概念からの対話音声合成と
合成ルールベース構築支援に関する研究

１９９３年３月

山　下　洋　一

概念からの対話音声合成と
合成ルールベース構築支援に関する研究

１９９３年３月

山　下　洋　一

内容梗概

本論文は，筆者が大阪大学産業科学研究所 溝口理一郎 教授の指導のもとに行なった研究の内，概念からの対話音声合成と合成ルールベース構築支援に関する研究の成果をまとめたものであり，次の６章をもって構成されている．

第１章序論では，本研究の社会的背景について概説し，従来の音声合成の研究の問題点を指摘すると共に，本研究の目的と工学的意義について述べている．

第２章では，まず音声合成の研究を出力内容の多様性の観点から分類し，特に規則合成方式について，合成単位，合成パラメータ，入力データについて述べている．また，音声対話システム構築の立場から，情報伝達媒体としての音声の長所・短所についても考察している．規則合成方式では合成音の品質を向上させるために支援システムが重要となることを指摘し，対話システム構築における音声合成の意義についても述べている．

第３章では，音声規則合成のためのルールベース構築支援環境 SED-SSRB (a Support Environment for Development of the Speech Synthesis Rule Base) について述べている．まず，帰納的学習に基づいて多数の音声データから合成規則を自動抽出する手法を示している．それぞれの合成パラメータについて，規則によって決定された値と人間が発声した音声データの分析値との差の記述を例題として用い，音声学に関する専門知識に基づいて条件部を一般化・特殊化することによって規則の自動抽出が行なわれる．同様の手法に基づいて合成規則の修正案の提示も行なわれる．音節継続時間長に関する規則を対象として合成規則の自動抽出・修正の実験を行ない，これらの手法の有効性を示している．さらに，規則インタプリタに基づいた音声合成システム SSRI (Speech Synthesis system based on Rule Interpreter) を開発している．音声合成時に発火した規則を入力ごとに管理することによって，ルールベース修正後に同じ入力に対して再び合成を行なう場合，処理時間を大幅に短縮できる．SSRI を用いることによって比較的短時間で有効なルールベースの修正が行なえることを実験によって検証している．

第４章では，計算機上に実現された種々の問題解決器から自然で高品質な対話

音声出力を実現するために，汎用音声出力インタフェースの枠組を提案すると共に，インタフェースへの入力である概念表現を音声に変換する概念表現からの音声合成システム SOCS (Speech Output from Case Structure representation) の基本アーキテクチャについて述べている．問題解決器・音声合成システム・対話管理部の観点からインタフェースへの入力として最適な表現形式を検討し，格構造に基づいた表現と文のパタンを組合わせた表現を概念表現として定義している．さらに，SOCS における韻律生成方式について述べている．概念表現中に記述される韻律オペレータ，文生成時に出力されるポーズマーカー，慣用テンプレート中の韻律修正関数によって，ポーズや基本周波数などの韻律的特徴が決定される．

第5章では，概念表現を用いた対話音声合成における対話管理について述べている．SOCS に基づいた対話音声合成の枠組では，問題解決器は出力の意味内容だけを決定し，汎用音声出力インタフェースにおける対話管理部が対話コンテキストを考慮して最適な音声言語表現を決定する．対話管理部では，SR-プランと TPN という汎用なモデルによって対話構造が表現される．特に TPN に関しては，収録された模擬対話を用いて話題の遷移を調べ，このモデルの妥当性を検証している．対話管理部における対話処理は，この二つの対話構造モデルと SOCS におけるテンプレートなどに関する知識を利用しながら行なわれる．具体的な対話処理として，強調語句の抽出，入力音声の認識結果の提示，慣用テンプレートへの書き換え，韻律オペレータの追加の例を挙げ，概念表現を用いることにより対話処理が容易に行なえることを示している．

第6章では，本研究で得られた主な成果をまとめ，今後に残された課題について述べている．

発表論文

A. 学会誌掲載論文

1. Masuzo Yanagida, Yoichi Yamashita, Osamu Kakusho and Donatus Graham Stuart : "Some Properties of the Givens' Reduction Applied to Linear Prediction Analysis of Speech", the Journal of the Acoustical Society of Japan (E), Vol.7, No.1, pp.47-56 (1986).

2. 山下洋一, 柳田益造, 溝口理一郎, 角所収 : "音響パラメータの回帰直線近似を用いた有声破裂音の識別", 電子通信学会論文誌, Vol.J69-A, No.2, pp.282-290 (1986).

3. 山下洋一, 柳田益造, 溝口理一郎, 角所収 : "音響パラメータの回帰直線により抽出された動的及び瞬時的特徴を用いた無声破裂音の識別", 電子情報通信学会論文誌, Vol.J70-A, No.1, pp.132-134 (1987).

4. Masuzo Yanagida, Yoichi Yamashita and Osamu Kakusho : "Detection and Identification of Plosive Sounds in Japanese Words", the Journal of the Institute of Electronics and Telecommunication Engineers, Vol.34, No.1, pp.82-87 (1988).

5. 山下洋一, 谷口賢一, 溝口理一郎, 柳田益造, 角所収 : "音声規則合成におけるルールベース構築支援システム --合成規則の自動抽出--", 電子情報通信学会論文誌, Vol.J71-D, No.6. pp.1012-1019 (1988).

6. 荒井和博, 鬼山康人, 野村康雄, 山下洋一, 北橋忠宏, 溝口理一郎 : "知識処理に基づく音声自動ラベリングシステム", 電子情報通信学会論文誌, Vol.J74-D-II, No.2, pp.130-141 (1991).

7. Tetsuya Yamamoto, Yoshikazu Ohta, Yoichi Yamashita, Osamu Kakusho and Riichiro Mizoguchi : "MASCOTS: Dialog Management System for Speech Understanding System", IEICE Transactions, Vol.E74, No.7, pp.1881-1888 (1991).

8. 荒井和博, 山下洋一, 堀尾裕幸, 村上雅義, 千葉喜英, 角所収, 溝口理一郎 : "信号解釈エキスパートシステムの開発", 電子情報通信学会論文誌, Vol.J75-D-II, No.5, pp.917-926 (1992).

9. Yoichi Yamashita, Hideaki Yoshida, Takashi Hiramatsu, Yasuo Nomura and Riichiro Mizoguchi : "MASCOTS II: A Dialog Manager in General Interface for Speech Input and Output", IEICE Transactions on Information and Systems, Vol.E76-D, No.1, pp.74-83 (1993).

10. 山下洋一, 水谷直樹, 角所収, 溝口理一郎 : "汎用音声出力インタフェースにおける概念表現からの音声合成", 電子情報通信学会論文誌 (1993年3月掲載予定).

B. 国際会議等発表論文

1. Yoichi Yamashita, Masuzo Yanagida, Riichiro Mizoguchi and Osamu Kakusho : "Discrimination of Voiced Plosives Based on Transition Characteristics of LPC Cepstrum Parameters", Proceedings of Second Western Pacific Regional Acoustics Conference, pp.348-353 (1985).

2. Yoichi Yamashita, Masuzo Yanagida, Riichiro Mizoguchi and Osamu Kakusho : "Discrimination of Voiced Plosives Using Transition Properties of the LPC Cepstrum Parameters", Proceedings of Montreal Symposium on Speech Recognition, pp.103-104 (1986).

3. Yoichi Yamashita, Riichiro Mizoguchi and Osamu Kakusho : "Quantitative Evaluation of the Perceptual Significance of Control Parameters in Synthesis by Rule", Second Joint Meeting of the Acoustical Society of America and the Acoustical Society of Japan, S24 (1988).

4. Yoichi Yamashita, Naoki Mizutani and Riichiro Mizoguchi : "Concept Description for Synthetic Speech Output System", Proceedings of the European Speech Communication Association Workshop on Speech Synthesis, pp.241-244 (1990).

5. Yoichi Yamashita, Hiroyuki Fujiwara, Yasuo Nomura, Nobuyoshi Kaiki and Riichiro Mizoguchi : "A Support Environment Based on Rule Interpreter for Synthesis by Rule", Proceedings of 1990 International Conference on Spoken Language Processing, pp.769-772 (1990).

6. Kazuhiro Arai, Yoichi Yamashita, Tadahiro Kitahashi and Riichiro Mizoguchi : "A Speech Labeling System Based on Knowledge Processing", Proceedings of 1990 International Conference on Spoken Language Processing, pp.1005-1008 (1990).

7. Tetsuya Yamamoto, Yoshikazu Ohta, Yoichi Yamashita and Riichiro Mizoguchi : "Dialog Management System MASCOTS in Speech Understanding System", Proceedings of 1990 International Conference on Spoken Language Processing, pp.1301-1304 (1990).

8. Kazuhiro Arai, Yoichi Yamashita and Riichiro Mizoguchi : "A Shell for Signal Interpretation Expert Systems", Proceedings of the World Congress on Expert Systems, pp.1180-1187 (1991).

9. Yoichi Yamashita and Riichiro Mizoguchi : "SOCS: A Speech Output System from Concept Representation", Proceedings of 1992 International Conference on Acoustics, Speech, and Signal Processing, 2, pp.69-72 (1992).

10. Yoichi Yamashita, Naoki Mizutani and Riichiro Mizoguchi : "Concept Representation for Synthetic Speech Output System", Talking Machine: Theories, Models, and Designs, North-Holland, pp.413-427 (1992).

11. Yoichi Yamashita and Riichiro Mizoguchi : "Dialog Management for Speech Output System from Concept Representation", Proceedings of 1992 International Conference on Spoken Language Processing, pp.1435-1438 (1992).

12. Shingo Nishioka, Yoichi Yamashita and Riichiro Mizoguchi : "A Powerful Disambiguating Mechanism for Speech Understanding Systems Based on ATMS", Proceedings of 1992 International Conference on Spoken Language Processing, pp.1641-1644 (1992).

