26日)の時点での、主要なファイルのバージョンは以下の通り。

```
atr-dp/data3/HMM-LR/REF-GRA/CSTAR.v1-0y/cstar_phrase.lex
atr-dp/data3/HMM-LR/REF-GRA/GRA-E.v1-3/eset_phrase.lex
atr-dp/data3/HMM-LR/REF-GRA/GRA-M.v1-2a/mset_phrase.lex

atr-dp/data3/HMM-LR/REF-GRA/CSTAR.v1-0y/cstar_sentence.gra
atr-dp/data3/HMM-LR/REF-GRA/GRA-E.v1-3/eset_sentence.gra
atr-dp/data3/HMM-LR/REF-GRA/GRA-M.v1-2a/mset_sentence.gra

atr-dp/data3/HMM-LR/REF-GRA/CSTAR.v1-0y/cstar_twolevel.cat
atr-dp/data3/HMM-LR/REF-GRA/GRA-E.v1-3/eset_twolevel.cat
atr-dp/data3/HMM-LR/REF-GRA/GRA-M.v1-2a/mset_twolevel.cat
```

#### 4.1.2 文法の変換方法

SSS-LR で用いている文法は、音素ラベルの表記がローマ字とは異なり、また、特別に追加しなければならない規則(<start> <-> <\_start> や、語頭、語尾の無音モデル q1、q2 に関する規則)がある。

そのため、オリジナルの文法を、以下の手順で SSS-LR 用に変換しなければならない。

##### 1. 変換の手順

文法変換のための実行シェル (Harald Singer 氏作成) を用いる。

```
atr-fs:/pub1/common/src/singer/ATREUS/EVAL_NEW/GRA/change2sss.csh
```

README を参照のこと。

##### (1) 具体的な使用例 (cstar 用)。

```
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/CSTAR.v1-0y/cstar_phrase.lex
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/CSTAR.v1-0y/cstar_sentence.gra
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/CSTAR.v1-0y/cstar_twolevel.cat
cp cstar_twolevel.cat cstar_phrase.sss.twolevel.cat
cp eset_sentence.gra cstar_sentence.sss.gra
change2sss.csh -o cstar_phrase -b cstar_sentence
```

注: eset\_sentence.gra と cstar\_sentence.sss.gra は共通に用いる。

##### (2) 具体的な使用例 (eset 用)。

```
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/GRA-E.v1-4/eset_phrase.lex
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/GRA-E.v1-4/eset_sentence.gra
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/GRA-E.v1-4/eset_twolevel.cat
cp eset_twolevel.cat eset_phrase.cat
cp eset_sentence.gra eset_sentence.sss.gra
change2sss.csh -o eset_phrase -b eset_sentence
```

##### (3) 具体的な使用例 (mset 用)。

```
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/GRA-M.v1-2a/mset_phrase.lex
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/GRA-M.v1-2a/mset_sentence.gra
ln -s /NFS/atr-dp/data3/HMM-LR/REF-GRA/GRA-M.v1-2a/mset_twolevel.cat
```

```
cp mset_twolevel.cat mset_phrase.sss.twolevel.cat
cp mset_sentence.gra mset_sentence.sss.gra
change2sss.csh -o mset_phrase -b mset_sentence
```

## 2. 変換手順の説明

change2sss.csh の処理内容について、簡単に説明する。

- (1) postfilter 版の SSS-LR を用いる場合について。

```
ONE_LEVEL:
```

- (2) 音素ラベルの変換、規則の追加等。

```
# change to SSS format
org2sss.sed $ORGLX > $SSSLEX
```

- (3) .lex を 文法と、かな漢字辞書に分ける。  
grmake 実行の時、拡張子は不要。

```
# split into .gra and .dic
grmake $SSS
```

- (4) 分けた .gra に対して .tab (LR テーブル) を生成する。  
slr 実行の時、拡張子は不要。

```
# compile .tab
slr $SSS
```

- (5) 文節間文法について、.tab (LR テーブル) を生成する。  
slr 実行の時、拡張子は不要。

```
cp $BUNORG.gra $BUNORG.sss.gra
slr $BUNORG.sss
```

- (6) 2 段 LR パーザを用いる場合について。

```
TWO_LEVEL:
```

- (7) ファイル名の変更。

```
cp $SSS.gra $SSS.twolevel.gra
cp $SSS.dic $SSS.twolevel.dic
cp $ORG.cat $SSS.twolevel.cat
```

- (8) 2 段 LR パーザ用の追加 LR テーブル (品詞予測用) の生成。

```
# compile for 2-dan LR
slr2 $SSS.twolevel
```

## 4.2 日本語構文解析部の文法・辞書

### 4.2.1 文法・辞書のバージョン管理

日本語構文解析部の文法・辞書は

```
as26:/usr/project/asura/develop/translation/analysis/grammar
```

に各バージョン毎に管理されている。CSTAR プロジェクト版文法 (語彙や規則を縮小することにより高速な解析を可能にしているバージョンの文法) は同ディレクトリの CSTAR というサブディレクトリに作成日付別に整理されている。

### 4.2.2 文法・辞書のロード方法

通常日本語構文解析部は文法・辞書を二次記憶上に置いているが、デモシステムにおいては、システムの立ち上がり速度を改善するためと、文法・辞書ファイルの存在地と実行プログラムを依存させないことによる移植性の良さを尊重して、すべての文法・辞書情報を主記憶に展開する方法をとっている。そのため文法・辞書を変更するには解析実行プログラム (cstar.analysis.base) 自体を変更する必要がある。

解析実行プログラム (cstar.analysis.base) は defsystem という Lisp のツール (大雑把に言って unix 上の make ツールと同じようなものだと考えてよい) を利用してコンパイル、ロードされる。文法・辞書を変更するには、この defsystem が要求するシステム構成要素の依存関係記述ファイル (makefile のようなものものだと思っよい) を変更すればいい。

文法・辞書の修正は

```
as26:/usr/project/asura/develop/translation/analysis/system/CSTAR/grammar.system
```

を変更することにより可能となる。

以下に grammar.system の内容と変更すべき箇所を示す。

```
;;;
;;; defsystem for Japanese analysis grammar
;;;   by Masaaki Nagata, 91/1/14
;;;
;;;   changed by T.TASHIRO for CSTAR, 92/12/3
;;;

#+lisp
(defun standard-nonterminal-examine-predicate (symbol)
  (not (characterp symbol)))
#+lisp
(defun standard-terminal-examine-predicate (symbol)
  (characterp symbol))

#-lisp
(defun standard-nonterminal-examine-predicate (symbol)
  (not (stringp symbol)))
#-lisp
(defun standard-terminal-examine-predicate (symbol)
```



```
(stringp symbol) )
```

```
(defsystem grammar
  :source-pathname

  ;;; 文法・辞書のあるディレクトリ。
  ;;; 文法・辞書は日本 I R から一つのディレクトリにまとめて納入されるので
  ;;; 通常は下の行を変更するだけでよい。
  "/usr/project/asura/develop/translation/analysis/grammar/CSTAR/921209"
  :source-extension "lisp"
  :binary-pathname nil
  :binary-extension nil
  :initially-do
  (progn
    (defgrammar japanese-dialogue-grammar
      :doc "japanese grammar for understanding dialogue sentences"
      :ssymbol start
      :fstype-system-name standard-fstype-system
      :fstemplate-system-name japanese-dialogue-grammar-fstemplate-system
      :nonterminal-examine-predicate #'standard-nonterminal-examine-predicate
      :terminal-examine-predicate #'standard-terminal-examine-predicate)
      (begin-grammar japanese-dialogue-grammar))
    :finally-do
    (progn
      (end-grammar japanese-dialogue-grammar)
      (compile-grammar japanese-dialogue-grammar)
      (set-grammar japanese-dialogue-grammar)
      ;; The two functions below reset the feature structure
      ;; unification keeper and
      ;; the feature structure description expansion keeper.
      (fdesc-unification-keeper-reset)
      (fdesc-expansion-keeper-reset)
      )
  )

  ;;; 文法ファイルの名前・構成が変更された場合は以下の記述を変更する
  :components ((:module templates
                :source-pathname ""
                :components (("93fstempl2" ;;; テンプレートファイル
                              :load-only t)))
               (:module ps-rules
                :source-pathname ""
                :components (("93grammar-rules1" ;;; 文法規則ファイル
                              :load-only t))
                :depends-on (templates))
               (:module lexicons ;;; 語彙規則ファイル群。
                ;;; 助詞 (93particles1) および
                ;;; 活用語尾 (93inflections1) はモデル会話に
                ;;; 出現するものはすべてカバーするように
```

```

;;; するため独立している。
;;; その他の一般の語彙は CSTAR コーパス
;;; に出現するもののみを一つのファイル
;;; (93dict) にまとめている。
:source-pathname ""
:components (
  ("93particles1" :load-only t)
  ("93dict" :load-only t)
  ("93inflections1" :load-only t))
:depends-on (templates))

(:module sentences ;;; コーパスの文字列ファイル。
  ;;; 解析部の単体テストを容易にするため
  ;;; 組み込みデータとして用意している。
:source-pathname ""
:components (("93sample-sentences1" :load-only t))) )
:depends-on (acp) )

```

grammar.system の変更後、解析実行プログラム (cstar.analysis.base) は以下のような手順で作成する。

```

# 作業は atrp20(atrp19) 上で行なう。
# as26:/users/project/asura/develop がマウントされている必要がある。

$ cd /users/project/asura/demo/cstar/english/world/mk-acp

$ lisp
;;; HP Common Lisp, Development Environment, 12 August 1991.
;;; HP-9000, Series 700/800 Dev Lisp 92640, Rev. B.04.0.1
;;; Kanji Version
;;;
;;; Copyright (c) 1988, 1989, 1990, 1991 by Hewlett-Packard, Co., All Rights Reserved.
;;; Copyright (c) 1985, 1986, 1987, 1988, 1989, 1990, 1991 by Lucid, Inc., All Rights Reserved.
;;;
;;; This software product contains confidential and trade secret
;;; information belonging to Hewlett-Packard. It may not be copied
;;; for any reason other than for archival and backup purposes.

> (load "make-world-for-cstar.lisp")
#P"../cstar.analysis.base"

> (exit)

$

# /users/project/asura/demo/cstar/english/world に cstar.analysis.base が
# 新たに作成されている。

```

### 4.3 変換用規則ファイル

本節では、日本語から英語およびドイツ語への言語変換処理で必要となる、規則ファイルについて説明する。まず、その概要を説明した後、ファイルのロード方法について説明する。

以下では、ファイル位置として、

```
as26:/usr/project/asura/develop/translation/transfer/
```

から見た相対パス名を使用する。

#### 4.3.1 変換規則のバージョン

変換規則は解析処理の文法に合わせたものを使用しなければならない。CSTAR および daily demo に用いられている翻訳規則群は、12 のモデル会話を処理対象とした、m-set と呼ばれる規則体系が基本となっている。CSTAR 用の変換規則は、m-set 変換規則のサブセットに相当する。また、m-set 変換規則では、変換処理の大まかなサブプロセス単位にまとめられた、数種類のファイルから構成されているが、CSTAR 用変換規則は次の通り、1～2 個のファイルに納められている。

```
英語用：    rules/grammar-mset/main.schema2
            rules/grammar-mset/gr_met.euc
ドイツ語用： rules/german/1-59.rules.930119.1400
```

#### 4.3.2 変換規則のロード方法

変換規則をロードするには、変換処理系(素性構造書き換えシステム)を立ち上げた後(rws-v2/load-v2.hp.lisp をロードした状態)、次の規則用ロードファイルを読み込めばよい。必要な初期設定等は、処理系のロードファイル中に書かれている。

```
英語用：    rules/load-cstar.lisp
ドイツ語用： rules/load-cstar-german.lisp
```