C. 研究会発表論文

1. 山下洋一, 溝口理一郎, 柳田益造, 谷口賢一, 鬼頭淳悟, 角所収 : "規則合成におけるルール抽出の支援", 電子情報通信学会技術研究報告, SP87-5, pp.33-40 (1987).

2. 澤多靖浩, 山下洋一, 鬼頭淳悟, 溝口理一郎, 角所収 : "規則規則抽出支援システムにおける規則の自動抽出", 電子情報通信学会技術研究報告, SP87-113, pp.1-8 (1988).

3. 山下洋一, 水谷直樹, 溝口理一郎 : "合成音出力における概念表現の利用", 電子情報通信学会技術研究報告, SP89-115, pp.41-48 (1990).

4. 藤原宏之, 野村康雄, 海木延佳, 鬼頭淳悟, 山下洋一, 溝口理一郎 : "ホルマントを用いる音声規則合成のためのルールインタプリタの開発", 電子情報通信学会技術研究報告, SP90-29, pp.9-16 (1990).

5. 山下洋一, 溝口理一郎 : "対話管理における次発話の予測に関して", 電子情報通信学会第二種研究会資料, SPREC-91-1, pp.63-66 (1991).

6. 柴田宜宏, 山下洋一, 溝口理一郎 : "合成音声出力における対話管理について", 電子情報通信学会技術研究報告, SP91-109, pp.9-16 (1992).

7. 吉田英昭, 平松敬史, 野村康雄, 山下洋一, 溝口理一郎 : "機械への音声入力のための汎用対話管理システム", 人工知能学会言語・音声理解と対話処理研究会資料, SIG-SLUD-9202-9, pp.77-85 (1992).

8. 平松敬史, 吉田英昭, 野村康雄, 山下洋一, 溝口理一郎 : "音声対話理解のための話題知識の利用", 電子情報通信学会技術研究報告, SP92-110, pp.55-62 (1992).

目次

第１章　序論 1

第２章　音声合成と音声対話 5

2.1　緒言 5
2.2　音声合成方式 5
2.3　規則合成方式による音声合成 7
2.3.1　合成単位 7
2.3.2　合成パラメータ 8
2.3.3　入力データ 9
2.4　情報伝達媒体としての音声 9
2.5　結言 10

第３章　音声規則合成のためのルールベース構築支援 11

3.1　緒言 11
3.2　ルールベース構築支援 12
3.3　合成規則の自動抽出 15
3.3.1　比較例の生成 16
3.3.2　専門知識の記述 18
3.3.3　規則抽出アルゴリズム 20
3.3.4　抽出された合成規則の評価 27
3.4　合成規則修正案の提示 30
3.4.1　修正案の自動生成 31
3.4.2　修正案の順序付け 34

3.4.3 音声合成ルールベースの修正例 34
3.5 規則インタプリタに基づいた音声合成システム 37
3.5.1 データの形式 38
3.5.2 合成規則管理部 44
3.5.3 SSRI の評価 46
3.6 結言 49

第 4 章 概念表現からの音声合成 51

4.1 緒言 51
4.2 音声出力インタフェースの設計 52
4.2.1 音声出力インタフェースの設計思想 52
4.2.2 音声出力インタフェースへの入力表現形式 55
4.3 概念表現 60
4.3.1 概念表現を利用した音声出力インタフェース 60
4.3.2 概念表現の構成要素 61
4.3.3 概念表現例 65
4.4 SOCSシステム 67
4.4.1 文生成 67
4.4.2 韻律生成 69
4.5 結言 74

第 5 章 対話音声合成における対話管理 75

5.1 緒言 75
5.2 対話構造モデル 76
5.2.1 SP-プラン 77
5.2.2 TPN 78
5.3 模擬対話における話題の遷移 83
5.3.1 模擬対話の収録 83
5.3.2 模擬対話の分析 84
5.4 概念表現に基づいた対話処理 86

5.4.1 強調語句の抽出 86
5.4.2 入力音声の認識結果の提示 90
5.4.3 慣用テンプレートへの書き換え 93
5.4.4 韻律オペレータの追加 94
5.5 結言 94

第 6 章 結論 97

謝 辞 101

参考文献 103

第１章

序論

近年，計算機技術が急速に発達し，情報提供，教育，設計など種々の問題解決，あるいはその支援が計算機を用いて行なわれるようになってきた．これに伴い，これら計算機上に実現された問題解決器 (エキスパートシステム，情報の収集や検索，教育などにおいて計算機を利用して問題解決を行なうシステムを本論文では問題解決器と呼ぶ) と利用者である人間とのコミュニケーション手段の高度化が必要となってきている．現在，計算機とのコミュニケーション手段としては，キーボードやボタンを用いた入力やディスプレイなどへの文字による出力が主に用いられているが，通常人間同士では音声によって自然にコミュニケーションを行なっていることなどから，媒体として音声を用いたインタフェース技術に期待が高まっている．計算機に代表される機械と人間の間で音声による対話を実現するには，音声認識・音声合成・対話管理など個々の要素技術を高度化し，さらにはそれらを統合化することが不可欠となる．現在，音声認識に関しては，少数の語彙に対する単語音声認識が一部実用化され，連続音声への対応が急務となっている．一方，音声合成では，漢字仮名交じり文を音声に変換する音声合成ボードが市販されており[中津90]，電話での応答サービスや新聞記事の校閲支援[河合87]などではすでに実用化されている．しかしながら，このような音声合成技術に関しても，合成音の品質はまだ満足できるものではなく改善の要求が強い．また，利用形態も音声による対話を指向したものではなく，一方向の情報伝

達のために合成音が使われている．

一般に，機械から音声を作り出す技術を広く音声合成と呼んでいる．機械が音声を作り出すために内部に保持すべきデータや，出力される音声の内容を決定するために機械に外部から与えられるデータには様々な形態があり，それらの組合せによっても数多くの音声合成方式が存在するが[斎藤81]，現在では，出力可能な文に制限のない規則合成方式を中心として研究が行なわれている．この中で，ホルマントを用いるパラメータ導出型の規則合成方式，いわゆるホルマント合成[Klatt80]は，音韻接続部でのパラメータの操作が容易であることや，多言語の音声合成に対応しやすいなどの利点を有している反面[樋口89]，音声合成に必要な全ての音響パラメータを制御するために非常に多くの規則が必要となるという問題がある．ホルマント合成を始めとして規則合成方式の音声合成では，人間が音声を発声する時に行なっている種々の制御を合成規則によって計算機上で実現しており，合成音の品質をさらに向上させるためには，合成規則をよりいっそう充実させることが必要となる．このような合成規則は従来から専門家が試行錯誤によって作成してきたが，専門家の負担を軽減し有効な合成規則を効率良く生成するためには，合成規則の集まりを一つのルールベースとして捉え，計算機を用いてルールベースの構築支援を行うことが必要と考えられる．

また，従来の音声合成の研究においては，音韻的および韻律的特徴をいかにして自然音声に近づけるかという問題に研究の主眼が置かれており，音声合成システムへの入力がどのような形式で与えられるべきかという問題についてはほとんど検討されていない．しかしながら，音声合成の応用として問題解決器からの自然で高品質な音声出力の実現を考える時には，問題解決器と音声合成システムという二つのシステム間の相互作用を考慮したシステム論的立場から対話音声生成を検討することも重要となる．規則合成方式の音声合成システムは，漢字仮名混じり文などの自然言語表現によるテキストが入力として与えられることがあり，特にテキスト合成と呼ばれている．このテキスト合成システムと自然言語出力機能を持つ問題解決器を接続し構成された音声出力システムは，最も工学的に安価な接続方式であるということができ，システム論的観点からは考察の余地が残さ

れている．問題解決器と音声合成システムとの情報伝達形式を核にして，システム全体の機能分担を総合的に再検討することより，音声合成システムへの入力記述形式として最適な表現を決定すべきである．さらに，このような音声出力システムを効率良く構築できるようにするためには，その中で共通して行なわれる処理を汎用モジュールとして構築することも非常に重要と考えられる．

以上のような立場から，問題解決器からの音声出力を高度化するための音声出力インタフェースの構築を目指して，音声合成ルールベース構築支援および概念表現を用いた対話音声合成に関する研究を行なった．

第2章では，これまでに行なわれた音声合成の研究を概説すると共に，情報伝達媒体としての音声の長所・短所について考察し，本研究の意義を明らかにする．

第 3 章では，音声規則合成のためのルールベース構築支援環境 SED-SSRB (a Support Environment for Development of the Speech Synthesis Rule Base) について述べ，合成規則の集まりを合成ルールベースとして独立した一つのモジュールとして捉え，計算機によってその構築支援を行なうことによって効率良く合成規則の抽出・修正が行なえることを示す．支援機能として，多数の音声データから帰納的学習に基づいて行なう合成規則の自動抽出，合成規則を修正する時の修正案の提示，規則インタプリタに基づいた音声合成システム SSRI (Speech Synthesis system based on Rule Interpreter) が提供される．SSRI では，音声合成時に発火した規則を入力ごとに管理することによって，ルールベース修正後の再合成の処理時間が大幅に短縮される．

第4章では，計算機上に実現された種々の問題解決器から自然で高品質な対話音声出力を実現するために，汎用音声出力インタフェースの枠組を提案し，インタフェースへの入力である概念表現を音声に変換する概念表現からの音声合成システム SOCS (Speech Output from Case Structure representation) の基本アーキテクチャについて述べる．問題解決器・音声合成システム・対話管理部の観点からインタフェースへの入力として最適な表現形式を検討し，格構造に基づいた表現と

文のパタンを組合わせた表現を概念表現として定義した．SOCSにおける韻律生成では，概念表現中の韻律オペレータ，文生成時に出力されるポーズマーカー，慣用テンプレート中の韻律修正関数によって韻律的特徴が決定される．

第5章では，概念表現を用いた対話音声合成における対話管理について述べる．対話管理部は，対話構造モデルとSOCSにおけるテンプレートなどに関する知識を利用しながら，問題解決器の生成した概念表現を修正し，対話コンテキストに合った音声言語表現を決定する．対話管理部を汎用なものとして設計するために，対話に一般的に存在する発話対と対話セグメントに注目し，それぞれSR-プランとTPNによってモデル化すると共に，収録された模擬対話を用いて話題の遷移を調べTPNの妥当性を検証した．さらに，具体的な対話処理の例を挙げ，概念表現を用いることにより対話処理が容易に行なえることを示している．

最後に，第6章では本研究を総括する．

第2章

音声合成と音声対話

2.1 緒言

計算機に代表される機械から人間が音声によって情報を受けとることができるようにするには，機械の内部情報を音声に変換する技術，すなわち音声合成が必要になる．音声合成技術は，音声生成過程のモデル化，特徴パラメータの接続，合成単位の設定，韻律生成，合成規則の生成，テキスト解析など広範な要素技術の集大成として実現される．本章では，これまでに行なわれた音声合成の研究を出力内容の多様性の観点から分類し，特に規則合成方式について概説する．また，音声対話システムを構築する上で情報伝達媒体としての音声の長所・短所についても考察し，対話システム構築における音声合成の意義についても述べる．

2.2 音声合成方式

これまでに，音声合成を行なう方式として様々な方式が提案されてきている[大泉72][斎藤81][Klatt87]．これを出力内容の多様性の観点から分類すると表2.1のように3種類に分類できる．

分析合成方式では，予め人が発声した内容を何らかのデータ形式で蓄えておき，それを音声波形に変換することによって音声合成を行なう．データの蓄積方

表2.1　音声合成方式の分類

音声出力方式	出力可能な内容	長所・短所
分析合成方式	登録されている内容のみ	・音質が良い. ・出力内容を予め発声しておく必要がある.
編集合成方式	限られた単語や文パタンの組み合わせ	・音質が比較的良い. ・韻律の制御が難しい.
規則合成方式	任意の文	・出力内容に制限がない. ・音質がやや劣る. ・合成規則の作成が困難.

法としては，PCM (Pulse Code Modulation) に代表されるような音声波形に限らず一般の波形データに適用可能な波形の符合化による方式と，線形予測分析などのように音声波形の特徴に基づいてスペクトルの符合化を行なう方式がある．分析合成方式では予め出力内容を収録しておかねばならないため，多様な内容の出力には不向きであるが，反面，データの接続・加工などが必要ないため音質が良いことが利点として挙げられる．出力内容が予め少数に限定できるような場合には有効な音声合成手法であり，種々の機械からの "お知らせ" や "警告" のメッセージなどで利用されている．

編集合成方式では，限られた数の単語・文節などのデータを組み合わせて接続することによって文などを構成し，音声に変換する．この方式においても使用する単語などを予め発声しておく必要があるが，組合せを行なう分，分析合成方式に比べて出力内容の多様性が増す．反面，データの接続を行なうため，接続部での不連続性や全体としての韻律の不自然さなどが生じることになり，これに伴う音質劣化を抑える工夫が必要となる．この方式の応用では，多様な文を出力する必要があるものの使用する語彙が限定されているような，例えば駅での列車の発車時刻のアナウンスなどが挙げられる．

以上の二つの合成方式では，出力内容自体や使用する語彙を予め人が発声して

おく必要があることから出力内容が制限されてしまうが，規則合成方式では出力内容に応じた音声を規則によって合成するため任意の内容を出力できる．そのために，人間が音声を発声する時に行なう種々の制御を計算機上で多数の合成規則によって実現する．この方式は，出力可能な文に制限がないことから機械との対話や自動翻訳電話といった幅広い対象に応用することが期待でき，現在精力的に研究が行なわれている[広瀬92].

2.3 規則合成方式による音声合成

規則合成方式による音声合成は，出力内容を表現した離散的な情報を音声波形に変換する技術である．この方式はさらに，合成に用いられる音声単位 (合成単位)，合成システム内部に持つデータを表現するパラメータ (合成パラメータ)，合成時に外部から与えられる入力データなどの観点から分類できる．

2.3.1 合成単位

規則合成における合成単位としては，音素，音節，半音節 (demi-syllable)，二音素連鎖 (dyad, diphone)，さらにはVCV，CVCなどの複合単位などが用いられる．音素は最も基本的な単位であり，用意する単位の数も少なくて済むが，音韻変形のための多くの規則が必要になる．他の合成単位では，音素間の調音結合の情報が合成単位内に含まれ明瞭性は向上するが，単位の種類が多くなる．音節の種類は日本語では100程度であるが，英語などでは1万近くにも及ぶ．このように，合成単位の選択は言語にも依存した問題である．

一般に連続音声では，調音結合などの影響で同じ音素でも音韻環境の違いによって微妙に特徴が異なってくる．このため，音声認識においてこのような音素の変化を吸収することが大きな課題となるように，逆に音声合成ではいかにしてこのような音素の変化をつくり出すかが大きな問題となり，これによって音質が決定される．従来は合成単位のデータを一つずつ用意しておき，それをそのまま，あるいは変形して用いることが多く行なわれてきたが，近年，計算機の発達

に伴い大量のデータを容易に利用できるようになったことから，多数の合成単位を用いる手法が試みられている．固定した合成単位に対して数多くのバリエーションを選択的に用いる手法として，合成時に音素ごとにデータベース (波形辞書) 中の最適な素片を検索する手法[広川88]，予めクラスタリングによって合成単位の代表的なバリエーションを求めておく手法[中嶌89]などが提案されている．また，必要な調音結合をできるだけ含んだ合成単位を用いるために，データベース中の音声データから様々な長さの合成単位を適応的に用いる手法[武田90]も行なわれている．

2.3.2 合成パラメータ

合成単位を表現するパラメータとしては，波形[広川88]，PARCOR・LSP・ケプストラムといった線形予測分析に基づくパラメータ[佐藤78][管村81][阿部81]，声道断面積や調音パラメータなどの生理学的なパラメータ，声道の共振周波数であるホルマント[Klatt80][樋口89]など，種々の特徴量が用いられてきた．波形をそのまま用いる方法では，合成素片に対する韻律的な特徴の操作を行なうことが難しく，生理学的なパラメータは実測値が得にくいことなどから，線形予測分析に基づくパラメータやホルマントが用いられることが多い．線形予測分析に基づいたパラメータは，音源の情報が分離されているため波形などに比べると規則による接続や韻律的特徴の設定が自由に行なえるが，スペクトルなどの物理量と対応がとりにくいため規則によってパラメータ値を変更することが非常に難しい．また，ホルマントは共振特性を周波数と帯域幅によって直接表現しているため，その値の解釈が容易に行なえ規則化に向いたパラメータであるが，音声波形の分析によって正確なホルマントを求めることが難しいため，規則化においては専門家の試行錯誤が必要となる．

規則合成方式では，人間が音声を発声する時に行なっている種々の制御を合成規則によって計算機上で実現する．このためには多数の合成規則が必要であり，合成音の品質を向上させるために，支援システムなどを利用し規則の生成・修正を効率良く行なうことが重要となる．

2.3.3 入力データ

規則合成システムでは，最終的には合成部へ音韻記号や韻律記号が与えられるが，その前の段階として自然言語表現されたテキストを入力として用いることがあり，特にこれを「テキストからの音声合成 (TTS: Text-to-speech)」と呼ぶ[Allen87]．また，自然言語表現以外の意味表現が用いられることもあり，概念からの音声合成と呼ばれる[Young79]．(本論文では，概念からの音声合成を CTS (Concept-to-speech) と呼ぶことにする．)

TTS では，与えられた自然言語表現を音韻記号列などに変換するために，構文解析，単語同定，読み・アクセント型の同定などテキスト解析の処理が必要になる．これらの処理は必ずしも容易ではなく，その解析誤りが合成音に大きな劣化を生じさせることから，テキスト解析も音声合成における重要な研究課題となっている．一方，CTS では，テキスト解析の代わりに文生成を行なわなければならないが，構文情報を文生成時に自ら決定するため誤りのない係り受け構造を利用できるなど，韻律生成を容易かつ的確に行なうことができる．これまでに行なわれた CTS の研究は非常に少ないが，4 章でも述べるように，機械との音声対話における音声出力の手段として非常に重要な技術となる．

2.4 情報伝達媒体としての音声

人間同士のコミュニケーションにおいて通常音声が用いられるように，音声は人間にとって最も自然で便利な情報伝達媒体であり，機械とのコミュニケーションにおいても音声を用いたいと考えることは当然の要求である．このような機械との音声対話を実現するためには，音声認識と並んで音声合成をインタフェース技術として確立する必要がある．ここで，機械からの出力に音声を用いる場合の長所，短所について考えてみる．まず長所としては，

- ほとんどの人が努力することなく理解できる．
- 離れた場所へも容易に情報を送れる．

・情報を受けとるために手，足，視覚などの自由を奪われない．

・聞き手が特に道具を必要とせず情報を受けとれる．

・一度に多数の聞き手に伝えられる．

などが挙げられ，逆に短所としては，

・テキストなどに比べると転送，蓄積にデータ量が多くなる．

・文字を読む場合に比べて，理解するのに時間がかかる．

・一過性であるため，早送りや巻き戻しが困難である．

などが指摘できる[Streeter88]．

テキストの表示などに比べて音声が劣る部分はあるものの，マルチメディア出力などにおいても音声出力の果たす役割は大きく，機械との対話システムを構築する上で対話を指向した音声合成の研究が重要となる．

2.5　結言

音声研究の目的の一つとして音声対話システムの構築が挙げられる．

種々の問題解決において音声対話を実現するには，任意の内容が出力可能な規則合成方式の開発が不可欠である．現在では，規則合成方式による合成音もかなりの明瞭性が確保されているものの自然性・聞きやすさなどにおいてはまだ不十分で，一般に受け入れられるレベルに達しているとは言えない．さらに合成音の品質改善を行なうには，新しい規則合成手法の開発や規則の洗練が必要となるが，合成音の品質改善は基本的に試行錯誤を伴う非常に"泥くさい"作業となることから，計算機などを用いた何らかの支援が望まれる．