#### 4.3.3 変換規則を更新する場合の注意点

当然ながら、規則のファイル名を変更する場合は、上の規則ロードファイルも更新する。規則がロードされている状態で、さらに同一固有名をもつ規則がロードされると上書きが行なわれ、新しい規則に置き換えられる。ただし、大幅に更新が行なわれる際には、処理系を立ち上げ直すか、次の関数により一旦規則をすべて削除してから、新しい規則ファイルをロードする。

```
(rws::remove-all-rw-rules2)
```

また、CSTAR 用の変換規則は主記憶版となっているが、将来多数の規則を用いるような場合には、二次記憶版(インデックス版)の使用が望ましい。その際には、規則をロードした状態で次のようにインデックスファイルと規則本体のデータファイルを作成しておく。

```
(rws::save-index-and-rules2 index-filename rule-filename)
```

これらのインデックスファイルと規則ファイルは通常、変換規則ファイルが置かれているディレクトリにセーブする。二次記憶版として動作させる場合は、インデックスファイル

のみをロードすればよい。大域変数 `*rws-rule-directory*` がセットされていれば、相対パスにより、システム実行の `current-working-directory` に関係なく、規則本体ファイルが参照可能である。この周辺の詳細は、文献 [7] を参照されたい。

## 4.4 生成用文法・辞書ファイル

本節では英語およびドイツ語の生成処理で必要となる文法・辞書関係のファイルについて説明する。まず、生成処理に必要な文法・辞書ファイルの概要について説明し、これらファイルのロードの仕方について CSTAR システムで用いられているファイルを用いて説明する。最後に文法・辞書ファイルを更新する場合の注意点について説明する。

[注意] パス名について

以下では特に断らない限りファイルの位置を

```
as26:/usr/project/asura/develop/translation/generation/
```

から見た相対パス名で表す。

### 4.4.1 生成用文法・辞書ファイルの概要

生成処理で必要となる文法、辞書関係のファイルは次の通りである。

1. 構文構造生成規則 (PD ファイル)  
構文を生成するための文法・辞書情報を規定する。
2. 形態素生成辞書 (mg dict)  
各単語の屈折変化のパターンを規定する。
3. 形態素生成規則 (mg rule)  
屈折変化のための文字列操作関数を規定する。
4. 形態素生成制御規則 (mg net)  
屈折変化を行なうための決定木 (decision tree) を規定する<sup>1</sup>。

これらのファイルには多くのバージョンがあり、原則として次のディレクトリに格納されている。

- 英語
 

構文構造生成規則 (PD ファイル)	pgen/pd-rule/
形態素生成辞書 (mg dict)	mgen/dict/
形態素生成規則 (mg rule)	mgen/mg-rule/
形態素生成制御規則 (mg net)	mgen/mg-net/
- ドイツ語
 

構文構造生成規則 (PD ファイル)	pgen/pd_ger/
形態素生成辞書 (mg dict)	mgen_ger/
形態素生成規則 (mg rule)	mgen_ger/mg-rule/
形態素生成制御規則 (mg net)	mgen_ger/mg-net/

---

<sup>1</sup>この決定木のことを mg-net と呼ぶ

辞書関係のファイルは原則として /usr/project/asura/develop/translation/ 配下のディレクトリに置かれているが、形態素生成ファイルだけは /usr/project/asura/demo/ ディレクトリの下にもコピーを置き、develop ディレクトリが見えなくてもアクセスできるようにしてある。これは、他の辞書ファイルが disksave 作成時のみに参照されるのに対して、形態素辞書ファイルは生成処理の実行時にもアクセスされるからである。

[注意] 歴史的経緯により、disksave 内ではデフォルトで develop の下の辞書を見るように設定されている。tra-gene.sh の記述によって、disksave 立ち上げ後に demo の下のコピーを見るように設定される。

#### 4.4.2 文法・辞書ファイルのロードの仕方

生成処理に必要な文法・辞書等のファイルのロードするには、生成処理を立ち上げたあと(生成エンジンをリスポにロードしたあと)で「生成処理初期ファイル」をロードする。

生成処理初期ファイルのパス名は

英語 genrc.921210.lisp

独語 genrc.ger.930119.lisp

である。

初期設定ファイルは生成処理で使われる文法・辞書等の定義だけでなく、生成処理系に与える各種パラメータが定義されている。システムで用いる文法のファイル名を変更する場合はこのファイルの内容を変更する<sup>2</sup>。次節で初期設定ファイルの例を示す。

[補足] ロードファイルについて

生成処理系では、生成エンジンのロードと上記初期設定ファイルのロードを全て自動的に行なうためのロードファイルを用意している。すなわち、このファイルをリスポ処理系に読み込むだけで、生成エンジンと必要な文法、辞書類がロードされ、各種の設定が完了する。ロードファイルのパス名は次の通りである。

英語 load.cstar.lisp

独語 load.german.cstar.lisp

#### 4.4.3 初期設定ファイル

英語用の初期設定ファイルを以下に示す。ドイツ語用のファイルも同様である。

```
*****
;;; 生成処理初期設定ファイル (CSTAR 用)
;;;
;;;
;;;-----
```

<sup>2</sup>もちろん、この初期設定ファイルで参照されている文法ファイル類を変更しても良い。

```

;;; 形態素生成規則
;;;
;;;      MGルール (形態素生成) ファイル名 morph. gen. (inflection) rule file
(mg-set-rule
  "/usr/project/asura/develop/translation/generation/mgen/mg-rule/defrule.lisp")
;;;      MGルールのローディング loading morph. gen. rules
(mg-load-rule)
-----
;;; 形態素生成制御規則 (MG ネット)
;;;
;;;      MG ネットファイル名 morph. gen. net. file
(mg-set-net
  "/usr/project/asura/develop/translation/generation/mgen/mg-net/defnet")
;;;      MG ネットのローディング
(mg-load-net)
-----
;;; 形態素生成辞書
;;;
;;;      マスター辞書ファイル名 master morph. gen. dic file
;;;
;;;      注意:ここに指定するファイルは形態素辞書のソースファ
;;;            イルである。生成処理に先だって、このソースファ
;;;            イルからインデックスファイル作成しておく。作
;;;            成の仕方は次節で述べる
(mg-set-m-dict
  "/usr/project/asura/develop/translation/generation/mgen/dict/atr_made.dict5")
;;;
;;;      ユーザ辞書使用の可否
(mg-use-u-dict nil)
-----
;;; 文構造生成規則
;;;
;;;      CSTARでは二次記憶版のPDは利用せず、オンメモリPD
;;;      のみを利用する。
;;;
;;;      メモリーPD使用の可否
(set-o_pd t)
;;;      メモリーPD用PD規則ファイルのパス名
(set-o_pd_path
  "/usr/project/asura/develop/translation/generation/pgen/pd-rule/pd.mset58.lisp")
;;;      メモリPD規則のロード
(load_pd :init t) ;;この関数呼び出しによってPD規則がロードされる
;;;
;;;      二次記憶マスターPD使用の可否
(set-m_pd nil)
;;;      二次記憶ユーザーPD使用の可否
(set-u_pd nil)
;;;
;;;+-----+

```

```

;;;
;;; 変換結果ファイル名 transout file name
;;;
;;; 生成処理系単独で実験する場合にとして用いる変換結果ファイル
;;;
(set-transout_path
  "/usr/project/asura/develop/translation/generation/pgen/fsdata/mset58.loop")
;;;
;;; 変換結果ファイルへのインデックス作成
;;;
(load_transout)
;;;+++++
;;;
;;; 英文生成処理パラメータの設定
;;;
(g-set-debug nil)          ;; デバッグ情報なし
(set_pruning t)           ;; 枝刈りあり
(set_severity 2)          ;; PD 適用失敗時の動作
(set_gen_direction :top-down) ;; PD 連鎖の作成方向 (下降)
(set_expand_mode :para)   ;; 探索方式
(structure_sharing t)     ;; 構造共有
;;;
;;;
(mg-start)                ;; 形態素辞書オープン
(gg 1 50)                 ;; 変換結果ファイルの 1-50 番までの変換結果に対
                          ;; して、生成処理を実行 (規則連鎖テーブルの学
                          ;; 習)
(mg-quit)                  ;; 形態素辞書クローズ
;;;*****

```

#### 4.4.4 文法・辞書ファイルを更新する場合の注意点

##### 1. 形態素辞書 (⇒ インデックスファイルの作成)

形態素生成辞書はその都度 disk にインデックスアクセスする。このため、インデックスアクセスに必要なインデックスファイルとデータファイルとをあらかじめ作成しておかなければならない。

インデックスの作成法は次の通りである。

- (a) 辞書ソースファイル名を XXX とする
- (b) 生成処理系を立ち上げて

```
(make_m_dict "XXX")
```

を実行する。

- (c) XXX.data, XXX.index という二つのファイルが作成される。

##### 2. 形態素生成規則 (⇒ コンパイル)



形態素生成規則ファイルはあらかじめコンパイルしておくのが望ましい。コンパイルはリスプのコンパイラを用いる。

### 3. 形態素生成制御規則 (⇒ コンパイル)

形態素生成制御規則ファイルも形態素生成規則と同じく、あらかじめコンパイルしておくのが望ましい。コンパイルはリスプのコンパイラを用いる。

### 4. その他の注意事項

#### (a) 形態素辞書のクローズ

disksave を作成する場合には必ず形態素生成辞書をクローズすること。クローズする関数は次の通りである。

```
(mg-quit)
```

disksave を立ち上げた時に

```
(mg-start)
```

とすれば辞書はオープンされる。

ASURA では、辞書のクローズを disksave 作成用シェルスクリプト (make-nlp.csh) で実行し、オープンを変換・生成処理起動用シェルスクリプト (tra-gene.sh) で実行している。

#### (b) 各種設定の表示

生成システムが立ち上がっている状態で

```
>(GS)
```

を実行すると、システムの基本的な設定やシステムが現在参照している文法、辞書ファイルなどが表示される。ある disksave で利用されている文法や辞書が分からない場合はこの関数で調べると良い。

#### (c) 生成実験

生成処理系が参照している文法・辞書が正しく動くかどうか確認するために、「正解変換結果ファイル」を入力として生成単独で実験を行なうことができる。この方法は次の通りである。

```
> (mg-start)
```

```
$> (gg <begin> <end>) ;; <begin> <end> は変換結果番号  
;; <begin> から <end> までの文を  
;; 生成する。
```

## 4.5 日本語音声合成システムに必要なデータ / ファイルフォーマット

## 4.5.1 CMU 側入力データ

```
MOSHI MOSHI @ KAIGI JIMUKYOKU DESU KA @
```

```
↑
```

‘@’ はポーズの挿入位置を示す。

## 4.5.2 Siemens 側入力データ

```
[[interj(moshimoshi,4,1,2)],[[n(kaigijimukyoku,7,5,2),cop(desu#,2,0,2)],s_prt(ka,1,0,2)]].
```

- 1ワードのフォーマット例

品詞コード	単語の発声表記 (へボン式ローマ字)	アクセント情報その他		
		mora 数	accent 型	accent 価
interj	(moshimoshi,	4,	1,	2)

- また、[] は係り受け構造を表している。

## 4.5.3 単語情報ファイル (\*.word)

moshimoshi	moshimoshi	感動詞	-	-	1	1	4	1	2
kaigijimukyoku	kaigijimukyoku	普名詞	-	-	1	5	7	5	2
desu	desu#	助動詞	-	-	2	0	2	0	2
ka	ka	普名詞	-	-	3	0	1	0	2