また，マンマシンインタフェースの高度化に対しては社会的要求も大きく，音声を用いたコミュニケーションにも期待が寄せられている．このような社会的背景から音声情報処理に基づくインタフェース技術の確立も重要な研究課題となっている．

第3章

音声規則合成のためのルールベース構築支援

3.1 緒言

音声合成に対する社会的要求の増加にともない，任意の文章を音声で生成できる規則合成方式の音声合成への期待が高まっている．これまでに日本語の規則合成に関して多くの研究が行われてきたが[匂坂85][石井86]，規則合成方式による合成音の品質はまだ満足できるものとは言えない．規則合成方式の音声合成では，音韻性，韻律性に関する種々のパラメータが規則によって決定されることから，合成音の品質をさらに向上させるためには，合成規則をよりいっそう充実させることが必要である．

音声規則合成システムは，合成規則の集まり，すなわち音声合成ルールベースを知識と見なすことにより，一種の知識ベースシステムと捉えることができる．従って，他の知識ベースシステムにおいて知識ベース構築が大きな問題となるように，音声規則合成システムにおいても音声合成ルールベースの構築，即ち合成規則の作成，修正が問題となる．従来，音声合成の専門家が自らの経験をもとに試行錯誤を繰り返すことにより音声合成ルールベースの構築を行なってきたが，この作業を人間が人手で行うことは，非常に困難な，しかも時間のかかる作業である．そこで，計算機を用いて音声合成ルールベースの構築を支援する環境を提

供し，効率良く容易に合成ルールベースの開発を行なうための研究が必要となる．Hertz らは，工学的研究者以外の人でも容易に合成規則の作成が行なえることを目的とし，柔軟に種々の規則表現形式に対応できる Delta と呼ばれる言語およびその開発環境を構築している[Hertz85].

本章では，音声合成の専門家が行なう音声合成ルールベースの構築において，計算機による有効かつ実現可能な支援と開発環境について考察した後，具体的な音声合成ルールベース構築支援手法を示す．3.3 では，帰納的学習に基づいて多数の音声データから合成規則を自動的に抽出する手法を示し，3.4 では，同様の手法に基づいて合成規則の修正案が生成できることを示す．また 3.5 では，インタプリタに基づいた音声合成システムについて述べ，音声合成時に発火した規則などのルール管理を行なうことによって，合成規則修正／評価の作業サイクルが大幅に短縮できることを示す．

3.2　ルールベース構築支援

前節でも述べたように，音声規則合成システムは一種の知識ベースシステムとみなすことができ，高品質の合成音を生成するためには音声合成における知識ベース，すなわち音声合成ルールベースの構築支援が重要な問題となる．このようなシステムの知識ベース構築では，大きく分けて，

(1) 処理の自動化を行なう．

(2) 情報管理を行ない，処理の作業サイクルを短縮する．

の二点において計算機を用いた支援が可能であると考えられる．近年，人工知能に関する研究が盛んに行われ，計算機を用いて知識の処理を行うことが可能になってきており，例題からの知識の自動生成あるいは知識の変換などが行なえる．知識ベースの構築においてもそれぞれの問題において自動化できるところがあり，知識処理技術の進歩に伴いその領域は拡大していくことが期待できる．しかしながら，全ての処理が自動化できるわけではないため，評価などのようにどうしても専門家が手作業で行なう処理が残される．このような処理として，修

正・評価・修正のような繰り返しの作業が多く行なわれると考えられ，それに必要な情報を計算機で管理することによってこの作業サイクルを短縮できる．以上のような観点から支援を行なうことが，知識ベース構築を行なう専門家の負担を軽減する．

次に，音声規則合成システムにおいて，ルールベースの構築を行なう音声合成研究者の立場から，どのような機能を計算機が提供することが有益であるかを考えてみる．まず，多数の音声データから合成規則の自動抽出を行うことは大きな支援機能と考えられる．それには音声データを管理し，供給することも必要となる．抽出された規則の評価を行い，合成音の品質が不十分である時の規則の修正，変更における支援も重要である．このために，個々の合成規則の履歴 (どの音声データから得られた規則か，どのように修正されたか) を管理し，修正されたルールベースを用いて短時間で音声を合成する．既存の合成規則を参照し，利用できれば規則の抽出・修正を有利に行えるであろう．さらに，マルチウィンドウ環境等を用いた優れたユーザインタフェースも支援システムにとって重要な要因である．これらの様々な機能を有機的に結合することによって，支援システムは実現されなければならない．また，支援システムは，それを利用することにより単に音声合成ルールベース，すなわち合成規則が得られるというだけでなく，総合的にルールベースを構築・管理することを目的として構築されるべきである．

以上のような考察のもとに，音声規則合成のためのルールベース構築支援環境 SED-SSRB (a Support Environment for Development of the Speech Synthesis Rule Base) を開発した．図3.1 に SED-SSRB のシステム構成図を示す．合成規則自動抽出部 REM (automatic Rule Extraction Module) では，多数の音声データから帰納的学習によって合成規則を自動的に抽出する．さらに修正案提示部 POM (Proposal Offering Module) では，合成規則修正のための修正案を提示する．規則条件部の一般化，特殊化の可能性を調べることにより自動的に修正案を生成し，修正に伴う副作用を評価することによってその優先順位を決定する．また，規則インタプリタに基づいた音声合成部 SSRI (Speech Synthesis system based on Rule Interpreter) で

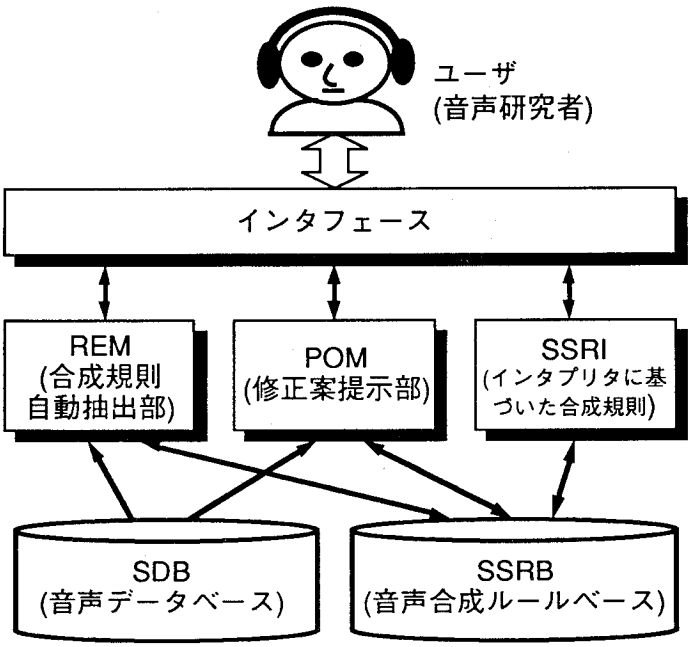


図3.1　SED-SSRB の構成

は，音声合成ルールベースと実行部を分離することにより，ルールベース修正後に合成システム全体を再構築することなく修正されたルールベースを用いた音声合成が行なえる．さらに，音声合成時に発火した規則を入力ごとに管理することによって，再合成時に要する処理時間を短縮する．REM と POM では，「正解」となる学習データに基づいて規則の抽出および修正案の自動生成を行なうため，どのようなパラメータ値にすると品質の良い合成音が生成されるかという例題が音声データベース SDB (Speech Data Base) として与えられなければならない．SED-SSRB は，以上四つのモジュールに加えて合成規則の集まりである音声合成ルールベース SSRB (Speech Synthesis Rule Base) とグラフィカルなインタフェースの六つのモジュールから構成されている．SED-SSRB における支援を上記観点から整理してみると，REM，POM での処理は (1) の自動化にあたり，SSRI での処理は検索や発火規則の管理など (2) の情報管理にあたると言える．

3.3 合成規則の自動抽出

多数のデータに基づいた帰納的な学習は，計算機の処理能力の著しい向上に伴い，情報処理全般において有効な問題解決パラダイムの一つとなっている．音声の研究においても，人の発声を録音，分析することによって種々の特徴パラメータの観測事例を比較的容易に収集できることから，知識ベースに基づく音声信号処理システムでは，多数の例題から帰納的学習によって規則を抽出する手法が有効であると考えられる．音声認識ではこれまでに認識規則の学習システムなどが提案されている[辻野89]．音声合成においても数量化法[三村91][海木92]，ニューラルネット[匂坂89]，決定木[Riley89][尼子91]など種々のモデルに基づいて，多数の音声データから合成パラメータ決定知識を獲得する試みがなされており，帰納的な合成規則の獲得に期待が大きい．このような観点から，音声合成のためのルールベース構築支援として条件部の一般化・特殊化に基づく合成規則の自動抽出を行なった．SED-SSRB における合成規則自動抽出部の構成を図3.2 に示す．

人間(音声研究者)が合成規則を作成する場合は，音声データを観察し，そこに現れる共通点を規則化していく．計算機にこのようなことを自動的に行わせるためには，単に音声データベースがあるだけではなく，計算機で扱うことのできる形式の例題を生成する必要がある．規則合成方式において合成音の品質が十分でない場合には，合成規則を用いて生成された合成パラメータ値と人間が発声する音声のパラメータ値との間に「差」があると考えられる．どのパラメータがどの様に聴覚系において重要であるかは明らかではないが，この二つのパラメータ値の「差」を小さくすることで一次近似的には合成音の品質が高くなると期待できる．そこで，パラメータ比較部では，多くの音声データに対して二つのパラメータ値，つまりルールベース中に予め用意されたデフォルト合成規則を用いて決定されたパラメータ値と，人間の発声した音声データの分析結果として得られるパラメータ値との差に注目した例題(以下，比較例と呼ぶ)を生成する．既に得られている合成規則がある場合には，デフォルトの合成規則としてそれらを用いることができる．帰納学習部では，「音声学に関する専門知識」が知識ベースとして

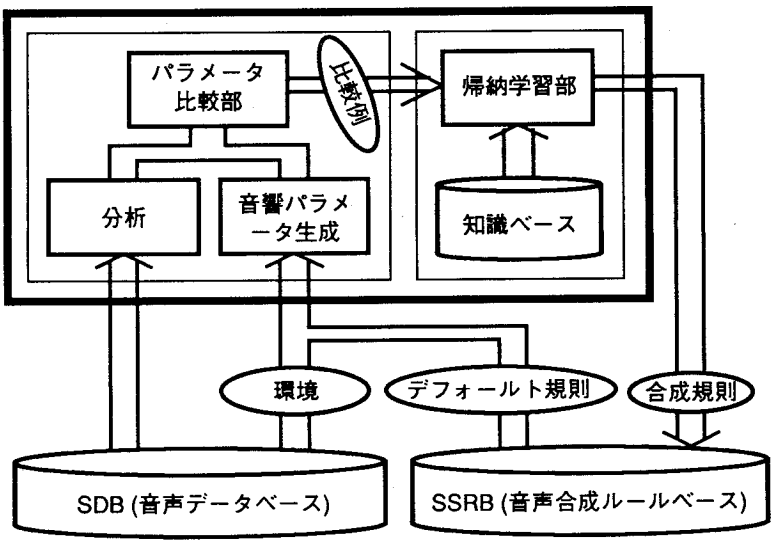


図3.2　自動抽出部の構成

与えられており，これを用いて比較例から帰納的に学習することにより合成規則の抽出を行う．

3.3.1 比較例の生成

図3.2のパラメータ比較部では，音声データベースに登録されている音声データに対し，ルールベース中のデフォルト規則を適用することにより生成された合成パラメータ値と，音声データを分析して得られたパラメータ値とを比較する．パラメータの「差」とその音声データの環境とを併せて比較例として記述する．比較例は，例えば，「後続する音節が /ka/ で，ピッチが低い語頭の音節 /a/ の継続時間長が 20ms 長い」というように，合成パラメータの違いと注目している音声データ (CV 音節，単語など) の環境とを関連づけて行われる．ここで環境は，

環境：[(環境素1,属性値), (環境素2,属性値), … (環境素n,属性値)]

のように環境素とその属性値の組から成る．環境素は，規則合成を行う場合に入力音韻系列から得られる音韻名に関する情報および辞書から得られるアクセン

```
duration(4000,1004,-35,
        [(pre_c,ts),(c,m),(post_c,r),
         (pre_v,u),(v,a),(post_v,u),
         (pre_l,no),(l,no),(post_l,no),
         (pre_u,no),(u,no),(post_u,no),
         (pitch,high),(pitch_V,no),
         (begin,no),(end,no),(mora,4)]).
```

図3.3　音節時間調における比較例の例

ト，単語境界，無声化などの情報で，音韻それ自身の素性も含まれる．例えば，上例での環境は，

環境：[(音節名,/a/), (後続音節名,/ka/), (語頭,yes),(ピッチ,low)]

と記述できる．

比較例の生成において，個々の音声データに対する環境はシステムがデータベースを検索することにより自動的に獲得される．環境素としてどの様な要因を用いるかは，制御すべき合成パラメータによって当然異なってくるため，どの合成パラメータのとき何を環境素として用いるかを「環境に関する知識」として予め与えておく．

音声データベースには，男女各1名のアナウンサーにより発声された連続文章，文節，単語，CV 音節等，合計約 2 時間の音声データが格納されており，文章，呼気段落，文節，単語，CV 音節の各レベルでセグメンテーションされている．

単語中の CV 音節時間長に関する比較例の生成例を図3.3 に示す．パラメータ比較部は Fortran で記述されているが，規則抽出を行う帰納学習部が Prolog で記述されているため，比較例はこのような Prolog の fact として出力され，二つのモジュール間のデータの引渡しを行っている．ここで，第 1，2 引数は音声データベース中の CV 音節の ID 番号に対応している．第 3 引数が合成パラメータの比較結果を，第 4 引数のリストが環境を表している．単語中の CV 音節時間長の場合には，環境素として，

音節の子音名，母音名，長音化，無声化，
先行音節の子音名，母音名，長音化，無声化，
後続音節の子音名，母音名，長音化，無声化，
語頭か否か，語尾か否か，ピッチの高低，ピッチの変化，単語モーラ数

の 17 の要因を用いており，それぞれ図3.3 では，

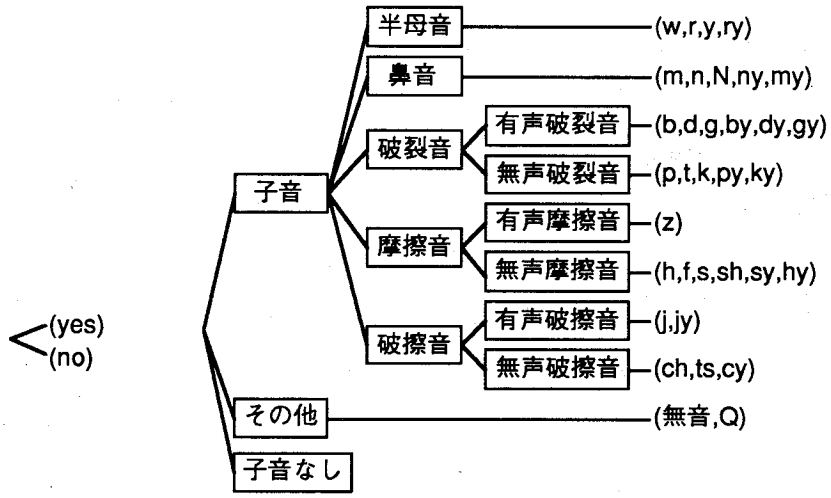
c, v, l, u,
pre_c, pre_v, pre_l, pre_u,
post_c, post_v, post_l, post_u,
begin, end, pitch, pitch_V, mora

という環境素名で表されている．この例では，第3引数がデフォルト規則を用いて生成した時間長より実際の音声データの時間長の方が 35ms 長いことを示しており，また，第4引数から CV 音節が語頭でも語尾でもない /ma/ で先行音節が /tsu/，後続音節が /ru/ であることなどがわかる．

3.3.2 専門知識の記述

規則抽出のアルゴリズムは 3.3.3 で述べるが，その過程において，各環境素に対し一般化・特殊化を行う．このためには個々の環境素に対する概念の階層性をシステムが知っている必要がある．本システムではこの階層性を「音声学に関する専門知識」として予め与えておく．図3.4 は，単語中の CV 音節時間長に対する 17 個の環境素のうち，(a) 二値のみをとる環境素 begin (語頭であるか否か) と，(b) 最も多くの値をとる環境素 c (子音) に関して，その属性値の階層性を示したもので，図3.5 はそれを Prolog で記述したものである．(b) に現れる記号 # は，母音のみの音節に対する子音名，あるいは語頭音節に対するピッチの変化などその値が定義できない場合に，計算機での処理を容易に行うために便宜上設けた属性値を表す．また，sil は無音音節を表している．

子音の場合にはその生成方法 (摩擦性，破裂性，...)，調音位置 (口唇，歯茎，...) あるいは有声／無声など異なった観点からの分類ができ，「口唇音」と



(a) 単語の開始か否か　　(b) 音節における子音

図3.4　環境に現われる概念の階層性の例

```
domain(begin,10,any,[no,yes]).
```

(a) 単語の開始か否か

```
domain(c,10,any,[cons,noCons,others).
domain(c,5,noCons,[#]).
domain(c,5,others,[q,sil]).
domain(c,8,cons,[affr,fric,stop,semivowel,nasal]).
domain(c,5,stop,[vStop,uStop]).
domain(c,5,fric,[vFric,uFric]).
domain(c,5,affr,[vAffr,uAffr]).
domain(c,5,semivowel,[w,r,y,ry]).
domain(c,5,nasal,[m,n,nn,ny,my]).
domain(c,2,vStop,[b,d,g,gy,dy,by]).
domain(c,2,uStop,[p,t,k,ky,py]).
domain(c,2,vFric,[z]).
domain(c,2,uFric,[h,f,s,sh,sy,hy]).
domain(c,2,vAffr,[j,jy]).
domain(c,2,uAffr,[ch,ts,cy]).
```

(b) 音節における子音

図3.5　環境に現われる概念の階層性の記述例

いった概念は図3.4 (b) の階層性には現れない．しかし，例題から帰納的に学習し規則性を発見する場合に，合成規則にはこのような概念が必要であることも考えられる．これに対処するため子音に関しては，図3.4 (b) で示した以外に二種類，計三種類の階層性を用意し，特殊化の操作において最も都合の良い階層性を選択している．

3.3.3 規則抽出アルゴリズム

音声データの観測例として得られた比較例の中には共通する傾向を持つものがあると考えられる．本論文で述べる規則抽出アルゴリズムでは，「比較例から傾向を見つけ出し規則として抽出する」操作を繰り返し行う．規則が抽出されるたびに比較例からその規則の効果を取り除くため，この処理は，凸凹になっている鉄板に板金を行うことにより平にしていくことと概念的には似ている．アルゴリズムを以下に示す．

Step.1　比較例の中で，「差」の絶対値が最大である比較例 c_{max}，およびその最大値 d_{max} を求める．(以下，$d_{max} > 0$ とする．)

Step.2　全ての比較例の中から c_{max} に似ているものを選択する．それには，まず比較例の「差」に着目し，「差」が閾値以上の (つまり，d_{max} に近い) 比較例の集合 C_1 を選択する．閾値は $d_{max}/2$ とした．さらに比較例の環境に着目し，C_1 の中から c_{max} との環境の距離が閾値以下の (つまり，環境が似ている) 比較例の集合を選択し，C_2 とする．閾値は 120 とした．(環境の距離の定義については後述する．)

Step.3　図3.4 で示したような環境素の概念の階層性に関する専門知識を用いて，(2) で後述するように各環境素の属性値を一般化し，C_2 の個々の環境を全て被覆する環境 E を求める．ここで一般化とは，ある概念をより広い意味の概念に変換する (例えば，"有声破裂音" を "破裂音" に変換する) ことを言い，比較例に見られる環境素に対する属性値そのものも一つの概念と考える．