## 4.5.4 アクセント句情報ファイル (\*.phra)

moshimoshi	1	0	4	0	0	0	0	1	0.000	0	1
kaigijimukyokudesu#ka	5	0	9	0	0	0	0	5	0.000	0	2

‘#’ は、その前にある音韻が無声化していることを示す。

## 4.5.5 マッチング用リストファイル (CHK\_list)

```
chk_019  moshimoshikaigijimukyokudesuka
  ↑                ↑
  文番号   ヘボン式ローマ字列 (accent 区切りなし)
```

文番号で、合成音声ファイル (~ / WV / \*. WV) との対応を取る。合成音声ファイル (~ / WV / \*. WV) には、文番号と同じ名前 (この場合なら、chk\_019.WV) を付けてある。

## 4.5.6 辞書ファイル

表 4.1: 日本語音声合成のための辞書ファイル例

文字列	品詞コード	単語の発声表記 (へボン式ローマ字)	アクセント情報その他		
			mora 数	accent 型	accent 価
AH	感動詞	aa	2	1	2
AI-LAB	固名詞	eiairabo	6	0	2
AI.LAB	固名詞	eiairabo	6	0	2
AIDA	普名詞	aida	3	0	2
AKI	普名詞	aki	2	0	1
AKIGA	格助詞	akiga	3	0	1
AMERIKA	固名詞	amerika	4	0	1
ANATANO	格助詞	anatano	4	0	1
ANNAI	普名詞	aNnai	4	3	1

図 4.1 に Siemens システムにおける日本語生成用辞書 (lexicon) フォーマットを示す。ただし、日本語合成音声モジュールと関係があるのは、品詞、音声表記、アクセント情報 (phon\_lex) である。詳しくは、文献 [8] を参照のこと。

```
lexicon(moshimoshi,interj) | ..... タグ
phon_lex(interj,interj(moshimoshi,4,1,2)), ..... 品詞、音声表記、アクセント情報
interj:sem:expr = 'tel_phatic(interj:sem:var)..... 独日翻訳モジュール用情報
interj:tree = interj(moshimoshi). ..... 独日翻訳モジュール用情報
```

図 4.1: Siemens システムにおける日本語生成用辞書 (lexicon) フォーマット

## 4.5.7 CMU 側入力と Siemens 側入力の違い

## ☆ CMU 側

入力データ処理では、文字列 (4.5.1 節) のみを受け取り、用意してある辞書ファイル (4.5.6 節) の中から検索を行ない、単語情報ファイルに必要なデータを取り出す。また、係り受け構造の生成は、辞書ファイルの品詞コードより簡単な規則を用いて生成している (文末以外は、右枝分かれ構造としている)。

## ☆ Siemens 側

入力データ処理では、辞書の検索を行ない、品詞、発声表記、アクセントデータが付加された文字列 (Siemens 側の翻訳モジュールの出力データ、4.5.2 節) を入力データとして受け取る (Siemens 側の翻訳モジュールは、表 4.1 の辞書ファイル例と同じ辞書を用いている。図 4.1 に Siemens 側の翻訳モジュールで使用している辞書のフォーマットを示す)。そのデータをもとに、単語情報ファイルを作成する。また、終助詞にともなう係り受け構造の補正処理を行なっている。

#### 4.6 その他、特記事項 (トラブルシューティングなど)

- プログラム修正等を行なう時  
プログラムの修正等が必要な場合、先に

```
% unsetenv LANG
```

とすること。

- 翻訳コントロールシステムからのエラー  
CSTAR システムを長時間あげたままで置いておくと、次のようなエラーメッセージをばく時がある。

```
select() error. in usleep (Translation)
```

または、

```
select() error. in socket_clear
```

このエラーは他のプロセスからの割り込みがあった時に上がる。CSTAR システム自体には特に影響はない。

- IOPBOX 不調の時

現象: 音声が入力できなくなる (ParaIn の波形が途中で止まってしまう)。この状態でオプションパネル等で強制的に処理を先に進めてしまうとマシンが落ちてしまうことがある。

対処: 1. CSTAR システムを一度、QUIT する。  
2. IOPBOX の電源を入れ直す。  
3. CSTAR 関連のプロセスが残っていないかチェックする。  
4. 再度、CSTAR システムを立ちあげる。

※ もし、マシンが落ちてしまった場合は、先に IOPBOX の電源を入れ直しておくこと。

## 第 5 章

### 自動翻訳電話国際共同実験のための対話コーパス

自動翻訳電話の国際共同実験 (CSTAR) で使用した対話コーパスについて、その決定までの経緯と最終版の内容について述べる。

#### 5.1 CSTAR プロジェクト開始までの経緯

CSTAR が発足する以前に存在していた対話コーパスは次の通りである。

1. ATR で作成した、国際会議への参加に関する問合せのモデル会話 A, B, 1 ~ 10
2. CMU で作成した、ATR のモデル会話に対応する対話コーパス 1 ~ 12

CMU (JANUS) 版の対話コーパスは、ほぼ ATR 版にならって作られているが、次のような点が異なっている。

- 会話番号の対応のずれ
- ATR 版と異なる固有名詞の使用
- ATR 版にはない、名前のスペル確認・応答文の挿入

CSTAR プロジェクトの開始に伴って、SIEMENS 版のドイツ語コーパスも作成されたが、会話番号を除いて同様の違いが見られる。また、発話の発声単位が、ATR (ASURA) と CMU (JANUS) では異なる点がある。たとえば、ASURA では「もしもし」「そちらは会議事務局ですか」が 2 文として、それぞれ認識・翻訳されるのに対し、JANUS では “Hello, is this the conference office?” が、続けて一文として発声され、認識・翻訳処理される。次のように、ASURA では 3 発話となるものが、一連の連続した発声として入力される場合もある。これらの相違点は、実際のデモ用対話コーパスを作成するに際しての留意事項となった。

ex.) Hello @ this is the conference office @ May I help you?

## 5.2 CSTAR プロジェクト発足後の経緯

3ヶ国間で、通信回線を通した双方向の音声翻訳実験を行なうことが合意されたことで、本報告の他の章に触れられているような種々の調整が必要となったが、最終的なデモの場でのどのような会話を実際にやりとりするかについては、しばらくの間具体的な提案はなかった。これは、デモの形態等についての議論が、かなり遅い段階になって始められたこと、さらに開発が先行していたCMU, ATRに比較して、SIEMENSの開発の見通しがはっきりしていなかったことによる。

ただし、1991年7月のミュンヘン会談(CSTAR発足についての最初の全体会合)と1992年1月のピッツバーグ会談(CSTARの名称が決定した2回目の全体会合)を経た後、次のような合意がとられた。

- 各サイトのコーパスを尊重し、開発作業はそれぞれのコーパスに従って行なう。
- 各サイトのコーパス間の違いを十分認識した上で、最終のデモ用コーパスを設計する。

上記の2番目の項目については、各サイト間で重要性の認識にずれがあったことは否めない。ATRでは、1992年の始め頃から、日独版の作成のため、Mark Seligman, Herbert Tropfの両氏が活動を開始したが、彼らが最初に取り組んだのは、3ヶ国語のコーパスの徹底的な比較分析であった。その過程で、双方向の対話コーパスを設計する上での種々の問題が指摘された。前節で挙げた相違点はその一部であるが、その他にも、一貫性のある翻訳結果として、技術的実現可能性を考慮した上で、どのような表現が適切かなども、ドイツ語版の作成に留まらない問題である。上記の比較レポートは、相手側サイトにも送付され検討を依頼したが、十分活用されたかどうかは定かではない。

さて、最終的なデモ用のコーパスを作成するに当たっては、各システムの特徴から来るもう幾つかの問題点があった。

- CMU, SIEMENSのシステムにおいては、単語(クラス) Bigram等を用いた認識方式を採っており、その学習範囲がほとんど最初のコーパス内に留まっているため、発話文の表現の変更が事実上不可能である<sup>1</sup>。従って、コーパス内の各発話は、そのまま使用するか、全く使わないか、のいずれかとなる。
- ATRのシステムでは、特に翻訳処理に膨大な計算コストがかかるため、単語数の多い長文は処理時間の点で不適切となる。また、極端に認識率の悪い文節を含むような文があれば問題となる。

これまで述べたような種々の問題点を背景に、1992年9月の京都(ATR)会談を機に、デモ用コーパスの作成は最終段階に入った。

## 5.3 デモ用コーパスの決定

ATRでの会合を前に、ATR内部ではA, B, 1~10の12会話の中から、音声認識率・翻訳時間・会話の内容等、様々な観点から、候補となりそうな会話の選択を試みた。これ

<sup>1</sup>CMUの場合、英日翻訳システムが扱える表現がもともとかなり限定されており、しかも、それを保守できる要員が既になかったため、翻訳という観点からも、コーパスを変更することは無理であった。SIEMENSの場合、その頃まだ日独翻訳システムの開発途上であった。(竹沢)

は、開発用のコーパスに準拠した複数の会話が、実際のデモで使用され得る、という前提に基づくものである。この時点で候補に挙げたのは、次の4会話である。

会話番号	サブトピック	備考
A	登録方法(用紙の送付)	最も基本的な会話, 固有名詞の問題
2	参加料について	基本的な会話だが、表現の修正が必要
6	市内観光	内容的に面白いが、長い文の短縮が必要
9	会議場への交通手段	同上, 固有名詞の問題

それぞれについて、固有名詞等の問題点を考慮した双方向コーパス案を作成し、会合に臨んだが、コーパスについての各サイトの考え方にギャップがあり、結局この場では結論までに至らなかった。しかし、デモまでに全体で議論する機会もないので、各サイトがそれぞれ主導するデモにおいて実行したいコーパスを相手側に提案して、それについて交渉することが合意された。

1992年9月の会合後に、ATRでは会話2と6に絞る方向でコーパス決定作業を進めていたが、10月末頃SIEMENS側(広報担当のHeller氏)から具体的な提案が寄せられた。それは、SIEMENS主導の独日デモでは、会話3(会議の内容について)の一部を使用したいという申し入れであった。その他、具体的なデモのイメージを考えているようであった。ATR側では、早速その内容を検討し、日本側の応答内容で一文だけ長い文を分割して二文にすることを前提として、SIEMENS側の提案に応じることにした。

さらに、ATR側でもCMUおよびSIEMENSに対して、それぞれ会話2と6に基づくデモ用コーパスの具体的な提案を行ない、次のように最終的な合意が得られた。

デモの主催ホスト	会話内容	問合せ者側	事務局側
ATR	参加費用について(会話2) 市内観光(会話6)	ATR SIEMENS	CMU ATR
SIEMENS	会議場への交通手段(会話9) 会議の内容について(会話3)	SIEMENS SIEMENS	CMU ATR
CMU	会議場への交通手段(会話9) 市内観光(会話6)	CMU ATR	SIEMENS CMU

ATRが関与するコーパスについては、巻末の付録Aを参照されたい。前述したように、音声認識率の向上と翻訳処理時間の短縮を意図して、日本語の入力文(の一部)の表現を変更したり、発話そのものを削除したりした。もちろん、対話としての自然な流れを損なわないよう配慮している。また、リハーサル段階でも、わずかながら修正が加えられた。