Step.4 環境Eに被覆されるような環境をもつ比較例の集合C_3を求める．C_3における「差」の平均値をd_{av}とする．

Step.5 C_3における「差」の分布がある範囲内に収まったとき，つまりC_3の中の「差」が最小である比較例c_{min}の「差」をd_{min}とし，

$$d_{min} > d_{th} \ (= -0.2 \times d_{max})$$

のとき，Step.8へ．

Step.6 c_{min}をC_3から取り除くためにc_{max}とc_{min}の環境の中で，異なった属性値を持つ環境素に注目し，(3) で後述するように環境Eを特殊化する．ここで特殊化とは，新しく条件 (環境素と概念の組) をEに付加すること，および，既にEに含まれる条件の概念をより狭い意味の概念に変換することを言う．

Step.7 Step.4へ．

Step.8 「環境Eのとき合成パラメータを$r_d \times d_{av}$だけ増加させる」が合成規則として抽出される．r_dは0.6とした．

Step.9 C_3の比較例に対し，「差」を$-r_d \times d_{av}$することによりこの規則の効果を取り除く．

Step.10 Step.1へ．

このように規則抽出を繰り返し行い，比較例の「差」の絶対値を小さくしていく．ここでは，$d_{max} > 0$の場合について述べているが，$d_{max} < 0$の場合も「差」の符号を全て逆に考えることで同様の処理を行う．

各手順において選択される比較例がどのように変わるかを図3.6に模式的に示す．図3.6では○●印が個々の比較例に対応しており，横軸はその頻度を縦軸は「差」を表している．また，●印が選択されている比較例を表している．Step.1～2では (a)で示すように誤差の最も大きい比較例c_{max}とそれに似た傾向を持つ比較例がC_2として選択される．Step.3では，選択された比較例の一般化を行っている．Step.4では (b) で示すように一般化された環境 (規則の条件部) とマッチする比較例がC_3として選択されている．一般に，C_3における「差」は広い範囲

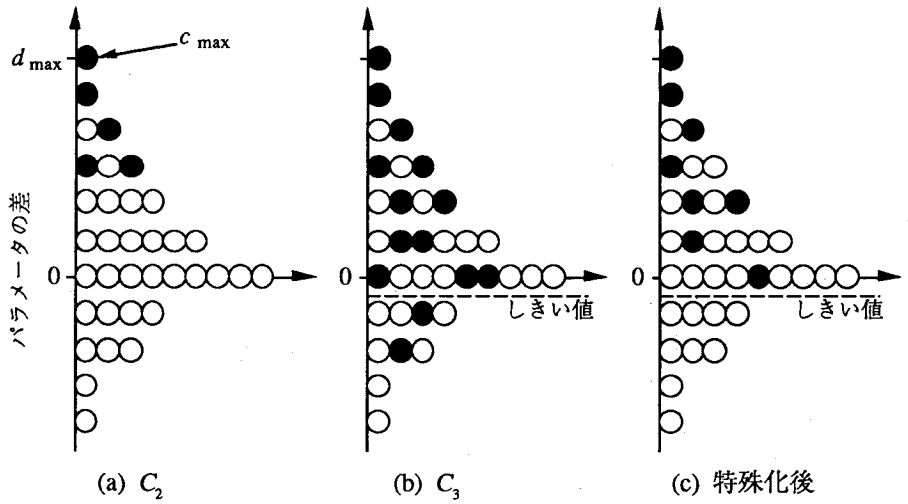


図3.6　規則抽出において選択される比較例

に分布するため，(c) で示すように d_{min} が閾値を越えるまで Step.6 で特殊化が行われる．

(1) 環境間の距離

環境 e_k，e_m の距離 d_{km} は次式のように求められる．

$$d_{km} = \sum_i w_i \cdot f_i(v_{ki}, v_{mi}) \tag{3.1}$$

ここで，v_{ji} は環境 e_j における i 番目の環境素の値である．環境の距離は，各環境素間の距離 f_i を重み w_i で荷重和をとったものである．f_i に関しては概念の階層性の知識とともに図3.5のようにシステムに与えておく．述語 domain の第2引数が距離を表しており，例えば，環境素 c の場合，/b/ と /d/ では「有声破裂音」という概念が両者を含むことから距離は2となり，また，/b/ と /k/ では「破裂音」に両者が含まれることから距離は 5 となる．各環境素の重み w_i は合成規則が抽出されるたびに比較例の分布に応じて決定されるべきであると考えられる．しかし，そのための制御が非常に難しく，比較例の分布を気にし過ぎると入力された比較例に依存した一般性に乏しい規則が抽出されるる危険性もあることから，本論文では重みは全て 1.0 に固定とした．

(2) 環境の一般化

Step.4 において環境の一般化が行われる．一般化は各環境素ごとに独立に行われ，図3.4, 3.5 に示した専門知識を用いることにより，例えば，

(c,b),(c,t) --> (c,[stop])

(begin,yes),(begin,no) --> (begin,[any])

というように複数の属性値からそれら全てを含む一つの概念が導かれる．ここで概念 any は図3.5 に示すように全ての環境素において階層性の最も上に位置する概念として便宜的に定義されたもので，その環境素に対する全ての属性値が含まれる．従って，上例の後者のように一般化の結果，概念が any となった場合には，環境 E から取り除かれる．

しかし，このような一般化を行った結果，出現頻度が少ない属性値が(例えば，u (無声化) に対する yes，c (子音) に対する q (促音) など) 比較例の数が少ないためにたまたま比較例として現れず，一般化が不十分となることがある．このようなデータによる不十分な一般化に対処するため，例えば，

(u,[no]) --> (u,[any])

(c,[cons]) --> (c,[any])

というような一般化規則を手続き的な「専門知識」として与えておき，このような「音節が無声化していない」とか「子音が無音・促音以外である」といったそれほど意味が無いと考えられる条件が，一般化の結果として環境 E に含まれることを防いでいる．これに対して図3.4 に示した知識は宣言的な「専門知識」であると考えることができる．

(3) 環境の特殊化

Step.6 において E を特殊化し，E により被覆される比較例の「差」の範囲を抑える．このために，比較例 c_{max} が環境 E にマッチしたまま c_{min} がマッチしないように E を特殊化する．一般に，c_{max} と c_{min} は複数の環境素においてその属性値が異なる．そのような環境素を特殊化の候補として全て求め，その中で最も良

```
cond_cand(l,any,[yes],'[]',[no]).
cond_cand(pitch,any,[high],'[]',[low]).
cond_cand(c,cons,[stop],[affr,fric,semivowel],[nasal]).
cond_cand(c,cons,[palatal],
        [labial,alveolar,glottal],[hatsuon]).
cond_cand(c,sglCons,[sglStop],
        [sglAffr,sglFric,sglSemivowel],[sglNasal]).
cond_cand(v,any,[vowel],'[]',[others]).
cond_cand(pre_c,any,[noCons],[others],[cons]).
cond_cand(pre_v,vowel,[fVowel],'[]',[bVowel]).
```

図3.7　特殊化のための条件の候補例

いものを選択する．図3.7は特殊化のために付加する(置き換える)条件を列挙した例である．図3.7の3行目の例ではc(子音)の属性に関して，c_{max}，c_{min}の値を共に含む最小の概念がcons (第2引数) であり，consに含まれる概念の内，stop (第3引数) がc_{max}の値を含み，nasal (第5引数) がc_{min}の値を含むことを示している．第4引数はconsに含まれる概念の内，stopとnasal以外を示しており，c_{min}を含む概念を除いたもの，つまり第3，4引数を合わせた (c,[stop,affr, fric, semivowel]) が特殊化のために追加される条件の候補となる．子音に対しては三種の階層性が用意されていることから，三つの候補が挙がっている．一般に特殊化を行うことによりEにより被覆される比較例の数は減少する．できるだけd_{max}に近い「差」を持つ比較例ができるだけ多くEにより被覆される方が規則の信頼性は高いと考えられる．C_3の比較例は特殊化を行いC_3から除くべき比較例C_{3neg}とC_3に含まれるべき比較例C_{3pos}に分けられる．このときの閾値がd_{th}である．一般には，一つの条件をEに付加することでC_3から除かれる比較例がC_{3neg}と一致しないため，各特殊化の候補に対してそれぞれ得点付けを行い，最も得点の高いものを特殊化のために付加する条件として選択する．得点は次式のように計算する．

$$得点 = \Sigma \mid d_{1i}-d_{th} \mid + \Sigma \mid d_{2i}-d_{th} \mid - \Sigma \mid d_{3i}-d_{th} \mid - \Sigma \mid d_{4i}-d_{th} \mid \quad (3.2)$$

d_{1i} : C_{3pos} のうち，特殊化により除かれない比較例の「差」

d_{2i} : C_{3neg} のうち，特殊化により除かれる比較例の「差」

d_{3i} : C_{3pos} のうち，特殊化により除かれる比較例の「差」

d_{4i} : C_{3neg} のうち，特殊化により除かれない比較例の「差」

(4) 自動抽出された合成規則

単語中のCV音節時間長に対する規則抽出の結果得られた合成規則の一部を図3.8に示す．これは男性アナウンサーの発声した65単語に含まれる231個のCV音節に対する比較例から自動抽出されたものである．デフォルトの合成規則としては，データベース中の単語データ (約 600 個) の各 CV 音節の平均時間長を用いた．例えば，一番目の規則は「長音化している音節の時間長を +140ms する」ことを表しており，二番目の規則は「長音化している音節で子音が歯茎音，口唇音，口蓋音，撥音のいずれかであればさらに +67ms する」ことを表している．図3.8に示される合成規則は，長音，語尾の場合に時間長が長くなり，無声化した場

```
rule(1,140,[(l,[yes])],1).
rule(2,67,[(l,[yes]),
        (c,[palatal,labial,alveolar,hatsuon])],1).
rule(3,39,[(end,[yes]),
        (post_c,[sil]),
        (post_v,[#]),
        (pre_c,[fric,affr,semivowel,nasal]),
        (pitch_V,[no,up,#]),
        (c,[cons,others])],1).
rule(4,28,[(end,[yes]),
        (post_c,[sil]),
        (post_v,[#]),
        (pitch,[high]),
        (v,[vowel]),
        (c,[stop,affr,fric,nasal])],1).
rule(5,-50,[(u,[yes])],1).
```