## 第 6 章

### 自動翻訳電話国際共同実験におけるテレビ会議システム等

#### 6.1 自動翻訳電話国際共同実験におけるテレビ会議システム

自動翻訳電話国際共同実験で利用したテレビ会議システム (Teleconference System) の説明を行なう。

今回の自動翻訳電話国際共同実験では、各サイトにおける音声翻訳処理にかなりの時間 (10 秒程度) を要する。これは、実験の見学者にとってかなり長い時間であり、対話が間延びしたものと感じられる恐れがある。この対策の一環として、音声翻訳システムの現在の処理状況を相互に伝送して表示することとした。しかし、これだけでは必ずしも十分な改善効果が期待できないため、テレビ会議システムを併用することとした。

テレビ会議システムの符号化方式は標準化されてはいるが、実際には異なるメーカーのシステム間ではうまく通信できないという問題がある。そこで、テレビ会議システムとしては、日米独 3ヶ国いずれでも利用可能な米国 PictureTel 社製システム 4200 を利用することとした。また、各サイトで他の 2 者間の自動翻訳電話実験をモニタしたいとの要求があるため、それぞれテレビ会議システムを 2 式用意してすべてのサイトが常時テレビ会議で結べるようにした。国際 ISDN ネットワーク経由で日米独を結び、日独間は 64kbps×2、日米・米独間は 54kbps×2 でデータ転送を行なった。具体的なテレビ会議システムのネットワーク構成を図 6.1 に示す。実際の運用では米国サイトが手続きの関係から国際への発呼が不可能なため、日本・ドイツからの着呼のみを受け入れる形態となった。また、日本・ドイツ間は ISDN 回線の容量が少ない (4 回線) ため、迂回ルート (フランス) 経由で 1 チャンネルのみで接続する事例が見られた。

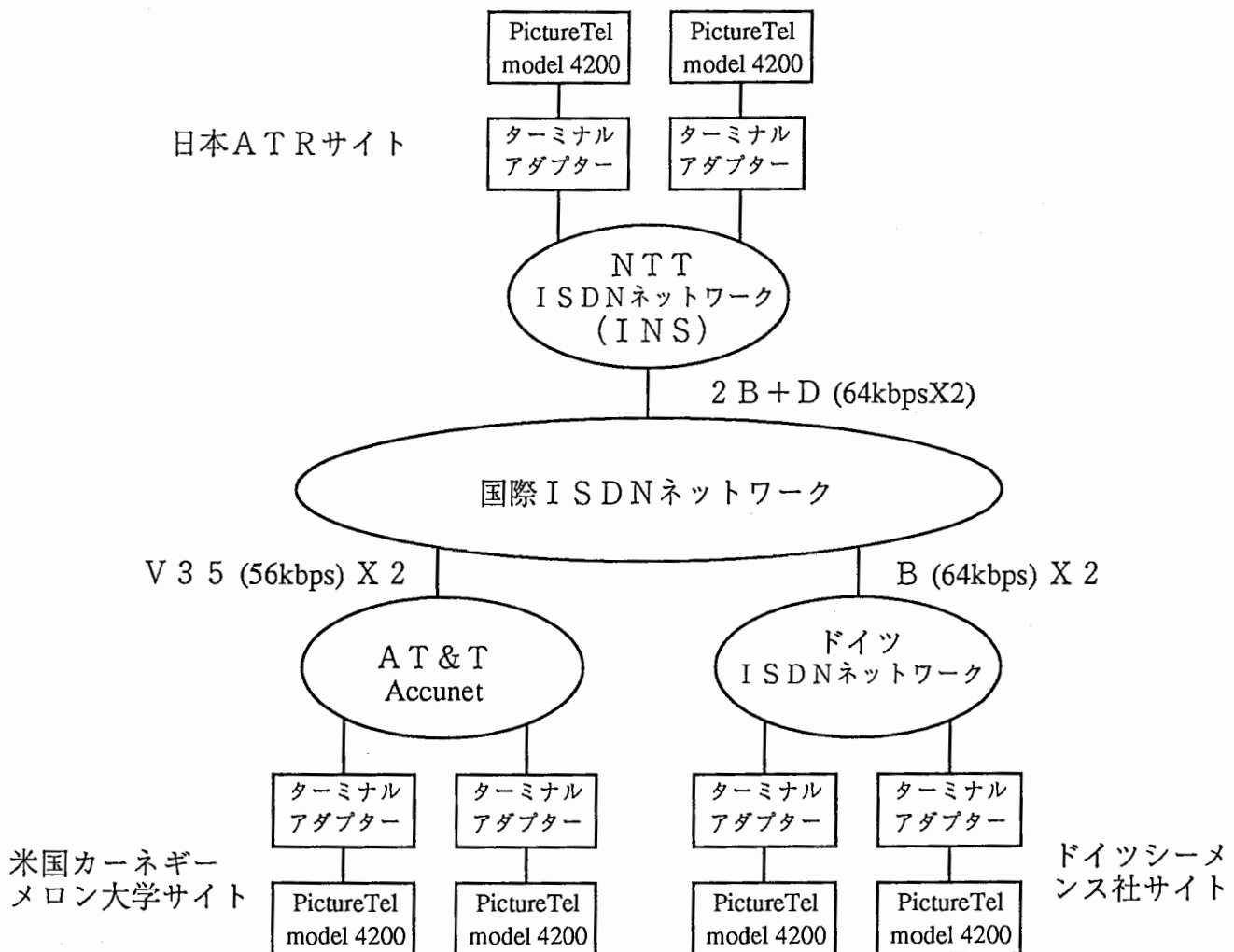


図 6.1: テレビ会議システムのネットワーク構成

ATR サイトにおけるテレビ会議システムの機器構成を図 6.2 に示す。動画像入力には、8 ミリビデオカメラ (SONY CCD-TR900) を使用した。このカメラを用いることにより、デモ会場の照明をかなり暗くしても発話者の表情を鮮明にとらえることができた。また、音響系では相手サイトの音声の回り込みを防ぐため、発話者の接話型マイクからの入力および日本語音声合成装置からの出力をミキサー経由でテレビ会議システムのライン音声入力端子に送る形式とした。この結果、テレビ会議システム附属のテレビカメラおよびマイクは使用しなかった。

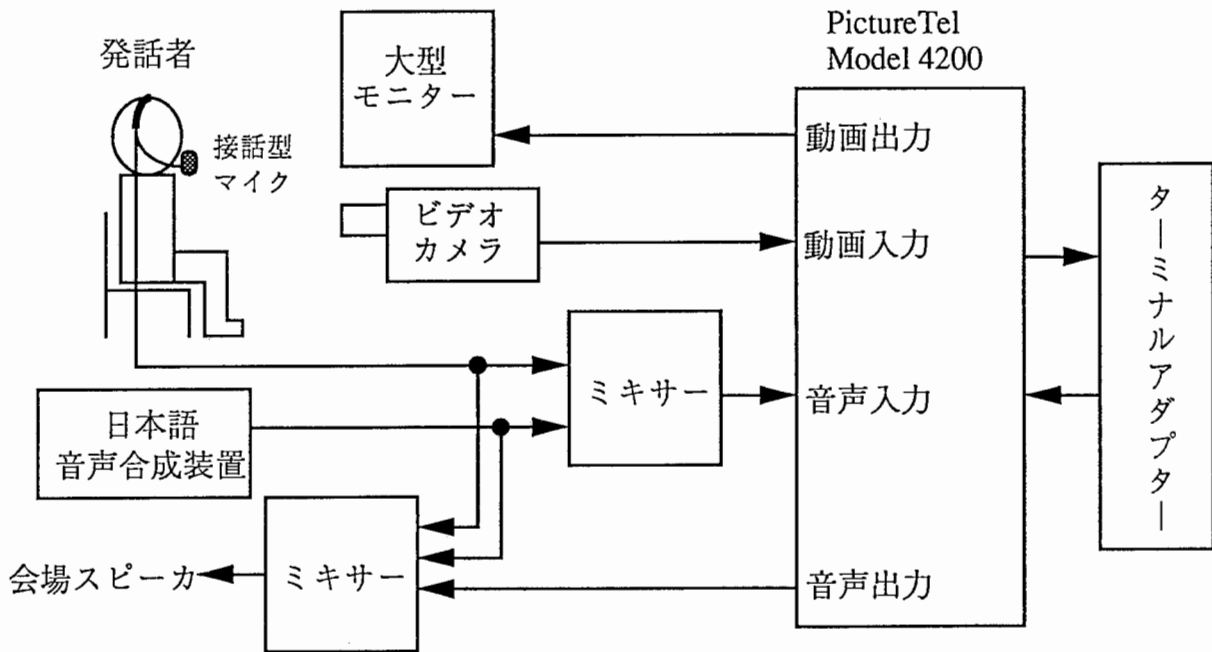


図 6.2: ATR サイトにおけるテレビ会議システムの機器構成

各サイトの発話者にテレビ会議システム使用の感想を聞いたところ、すべての発話者がテレビ会議システムを併用して良かったと答えている。その理由としては、

1. 音声翻訳の処理時間はかなり長い、テレビ画像を見ることで気が紛れる。
2. 相手の表情から翻訳内容が相手に理解されたことを確認できる。
3. 発話のタイミングが取りやすい。

などの点を挙げていた。したがって、テレビ会議システムの併用はデモ演出上の効果ばかりでなく、発話者にとっても、ユーザインタフェースを改善する上で有効であったと結論付けられる。

## 6.2 自動翻訳電話国際共同実験における音響系等

自動翻訳電話国際共同実験で利用した音響系の接続構成、および、ビデオプロジェクタへの接続について述べる。音響系の接続は、ミキサー2台を用い、各々をCMUサイトとSiemensサイトに割り当てた。この時のミキサーの接続を図6.3に示す。これは、相手サイトの音声の回り込みを防ぐためである。CMU-Siemens間のデモの間は、両サイトへの音声出力を切った。ビデオプロジェクタとの接続を図6.4に示す。大会議室のビデオプロジェクタとの接続には、スキャンコンバータ(山下電子設計, CVS-980N)を用いた。これにより、CSTAR用メイン画面を大画面に出力できるようになった。

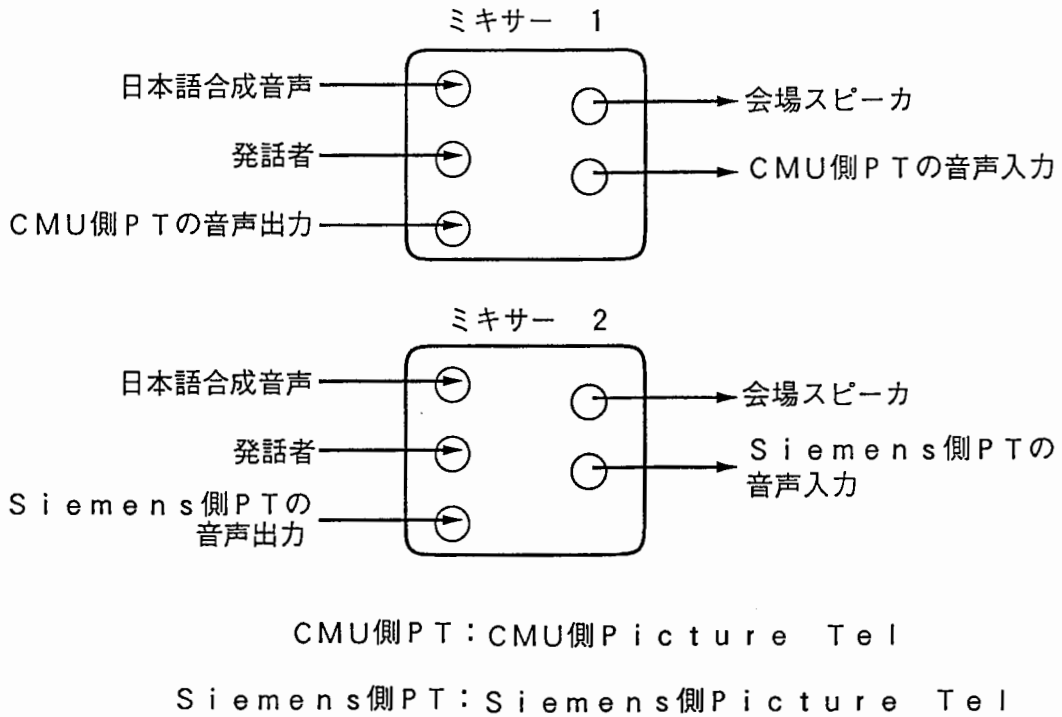


図 6.3: ミキサーの接続構成

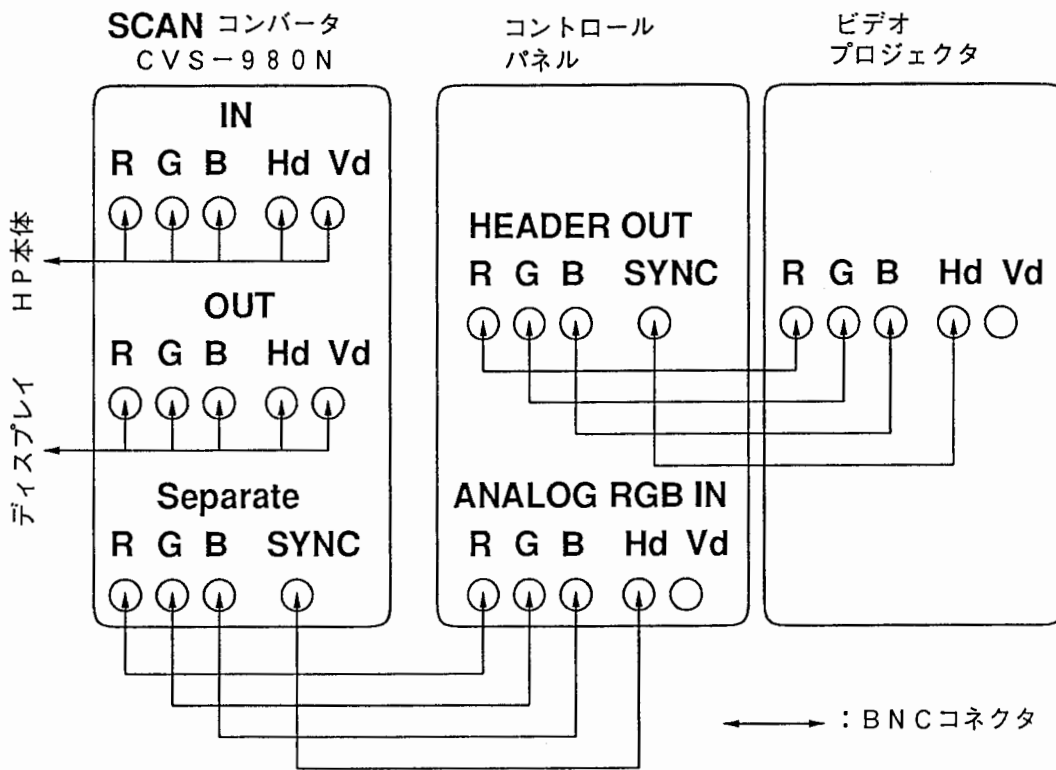


図 6.4: 大会議室のビデオプロジェクタとの接続

## 第 7 章

### daily demo 用 ASURA の概要

第 7 章ではプロトタイプ室で日々の乗客のためのデモを行なうための (daily demo 用) ASURA に関する簡単な説明を与える。このバージョンの ASURA は今後も改良が加えられるに違いないので、簡単にしか触れない。

#### 7.1 動作環境

HP9000/750 ワークステーション上で動作する。音声認識処理に 1 台 (atrp13)、言語翻訳処理に 1 台 (atrp20) のマシンを使っている。英語音声合成は DEC Talk を利用し、ドイツ語音声合成はベンツ社の試作品を利用している。

#### 7.2 操作方法

##### 7.2.1 起動方法

1. atrp13 で demox を起動する。
2. 開かれたメニューから **ASURA** を選択する。
3. 開かれたメニュー (図 2.5) から話者名を選択する。
4. 開かれたメニュー (図 2.7) から **Speech\_Translation** を選択する。
5. しばらく待つと、ASURA システムが起動される。

##### 7.2.2 基本的な操作方法および音声入力方法

音声入力などの操作は 2.2.2 節と同じ。

##### 7.2.3 終了方法

1. 音声入力窓で q と入力する。
2. 開かれたメニュー (図 2.7) から **Exit** を選択する。
3. 開かれたメニューから **\*\*quit\*\*** を選択する。
4. 開かれたメニューから **QuitAll** を選択する。

### 7.3 話者適応

起動方法の3で開かれるメニューが図2.5に相当する画面である。daily demo 用 ASURA における話者適応の操作方法は、2.3節と基本的に同じ。

### 7.4 システムパラメータの初期設定と変更

話者名選択後のメニュー画面(図2.7)で **Options** を選択すると、ユーザに許されている範囲で、システムパラメータの変更ができる。いろいろなモード(上から3位までの文候補を自動的に翻訳へ送るか、あるいは、オペレータがマニュアルで文を選択するかなど)が用意されている上に、文法ファイルを差し替えることさえ可能である。しかし、日々変更が加えられていることが多いので、詳しいことには触れない。なお、文法ファイルを変更する場合、文節内文法と文節間文法が別々に選択可能であるが、許される組み合わせが限られているので、不用意に文法を変更しないこと。

## 第 8 章

### むすび

ATR 音声翻訳システム ASURA のプログラムの実装方法および操作方法について、自動翻訳電話国際共同実験のために特別に開発されたバージョンを中心に報告した。これまでずっと謎に満ちていた ASURA の実装方法が少しでも皆に伝われば幸いである。ASURA のように音声認識から言語翻訳へ単に階層的につなぐ手法がよいかどうかについては議論が分かれるであろうが、この実装方法は現在利用可能なマシン等を使ってそれなりにかなり技を極めているような気がしている。今後もまだしばらくこのシステムを保守して改良することになろうが、これとはまったく異なる音声翻訳システムアーキテクチャを検討する必要も感じる。

## 謝 辞

本システムの構築・実装には、筆者ら以外にも多くの人の努力と貢献がありました。

保坂順子氏は音声認識のための文法の開発および Siemens で開発された独日翻訳システムの日本語生成に寄与されました。永田昌明氏は翻訳のための日本語解析部の開発に寄与されました。Herbert S. Tropf, Mark Seligman 両氏はドイツ語への翻訳の開発に寄与されました。匂坂芳典、海木延佳(現在、シャープ株式会社)、岩橋直人 各氏は日本語音声合成システム ATR v-Talk の開発に寄与されました。嵯峨山茂樹、杉山雅英、大倉計美、山口耕市、藤原紳吾各氏は音声認識システムの構築に寄与されました。

日本アイアール株式会社の関倫彦さん、衛藤純司さん、坂口明子さんらは翻訳のための文法の開発に寄与されました。株式会社 東洋情報システムの松尾秀彦さん、古崎博久さん、渡辺学さん、高橋誠さんらは翻訳のためのプログラムの開発に寄与されました。

林輝昭氏は音声認識と言語翻訳を接続する部分のプログラム開発およびシステムの評価実験に寄与されました。泉昌明氏は画面の開発等に寄与されました。久世祥司氏は SUN/SPARC を中心にマシンが安定して動作することに寄与されました。小野佳範、山口毅 両氏はネットワークの整備やマシンの管理に寄与されました。

記して感謝いたします。

さらに、国際共同実験の会場設営時に必要となった音響系のための接続コード作成を手伝っていただいた浦谷則好、小坂哲夫、大倉計美、片岸一起 各氏、大画面に投影するための BNC 延長コードを快く貸して下さった ATR 通信システム研究所 竹村治雄氏、並びに、リハーサルおよび実験当日に音響系接続について貴重な御助言をいただいた KDD テレサーブ 安永正幸氏に、この場を借りて、心より感謝申し上げます。



## 参考文献

- [1] 竹沢寿幸, 森元逞, 谷戸文廣, 鈴木雅実, 嵯峨山茂樹, 樽松明 (1993-03): “ATR 音声言語翻訳実験システム ASURA”, 情報処理学会第 46 回全国大会, 6B-5.
- [2] 嵯峨山茂樹, 鷹見淳一, 永井明人, ハラルド・シンガー, 竹沢寿幸, 谷戸文廣, 鈴木雅実, 森元逞, 樽松明 (1993-03): “自動翻訳電話実験システム ASURA の概要”, 日本音響学会平成 5 年度春季研究発表会 講演論文集, 3-4-17.
- [3] 鈴木雅実, 菊井玄一郎, M. Seligman, H. Tropsf, 森元逞, 樽松明 (1993-03): “日独音声言語翻訳実験システム”, 情報処理学会第 46 回全国大会, 6B-6.
- [4] Y. Sagisaka, N. Kaiki, N. Iwahashi, K. Mimura (1992-10): “ATR  $\nu$ -Talk Speech Synthesis System”, *Proc. of ICSLP-92*, pp. 483-486.
- [5] 海木延佳, 三村克彦, 岩橋直人, 匂坂芳典, 道尻佳憲 (1992-10): “ATR  $\nu$ -Talk 音声合成実験システム”, *ATR Technical Report*, TR-I-285.
- [6] H. Singer, M. Sugiyama (1993-03): “Manual for Speech Input Programs: SpeechIn, SpeechPress”, *ATR Technical Report*, TR-I-302.
- [7] 鈴木雅実, 古崎博久 (1993-03): “言語変換処理系解説書”, *ATR Technical Report*, ATR-I-0330.
- [8] J. Hosaka (1993-03): “A Grammar for Japanese Generation in the TUG Framework”, *ATR Technical Report*, TR-I-346.