図3.8　抽出された合成規則の例

```
ele_dependency((u,[yes]),[l]).
ele_dependency((l,[yes]),[u]).
ele_dependency((begin,[yes]),
        [end,pre_u,pre_l,pre_c,pre_v,pitch_V]).
ele_dependency((end,[yes]),
        [begin,post_u,post_l,post_c,post_v]).
ele_dependency((pre_u,[yes]),[pre_l]).
ele_dependency((pre_l,[yes]),[pre_u]).
ele_dependency((post_u,[yes]),[post_l]).
ele_dependency((post_l,[yes]),[post_u]).
ele_dependency((pitch_V,[up]),[pitch]).
ele_dependency((pitch_V,[down]),[pitch]).
```

図3.9　環境素における依存性の記述

```
rule(1,140,[(l,[yes])],1).
rule(2,67,[(l,[yes]),
        (c,[palatal,labial,alveolar,hatsuon])],1).
rule(3,39,[(end,[yes]),
        (pre_c,[fric,affr,semivowel,nasal]),
        (pitch_V,[no,up,#]),
        (c,[cons,others])],1).
rule(4,28,[(end,[yes]),
        (pitch,[high]),
        (v,[vowel]),
        (c,[stop,affr,fric,nasal])],1).
rule(5,-50,[(u,[yes])],1).
```

図3.10　冗長性を取り除いた合成規則の例

合に時間長が短くなることを示しており，専門家の知見と一致していることがわかる．

これらの合成規則の条件部はいくつかの環境から成っているが，環境素は互いに独立ではない．例えば，図3.8 の三番目の規則の場合，(end,[yes]) で語尾であるから当然後続音節は無音であり，(post_c,[sil]),(post_v,[#]) は冗長となる．このような合成規則の条件部の冗長性を取り除くために環境素間の依存関係を図3.9 に示すように「音声学に関する専門知識」として与えておく．図3.10 は図3.8 の規則に対して条件の冗長性を取り除いた結果である．

(5) 他の合成パラメータへの適用

本論文では，規則抽出の対象となる合成パラメータとして，時間長だけを取り上げている．ピッチ，ホルマント周波数などが時間的に変化する合成パラメータであるのに対して，時間長は各音節に対して一つの値を決定すれば良く，自動抽出を行い易い合成パラメータであると言える．このため，いくつかある合成パラメータの中で，まず時間長を取り上げた．ピッチなど他の合成パラメータの場合にも，一つの音節に対して複数の時点で比較例を生成し，時間変化に対応することで，本手法を適用できると考えられる．

また，本論文では単語データに対してのみ規則抽出を行ったが，これは単語の場合には環境として考慮すべき要因が文章に比べて少ないためである．文章データに対する規則抽出では，文章中の位置や品詞に関する情報など，さらに環境素を増やすことが必要であろう．

3.3.4 抽出された合成規則の評価

自動抽出された合成規則の評価を行うため，合成規則を順次適用していくことにより合成パラメータの誤差がどう減少していくかを調べた．男性アナウンサー一名により発声された単語中のCV 音節時間長に対して抽出された合成規則は30個である．結果を図3.11 に示す．(a) は規則抽出に用いたデータ (65 単語に対する比較例 231 個) に対して，(b) はそれ以外のデータ (64 単語に対する比較例 215

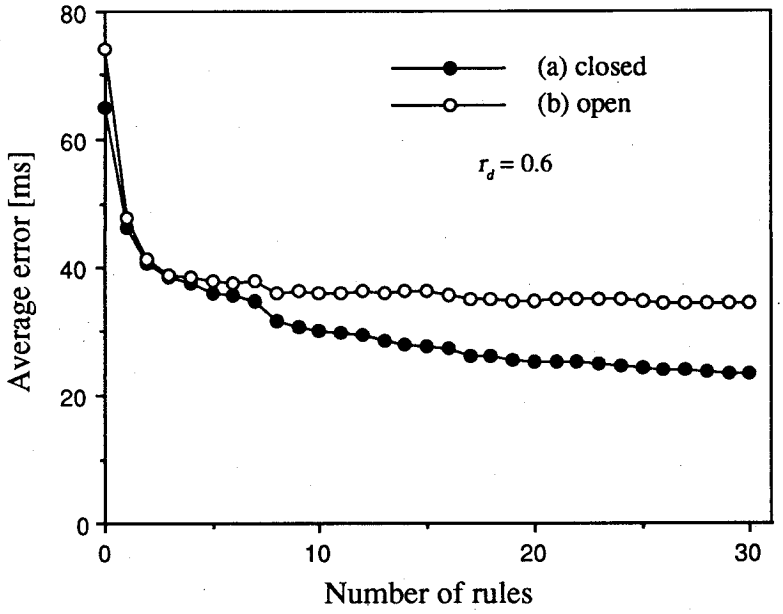


図3.11　ルール数の増加による誤差の減少 (I)

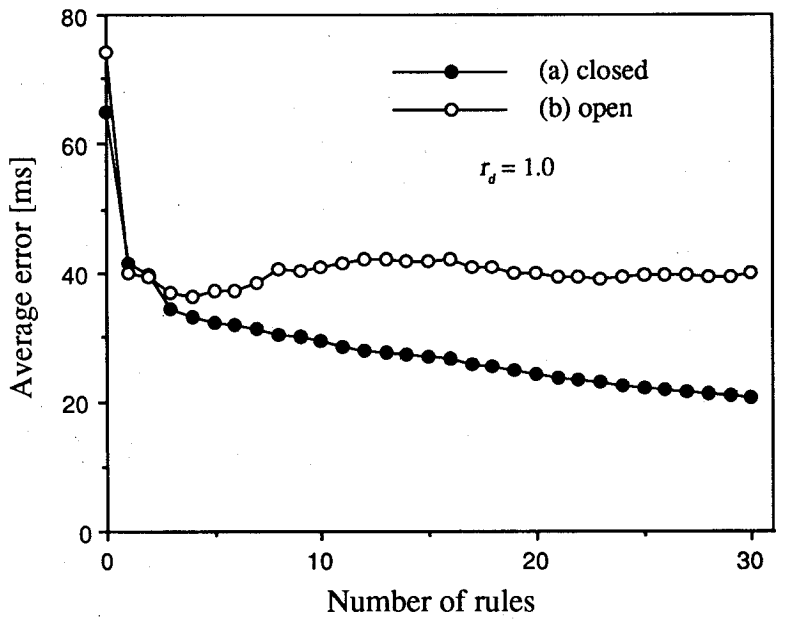


図3.12　ルール数の増加による誤差の減少 (II)

個) に対して誤差の変化を調べたものである．(a) では，デフォルト規則のみを用いた場合に 65ms であった誤差が，抽出された 30 個の合成規則を用いることにより 23ms にまで徐々に減少している (用いた音声データの平均 CV 音節時間長が約 220ms であったことから，相対的には約 20% の改善である)．誤差は各 CV 音節時間長の「差」の絶対値の平均として表している．(b) では，オープンな実験であるため，いくつかの規則はそれを適用することで誤差が増加しているが，(a) において誤差の減少に大きく寄与する規則は (b) においても同様の傾向を示している．

本章で述べた規則抽出アルゴリズムでは，誤差の大きいところに注目し，規則抽出を行った後で，その規則の効果をデータ (比較例) から取り除いている．このため，次々にデータを変更していくことで規則として抽出されるべき誤差の傾向が壊されないようにしなければならない．規則抽出アルゴリズムの Step.8 における r_d は比較例の傾向をどの程度規則に反映させるかを表すパラメータである．図 3.12 は r_d を 1.0 とした場合に抽出された 30 個の合成規則による誤差の減少を示したものである．(a), (b) の区別は図3.11 と同じである．尚，図3.11 での r_d は 3.3.3 節で述べたように 0.6 である．(a) の規則抽出に用いたデータに対しては図 3.11，3.12 ともほぼ同様の傾向を示しているが，(b) の規則抽出に用いなかったデータに対しては図3.12 の場合に規則を適用することでかなり誤差が増加している．合成パラメータは環境による複数の要因が重なり合って値が決定され，比較例における「差」も複数の合成規則の欠如，不備によるものと考えられる．従って，規則抽出アルゴリズムの Step.4 から得られる d_{av} の値は環境 E によるものだけではない．d_{av} における環境 E による成分だけを，規則により修正される合成パラメータの値として用いることが理想的であるが，それは難しく，本手法では徐々に合成パラメータを修正していくことにした．(b) の場合に図3.12 より図3.11 の方が誤差が減少していることから，r_d=0.6 とすることが妥当であることがわかる．

3.4　合成規則修正案の提示

前節で述べたように，REMでは合成規則を自動抽出し初期的な音声合成ルールベースを構築できる．しかし，学習データは限られた一部であることなどから自動抽出だけで満足できる音声合成ルールベースが構築できるわけではない．合成音の品質が充分満足できるものでなければ，専門家は人が発声した自然音声の分析結果を参考にするなどして，音声合成ルールベースの修正を行なわなければならない．しかしながら，規則を修正すると，あるテキストの音声合成ではうまくパラメータの設定を行なうことができても，別のテキストではその修正が悪影響を及ぼすこと，つまり副作用が考えられる．一般には多数の合成規則が存在し，どのような規則の修正が最も適当であるかを判断することは極めて困難である．そこで，POM (Proposal Offering Module) は音声合成規則の修正案 (以下，単に修正案) を自動的に生成し，評価結果の良いものから順に専門家に提示する．