## 付録 A

### 国際共同実験用コーパス

コーパスのファイル名のつけ方は次の通りです。

- 最初の文字 数字で、基になったモデル会話番号を示す (2,3,6)
- 2 番目の文字 J, E, G のどれかで、問い合わせ者の入力言語
- 3 番目の文字 J, E, G のどれかで、事務局側の入力言語

現在、2JE, 3GJ, 6GJ, 6JE の4ファイルが存在します。その会話固有のトピックや使用されるときデモの主催者等の情報をファイルの冒頭にコメントしてあります。

各ファイル内の発話番号 (形式 F-XYZ)

- F 最初に prefix としてファイル名 (会話 ID) がついています (3桁)
- X 発話者のサイト位置 K (京都) P (ピッツバーグ) M (ミュンヘン)
- Y 同一会話 ID 内で同一サイトの何番目のターンかを示す数字 (1, 2, 3, ...)
- Z 上の同一ターン内での発話のシリアル番号 (1, 2, 3)

- 例：
- 2JE-P31 ピッツバーグ (CMU) 側 3 回目のターンの最初の発話
  - 3GJ-M42 ミュンヘン (Siemens) 側 4 回目のターンの 2 番目の発話
  - 6GJ-K23 京都 (ATR) 側 2 回目のターンの 3 番目の発話

なお、入力言語での発話の下に () で包んだ翻訳結果を記してあります。

## A.1 2JE (参加料の問合せ)

ATR = 問合せ者 vs. CMU = 会議事務局

- (話者) (発話番号)
- ATR 2JE-K11 もしもし。  
(Hello.)
- CMU 2JE-P11 Hello, this is the conference office. May I help you?  
(もしもし, こちらは会議事務局です。ご用件は何でしょうか。)
- ATR 2JE-K21 会議の/参加料について/教えてください。  
(Please tell me about the attendance fee of the conference.)
- 2JE-K22 現在/参加料は/いくらですか。  
(How much is the attendance fee right now?)
- CMU 2JE-P21 Well let's see. It costs two hundred and fifty dollars per person.  
(ええとそうですね。一人あたり 250 ドルかかります。)
- 2JE-P22 If you apply next month it will cost you three hundred and twenty five dollars.  
(もし来月申し込めば 325 ドルかかります。)
- 2JE-P23 The proceedings and the reception are included in the application fee.  
(予稿集とレセプションが参加料金に含まれています。)
- ATR 2JE-K31 わたしは/情報処理学会の/会員なのですが。  
(I'm a member of the Information Processing Society.)
- 2JE-K32 参加料の/割引は/ないのですか。  
(Isn't there a discount of the attendance fee?)
- CMU 2JE-P31 No, there is no discount this time.  
(いいえ, 今回は割引はありません。)
- ATR 2JE-K41 わかりました。  
(I see.)
- 2JE-K42 参加料は/どのように/お支払いしたら/よいのですか。  
(How should I pay the attendance fee?)
- CMU 2JE-P41 Please remit to our bank account which is mentioned in the announcement.  
(案内に書いてあるわたしどもの銀行口座へ送金してください。)
- 2JE-P42 The deadline is the end of the year.  
(締め切りは年末です。)
- ATR 2JE-K51 わかりました。  
(I see.)
- 2JE-K52 ありがとうございます。  
(Thank you.)
- CMU 2JE-P51 You're welcome.  
(どういたしまして。)
- ATR 2JE-K61 失礼いたします。  
(Good-bye.)

## A.2 6JE (市内観光)

CMU = 会議事務局 vs. ATR = 問合せ者

(話者) (発話番号)

- CMU 6JE-P11 Hello. Conference office.  
(もしもし, 会議事務局です.)
- ATR 6JE-K11 会議の／間に／市内観光が／あるそうですが。  
(I've heard that there's a city sightseeing during the conference.)
- 6JE-K12 まだ／参加可能ですか。  
(Can I still attend?)
- CMU 6JE-P21 Yes, you can.  
(はい, できます.)
- 6JE-P22 We will visit Heinz Hall Mount Washington and the Mellon Museum on the  
afternoon of August fifth.  
(八月五日の午後にハイツホールとマウントワシントンとメロン博物館へ行きます.)
- 6JE-P23 Would you like to join us?  
(参加なさいますか.)
- ATR 6JE-K21 参加料は／いくらですか。  
(How much is the attendance fee?)
- CMU 6JE-P31 Thirty five dollars. That includes dinner.  
(35 ドルです. 夕食を含みます.)
- ATR 6JE-K31 わかりました。  
(I see.)
- 6JE-K32 それでは／参加したいと／思います。  
(Then, I'd like to attend.)
- CMU 6JE-P41 Okay please give me your name and the number of people in your party.  
(そうですか. 名前と全体の人数を教えてください.)
- ATR 6JE-K41 樽松明です。  
(I'm Akira Kurematsu.)
- 6JE-K42 家内と／参加します。  
(I'll attend with my wife.)
- CMU 6JE-P51 We'll meet in front of the reception desk.  
(受付の前で待ち合わせます.)
- 6JE-P52 Please pay the tour fee there when you arrive.  
(着いたときに, そこで観光代金を払ってください.)
- ATR 6JE-K51 わかりました。  
(I see.)
- 6JE-K52 ありがとうございます。  
(Thank you.)
- CMU 6JE-P61 We'll be expecting you.  
(楽しみにしています.)

## A.3 3GJ (会議のテーマについて)

Siemens = 問合せ者 vs. ATR = 会議事務局

(話者)	(発話番号)	
Siemens	3GJ-M11	hallo (もしもし.)
ATR	3GJ-K11	こちらは/会議事務局です。 (Hier ist das Konferenzsekretariat.)
Siemens	3GJ-M21	bitte sagen sie etwas ueber das thema der konferenz (会議の内容について教えてください.)
ATR	3GJ-K21	今回の/会議の/話題は/通訳電話です。 (Das Thema dieser Konferenz ist Telefondolmetschen.)
	3GJ-K22	広範な/研究分野を/含んでいます。 (Das umfasst ein weites Forschungsgebiet.)
Siemens	3GJ-M31	ich verstehe (わかりました.)
	3GJ-M32	uebrigens was ist auf dieser konferenz die offizielle sprache (ところで、会議での公式言語は何ですか.)
ATR	3GJ-K31	ドイツ語と/日本語です。 (Das ist Deutsch und Japanisch.)
Siemens	3GJ-M41	ich verstehe japanisch ueberhaupt nicht (わたしは日本語が全然わかりません.)
	3GJ-M42	gibt es eine simultanuebersetzung ins deutsche, wenn etwas auf japanisch praesentiert wird (発表が日本語で行なわれたら、ドイツ語への同時通訳はあるのですか.)
ATR	3GJ-K41	はい。 (Ja.)
	3GJ-K42	ドイツ語への/同時通訳を/用意しております。 (Wir bereiten eine Simultanuebersetzung ins Deutsche vor.)
Siemens	3GJ-M51	ich verstehe (わかりました.)
	3GJ-M52	vielen dank (どうもありがとうございました.)
	3GJ-M53	auf wiederhoeren (さようなら.)

## A.4 6GJ (市内観光)

ATR = 会議事務局 vs. Siemens = 問合せ者

- (話者) (発話番号)
- ATR 6GJ-K11 こちらは／会議事務局です。  
(Hier ist das Konferenzsekretariat.)
- Siemens 6GJ-M11 ich habe gehoert dass es eine stadtrundfahrt waehrend der konferenz gibt  
(会議の間に市内観光があるようですが.)
- 6GJ-M12 koennen wir noch teilnehmen  
(まだ参加できますか.)
- ATR 6GJ-K21 まだ／参加可能です。  
(Sie koennen noch teilnehmen.)
- 6GJ-K22 三月／五日に／金閣寺と／龍安寺を／見学します。  
(Am fuenften Maerz besichtigen wir den Kinnkahkuh-dschi-Tempel und den Ryoh-ahn-dschi-Tempel.)
- 6GJ-K23 参加なさいますか。  
(Nehmen Sie teil?)
- Siemens 6GJ-M21 wieviel betraegt die Teilnahmegebuehr  
(参加料はいくらですか.)
- ATR 6GJ-K31 七千円です。  
(Siebentausend Jenn betraegt es.)
- Siemens 6GJ-M31 ich verstehe  
(わかりました.)
- 6GJ-M32 also gut ich wuerde gerne teilnehmen  
(それでは参加したいのですが.)
- ATR 6GJ-K41 では／お名前と／人数を／お願いいたします。  
(Also gut, kann ich Ihren Namen und die Anzahl der Personen in Ihrer Gruppe haben, bitte?)
- Siemens 6GJ-M41 mein name ist harald hoege  
(名前はハラルド・ヘーゲです.)
- 6GJ-M42 ich werde mit meiner frau teilnehmen  
(家内と参加します.)
- ATR 6GJ-K51 集合場所は／受付の／前です。  
(Der Treffpunkt ist vor der Rezeption.)
- 6GJ-K52 参加料は／当日／お支払いください。  
(Bitte bezahlen Sie dann die Teilnahmegebuehr!)
- Siemens 6GJ-M51 ich verstehe  
(わかりました.)
- 6GJ-M52 vielen dank  
(ありがとうございました.)
- ATR 6GJ-K61 では／お待ちしております。  
(Also gut, wir erwarten Sie.)





## 付録 B

### CSTAR用 会話文 合成音声リスト

- 2JE\_K11 もしもし。  
2JE\_P11 こちらは、会議事務局です。  
2JE\_P12 どんなど用件でしょうか。  
2JE\_K21 会議の参加料について教えてください。  
2JE\_K22 現在、参加料はいくらですか。  
2JE\_P21 ええっとそうですねえ、一人あたり 220 ドルかかります。  
2JE\_P22 でも、もし来月申し込めば、325 ドルかかります。  
2JE\_K31 わたしは、情報処理学会の会員なのですが。  
2JE\_K32 参加料の割り引きはないのですか。  
2JE\_P31 いいえ、今回は、割り引きはありません。  
2JE\_K41 そうですか。  
2JE\_K42 参加料は、どのようにお支払いしたらよいのですか。  
2JE\_P41 案内に書いてある、わたしどもの銀行口座へ、送金してください。  
2JE\_P42 締切は年末です。  
2JE\_K51 わかりました。  
2JE\_K52 どうもありがとうございました。  
2JE\_P51 どういたしまして。  
2JE\_P52 失礼します。  
3GJ\_M21 会議の内容について教えてください。  
3GJ\_K21 今回の会議の話題は、通訳電話です。  
3GJ\_K22 広範な研究分野を含んでいます。  
3GJ\_M32 ところで、会議での公式言語はなんですか。  
3GJ\_K31 ドイツ語と日本語です。  
3GJ\_M41 わたしは日本語が全然わからないのですが。  
3GJ\_M42 発表が日本語で行なわれる場合、ドイツ語への同時通訳はあるのですか。  
3GJ\_K41 はい。  
3GJ\_K42 ドイツ語への同時通訳を用意しております。  
3GJ\_M53 さようなら。  
6GJ\_M11 会議の間に、市内観光があるようですが。  
6GJ\_M12 まだ、参加できますか。  
6GJ\_K21 まだ、参加可能です。  
6GJ\_K22 三月五日に、金閣寺と龍安寺を見学します。  
6GJ\_K23 参加なさいますか。  
6GJ\_M21 参加料は、いくらですか。  
6GJ\_K31 7000 円です。  
6GJ\_M32 それでは、参加したいと思います。

6GJ\_K41 では、お名前と人数をお願いいたします。  
6GJ\_M41 名前は、ハラルド ヘーゲです。  
6GJ\_M42 家内と、参加します。  
6GJ\_K51 集合場所は、受付の前です。  
6GJ\_K52 参加料は、当日お支払いください。  
6GJ\_M52 ありがとうございます。  
6GJ\_K61 では、お待ちしております。  
6JE\_P11 もしもし、こちらは会議事務局です。  
6JE\_K12 まだ、参加可能ですか。  
6JE\_P21 はい、できます。  
6JE\_P22 八月五日の午後に、ハイツホールと、マウントワシントンと、メロン博物館へ行きます。  
6JE\_P31 35 ドルです、夕食を含みます。  
6JE\_P41 そうですか、名前と全体の人数を教えてください。  
6JE\_K41 樽松 明です。  
6JE\_P51 受付の前で、待ち合わせます。  
6JE\_P52 着いた時に、そこで観光代金を払ってください。  
6JE\_P61 楽しみにしています。  
chk\_001 ありがとうございます。  
chk\_002 銀行送金で払ってください。  
chk\_003 まだ、参加できるのですか。  
chk\_004 もしもし、会議事務局です。  
chk\_005 もし来月申し込めば、325 ドルかかります。  
chk\_006 日本語で発表が行なわれたら、ドイツ語への同時通訳があるのですか。  
chk\_007 それでは、参加したいのですが。  
chk\_008 ところで、会議の公式言語はなんですか。  
chk\_009 予稿集と、レセプションが、申し込み費用に含まれています。  
chk\_010 ご連絡ください、もし、質問がありましたら、失礼します。  
chk\_011 もし、質問がありましたら、ご連絡ください、失礼します。  
chk\_012 会議の、公式言語はなんですか。  
chk\_013 名前は、ベルタ シュタングルです。  
chk\_014 日本語が、全然わかりません。  
chk\_015 もしもし、会議事務局です、ご用件はなんでしょうか。  
chk\_016 ええっとそうですねえ、一人あたり 250 ドルかかります。  
chk\_018 でも、もし来月申し込めば、350 ドルかかります。  
chk\_019 もしもし、会議事務局ですか。

- chk\_\* は、リハーサル中に受信したコーパスとマッチしなかった文である。

## 付録 C

### CSTAR 用 通信規約 (プロトコル)

(原案: Arthur E. McNair 一部修正: 谷戸 文廣)

#### C.1 メッセージ書式

メッセージは7ビットの ASCII 文字列から構成される。各メッセージは以下のフォーマットで構成される。

Innn_{text}	
I	識別子 (メッセージの種別を表示)
nnn	メッセージの文字数 (000-999)
-	ヌル (\0) セパレータ

#### C.2 メッセージ種別

?005_	相手側モデムプログラムの状態チェック
!005_	モデムプログラム動作中の返事
R005_	音声入力準備完了
B005_	音声入力開始
W005_	相手側システムの出力待ち
S013_Hi..Hi..	音声出力用文字列
P015_Bye..Bye..	音声認識結果文字列
X005_	伝送誤り、S または P メッセージの再送
F005_	直前の発話が翻訳失敗/取消
H005_	回線の接続終了
A005_	メッセージの受領確認
O005_	当方は事務局側
L005_	当方は参加者側
M008_02_	モデル会話 2 を実施
Z005_	発話の終わり、発話サイト交代
N005_	音声認識実行中
K005_	構文解析実行中
T005_	変換生成実行中

## C.3 メッセージ交換例

コンピュータ A が ATR で問い合わせをする利用者側で、対話 2 を選択した場合。

computer A:	<i>Call computer B via modem.</i>	
computer B:	<i>Picks up connection via modem.</i>	
computer A:	?005_	(Is modem program attached?)
computer B:	!005_	(Yes, modem program is attached.)
computer A:	L005_	(Computer A plays caller part.)
computer B:	A005_	(Okay.)
computer A:	M008_02_	(Use script 2.)
computer B:	A005_	(Okay.)
computer A:	R005_	(Ready for speech input.)
computer B:	A005_	(Okay.)
computer B:	R005_	(Ready for speech input.)
computer A:	A005_	(Okay.)
computer A:	B005_	(User has begun an utterance.)
computer B:	W005_	(Waiting for output from other site.)
computer A:	N005_	(Speech recognition in progress.)
computer B:	W005_	(Waiting.)
computer A:	P025_moshimoshi..moshimoshi..	(Utterance is "moshimoshi.")
computer B:	W005_	(Waiting.)
computer A:	K005_	(Parsing in progress.)
computer B:	W005_	(Waiting.)
computer A:	T005_	(Translation in progress.)
computer B:	W005_	(Waiting.)
computer A:	S019_Hello..Hello..	(Speak "Hello.")
computer B:	A005_	(Okay.)
computer A:	Z005_	(End of utterance, turn over.)
computer B:	A005_	(Okay.)
computer B:	B005_	(User has begun an utterance.)
computer A:	W005_	(Okay.)
	...	
computer A:	S015_Bye..Bye..	(Speak "Bye.")
computer B:	A005_	(Okay.)
computer A:	H005_	(Request to hang-up.)
computer B:	H005_	(Request to hang-up.)
computer A:	<i>Disconnects from modem, returns to wait for call.</i>	
computer B:	<i>Disconnects from modem, returns to wait for call.</i>	

## C.4 通信規則

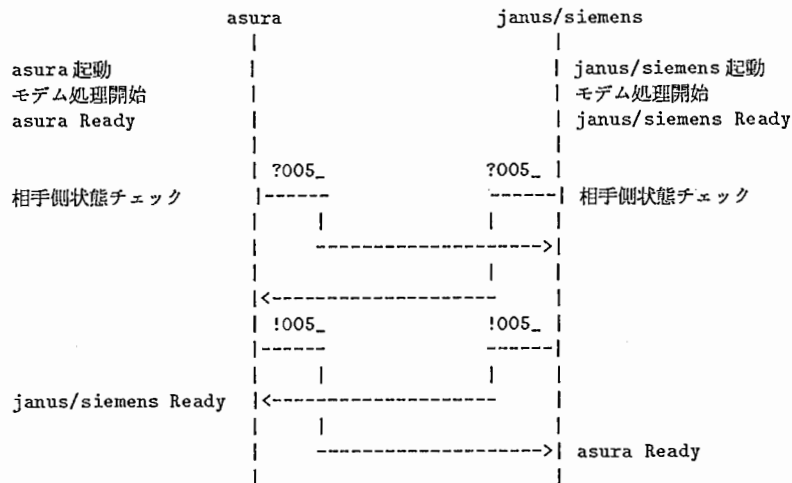
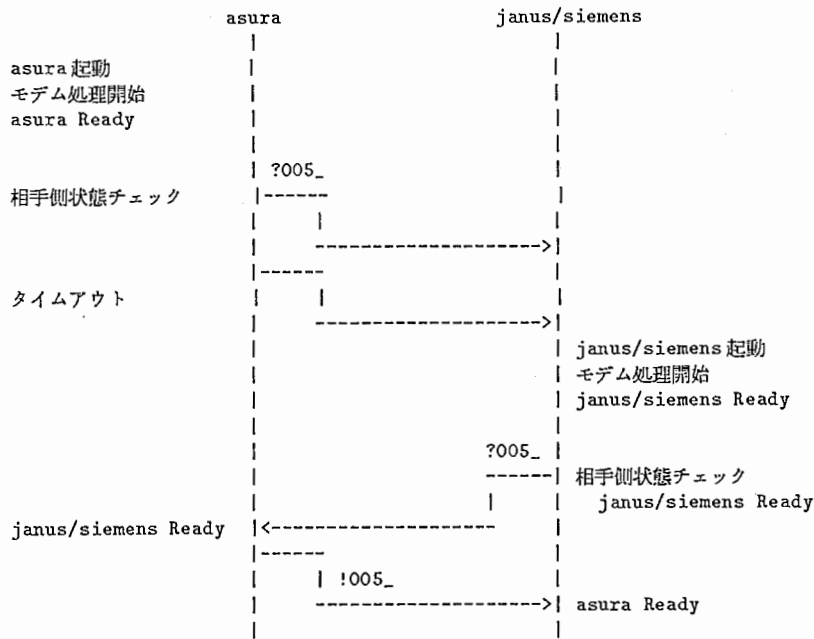
サイト間での通信は以下の規則に沿って行なう。

1. あるサイトからのすべてのメッセージは他のサイトの応答を得なければならない。
2. モデムプログラムに関する質問・応答のプロセスを最初に行なう。
3. ?メッセージに対して応答がない場合は!が得られるかタイムアウトするまで質問を繰り返す。
4. どちらのサイトからでもRメッセージは送出可能、ただし両サイトでのRおよびAメッセージ確認後に音声入力は許される。
5. 一旦あるサイトがBメッセージを送出したら、他のサイトはWメッセージで応答する。Zメッセージが送出された時点で音声入力可能なサイトは入れ替わる。サイトにおいて音声翻訳処理を実行中にはR, K, Tメッセージを送出して他のサイトに状況を知らせる(CMUの提案では3秒毎)。
6. もしFメッセージを受取ったら、Aメッセージで応答して相手の再入力を待つ。
7. 転送エラーが発生した場合、受信側は1秒間入力データを無視して待状態となった後、Xメッセージを相手方へ送る。エラーを含むメッセージを受信してXメッセージを送信した後、プログラムはエラーを含むメッセージを受信する以前の状態に戻る。
8. SおよびPメッセージでは同一テキストを繰返し送り、誤り検出に利用する。誤りが発生した場合は(g)の手続きを行う。
9. 対話過程においてあらかじめ定めた時間内にまったくメッセージが受信できない場合には、?メッセージを送出する。いかなる時点においても両サイトにおいて?メッセージに対して!メッセージを応答できるようにする。

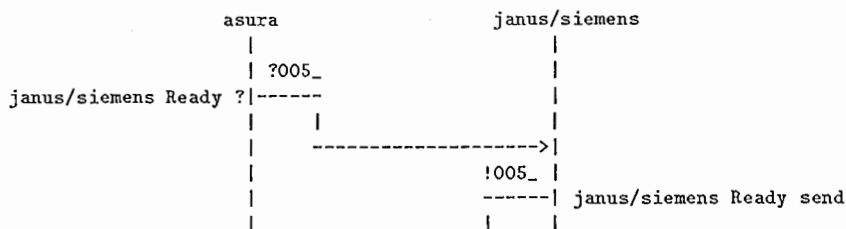
### C.5 タイミングチャート

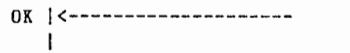
(作成: 石田 聡)

- ASURA-JANUS/Siemens 間の回線オープン
  - この処理が一番始めに行なわれなければならない。



- ASURA-JANUS/Siemens 間の回線チェック
  - 常時可能 (マニュアル)。
  - 音声入力 → 音声認識結果文字列送信の間は一定間隔以上時間が空いた時、自動的に送る (受信側、ASURA では未対応)。

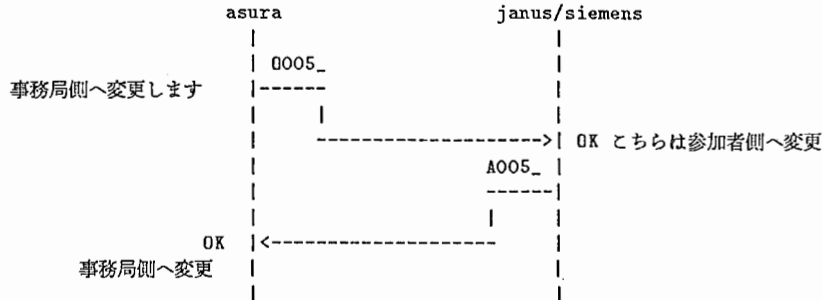




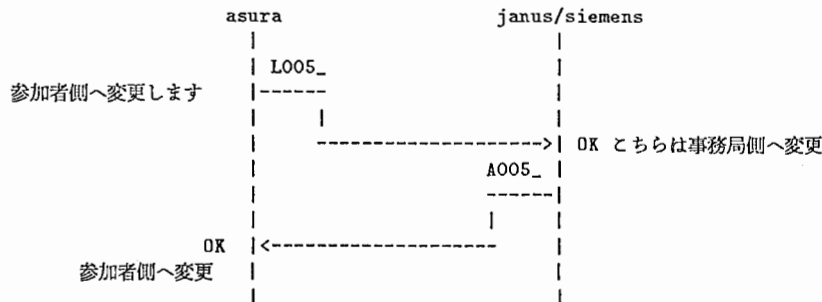
• 話者 (CALLER-OFFICE) の切替え要求、応答

- 話者切替えがあった場合、受領確認がありしだいモデル会話の切替え要求を送ること。

事務局側への変更時

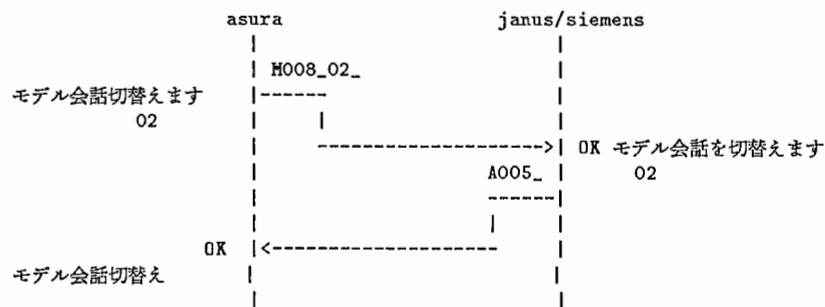


参加者側への変更時

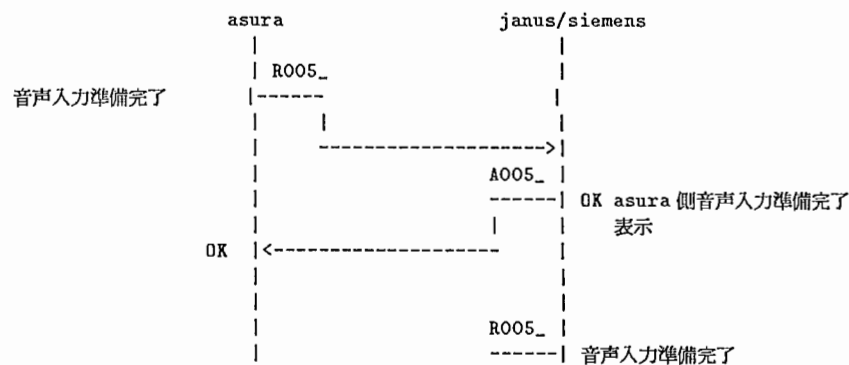


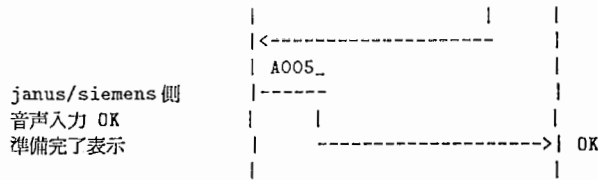
• モデル会話の切替え要求、応答

- モデル会話の切替え時、話者の切替えが必要な場合、先に話者切替えを送ること。
- 受信の場合、既にモデル会話を選択されていればモデル会話の切替えが発生します。



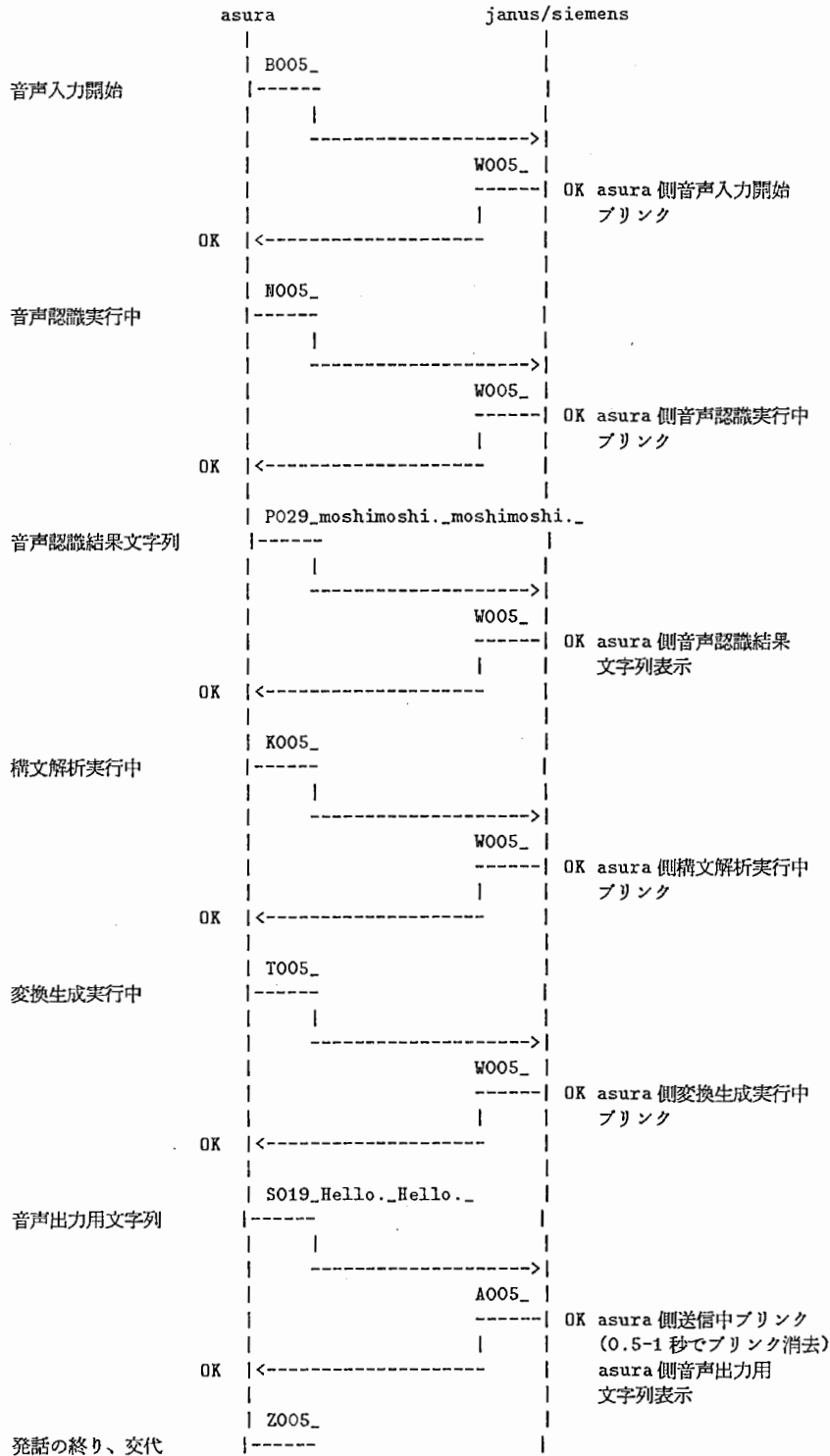
• 音声準備完了



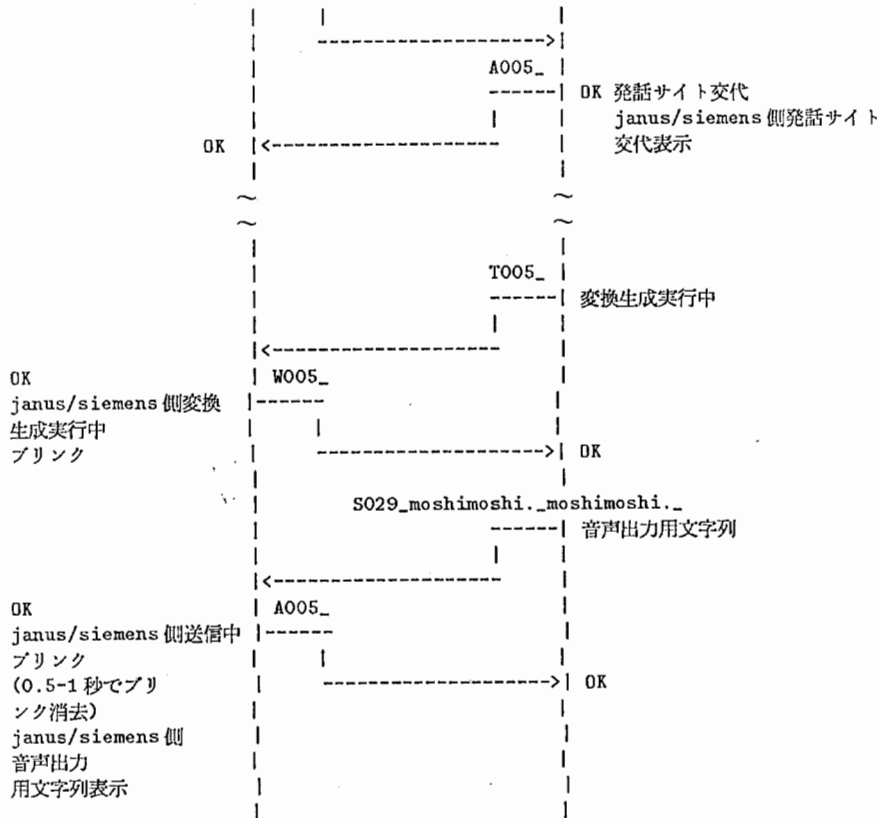


• 状態の送信 (入力中、翻訳中など)

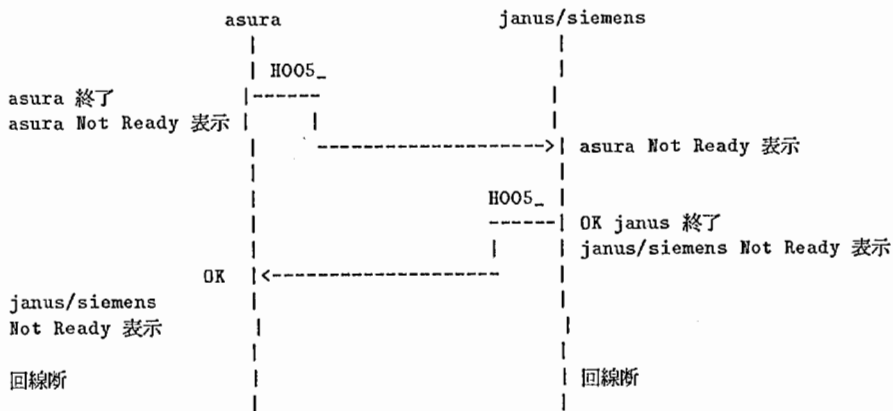
- 音声認識が失敗した場合、音声認識実行中から音声入力開始へメッセージが戻る場合があります。





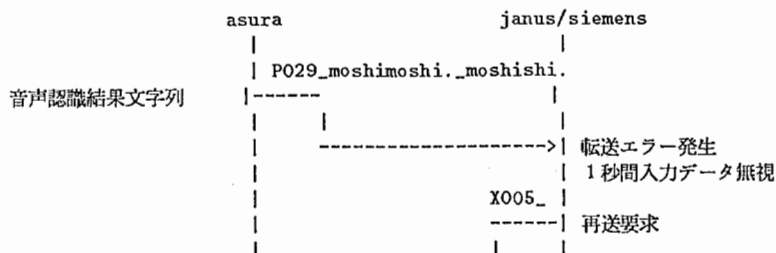


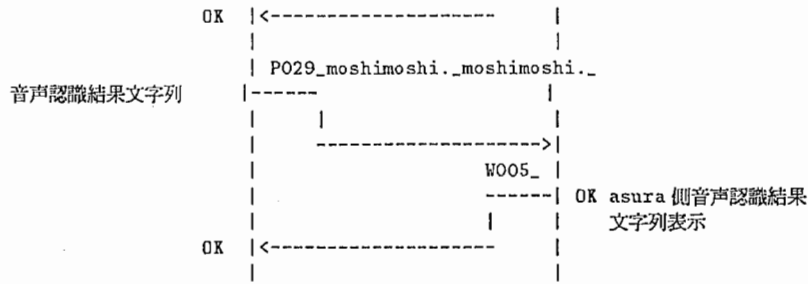
• ASURA-JANUS/Siemens 間の回線クローズ



• 転送エラー発生時の処理

- S または P メッセージを受信した時、テキストチェック。異常なら再送要求。
- 識別子異常の時も再送要求。
- 転送エラー時、データを読みとばす時間は可変にすること。
- 指定のバイト数分、データがこないときのタイムアウト時間も設定しておくこと。





● 翻訳失敗時の処理

- ・ 翻訳失敗時には翻訳結果の欄に“翻訳失敗”を表示後、画面のブリンクを消去。