ここで，修正案の生成は次の状況で行なわれる．第一に，本手法は「正解」となる学習データに基づいて行なわれるため，音声合成システムの出力すべき発声内容を人が予め発声しておかなければならない．一般に，規則合成方式の音声合成は任意の発声内容を出力できることが大きな利点であり，従って，その全ての発声を自然音で用意することは不可能かつ意味のないことである．しかし，ある定型の文やある語彙が頻繁に用いられるような合成音の応用も考えられ，このような場合には汎用的な音声合成ルールベースをある領域にチューニングすることも必要となる．第二に，修正案の生成はどの合成パラメータを修正するかを専門家が決定した後に行なわれる．言い替えると，「異種のパラメータ間でどれを修正するか」という判断は，専門家によって既に決定されている．

また，合成規則の実行部は一般に複数のパラメータ操作関数から成っているが，修正案の生成では簡単化のために，実行部は合成パラメータを増減させる一つのパラメータ操作関数から成っているものとする．実行部に複数の関数を持つ規則は，実行部が一つの関数から成る複数の規則に容易に分解できるため，このような仮定は一般性を失わない．

3.4.1 修正案の自動生成

ある入力音韻列における音節Sに対して規則によって決定されたパラメータPが，その理想値に比べて大きな誤差dを含んでいる場合に，$|d|$が減少するように規則を修正することを考える．ただし，理想値は人間が発声した音声の分析などにより決定できるものとする．また，ここでは規則の修正は条件部のみの修正に限る．従って，音節Sに対して既に発火して誤差を増大させている規則の条件部を厳しくする(特殊化する)操作と，まだ発火していない規則の条件部を緩める(一般化する)操作が考えられる．(ここで，条件部の適合した規則の実行部が実行されることを「規則が発火する」と呼ぶ．)

(1) 合成規則の特殊化

パラメータPを決定する規則に関して，音節Sに対して発火した規則のうち，パラメータPの誤差を増大させている規則R_{bad}が存在することがある．このような場合には，規則R_{bad}の条件部C_{bad}を特殊化し音節Sに対する発火をおさえることにより，音節SにおけるパラメータPの誤差を減少させることができる．ここでも規則の自動抽出と同様に，図3.4に示すような合成規則の条件部に現れる各環境素の属性値に関する階層性を利用して合成規則の環境条件を特殊化する．

条件部C_{bad}の特殊化は，任意の環境素に関して行うことができるため，R_{bad}に対して一般に複数の修正案が生成される．以下のようにC_{bad}を特殊化し修正案を生成する手続きを以下に示す．

Step.1 全ての環境素の中から一つを選択し，e_jとする．

Step.2 Sにおけるe_jの属性値をa_jとする．C_{bad}にe_jに関する条件が存在すればstep.4へ．

Step.3 C_{bad}に(e_j,[any])の条件を付加する．(全ての環境素に対する概念の階層性において，その最上位概念がanyである．)

Step.4 C_{bad}におけるe_jに対する属性値リストの中にa_jが存在すればこれを削除する．step.6へ．

Step.5 C_{bad}におけるe_jに対する属性値リストの中からa_jを包含する概念を選

(a) (e_j,a_j) = (c,g),
[(begin,[yes]),(c,[affr,stop])]
=> [(begin,[yes]),(c,[affr,uStop,b,d,by,dy,gy])]

(b) (e_j,a_j) = (pitch,low),
[(begin,[yes]),(c,[affr,stop])]
=> [(begin,[yes]),(c,[affr,stop]),(pitch,[high])]

図3.13　条件部の特殊化例

択し，それを下位の概念のリストに置き換える．step.4 へ．

Step.6　全ての環境素に対して処理が終れば終了．でなければ step.1 へ．

図3.13 は，ここでの特殊化の例を示しており，=> の左辺，右辺がそれぞれ特殊化前，特殊化後の規則の環境条件である．(a) では環境素 c に関して特殊化が行われ，S における環境素 c の属性値 g が環境条件から削除されている．また，(b) では環境素 pitch に関する条件が環境条件に存在していないため，新しく pitch に関する条件が付加されている．

(2) 合成規則の一般化

パラメータ P を決定する規則に関して，音節 S に対して発火していない規則のうち，それが発火することによりパラメータ P の誤差を減少させる規則が一般に存在する．このような規則が，その環境条件を変更することにより音節 S に対して発火する場合には，環境条件を一般化することを考える．生成される修正案の数を抑えるために，本システムでは条件部に含まれる環境素のうち一つだけの環境素に関してその属性値リストを変更することを考慮の対象としている．

条件部の一般化を行うには，(1) と同様に図3.4 に示した知識を用いる．条件部の変更によって音節 S に対して発火する規則を R_{near}，その環境条件を C_{near} とする．以下のように C_{near} を一般化し，修正案を生成する．

Step.1　規則 R_{near} を音節 S に発火させるために緩めるべき条件を求め，それに関する環境素を e_j とする．

(a) (e_j,a_j) = (c,g),
[(begin,[yes]),(c,[affr])]
=> [(begin,[yes]),(c,[affr,g])]

(b) (e_j,a_j) = (c,g),
[(begin,[yes]),(c,[affr,uStop,b,d,by,dy,gy])]
=> [(begin,[yes]),(c,[affr,stop])]

(c) (e_j,a_j) = (begin,no),
[(begin,[yes]),(c,[affr,stop])]
=> [(c,[affr,stop])]

図3.14 条件部の一般化例

Step.2 S における e_j の属性値を e_j とする．C_{near} における e_j に対する属性値リストに a_j を追加する．(ここで，R_{near} の定義により，他の環境素に関する条件は S に対して満たされているから，e_j が C_{near} に含まれていなければ R_{near} は S に対して既に発火している．したがって，C_{near} には e_j に関する条件が必ず存在する．)

Step.3 a_j の上位概念を a_{super} とする．a_{super} に包含される概念が全て e_j に関する属性値リストに存在する場合には，これを a_{super} に置き換える．さらに上位の概念に置き換えることができる場合には繰り返しこれを行う．

Step.4 e_j に対する属性値リストが [any] であれば，C_{near} から e_j に関する条件を削除する．

Step.5 終了．

図3.14は，一般化の例を示しており，=>の左辺，右辺がそれぞれ一般化前，一般化後の規則の条件部である．(a)，(b) は共に環境素 c に関して一般化が行われた例であるが，(a) では S における e_j の属性値 g が属性値リストに単に追加されただけであるが，(b) では b, d, g, by, dy, gy が vStop に置き換えられ，さらに vStop, uStop が stop に置き換えられている．(c) では begin に関して属性リストに yes が追加された後 yes, no が any に置き換えられたため，条件部から削除されている．

3.4.2 修正案の順序付け

合成規則の修正案はシステムによって生成されるが，どの案を採用して実際の規則修正を行なう (あるいは，全ての案を破棄して新しい規則の追加を行なう) のかの判断は，専門家が行なう．従って，数多くの修正案が生成された場合には (実際には，数十個の案が生成される)，専門家の判断を助けるための情報を提供する必要がある．このためにシステムは各修正案の良さを評価し，修正案の順序付けを行なう．

修正案の良さは，その修正案を採用することにより，どの程度誤差が減少するかで評価する．D をその修正案によって影響を受けるデータ，d_i は音節 i の規則修正前の誤差，d_i' は規則修正後の誤差とすると，修正案の良さ P は

$$P = \sum_{i \in D} (d_i^2 - d_i'^2) \tag{3.3}$$

で評価する．システムは，P が大きい順に修正案を提示する．

3.4.3 音声合成ルールベースの修正例

SED-SSRB を用いた音声合成ルールベースのチューニングでは，ユーザである音声合成の専門家がインタフェースを通じてコマンドを選択し，ルールベースに対する操作を行なっていく．インタフェース部は，コマンド選択のためのメニューの表示の他，全ての音節に対する誤差 (規則で決定された音節時間長と自然音声データにおける時間長の差) の分布や，選択された音節における発火規則の表示などを行なう．専門家は，規則数が増加しないようにするために先に POM に対し修正案の提示を求め，良い案が提示されてない場合には REM によって規則の追加を行なう．また，この二つのモジュールの呼び出しの他に，専門家が思いついた規則の修正／追加／削除をマニュアルで実行することもできる．この場合にも，その修正などの操作による効果が誤差の分布図上に示される．もちろん，次節で述べる SSRI を起動することによってその時点でのルールベースを用いた音声合成を行なうこともできる．

表3.1　SED-SSRB を用いた時間長規則修正における操作の内訳

操作内容	操作数
マニュアルによる規則追加	4
マニュアルによる規則修正	2
提示された修正案の採用	1 4
規則自動抽出による規則追加	1

前節で自動抽出によって得られたCV音節時間長規則のうちの最初の20個を，別の(人口20万人以上の)100都市名の単語データへSED-SSRBを用いてチューニングした．100都市名の音声データとして，男性話者一名が一回発声し，視察により音節単位でラベルを付けたものを学習データとして用いた．一人の専門家が約75分間のルールベース修正作業を行ない，21回のルールベース修正操作を実行した．この結果，100単語中の356音節に対する二乗平均誤差が80msから36msに減少した．21回の操作の内訳は，表3.1に示すとおりである．修正案の提示や規則の自動抽出を利用することにより，比較的短時間で有効なルールベースの修正を行なうことができた．

図3.15は，SED-SSRBを動作画面の例を示したものである．左上の画面には，音声波形，各音節に対して発火する規則の番号と規則の発火による時間長の変化が表示されている．中央下に表示されたメニューでは，コマンドの入力，修正案の選択などを行なう．この画面では一つの修正案を選択している．左下のグラフは修正案の効果および副作用を表示するためのもので，縦軸が時間長の誤差，横軸がデータ数，丸印がそれぞれの音節データを表す．メニューで選択された修正案によって影響を受けるデータが●で示され，その大きさが矢印で示されている．その他の右のウィンドウは，修正案の内容や音声データのID，システムの状態を表示している．

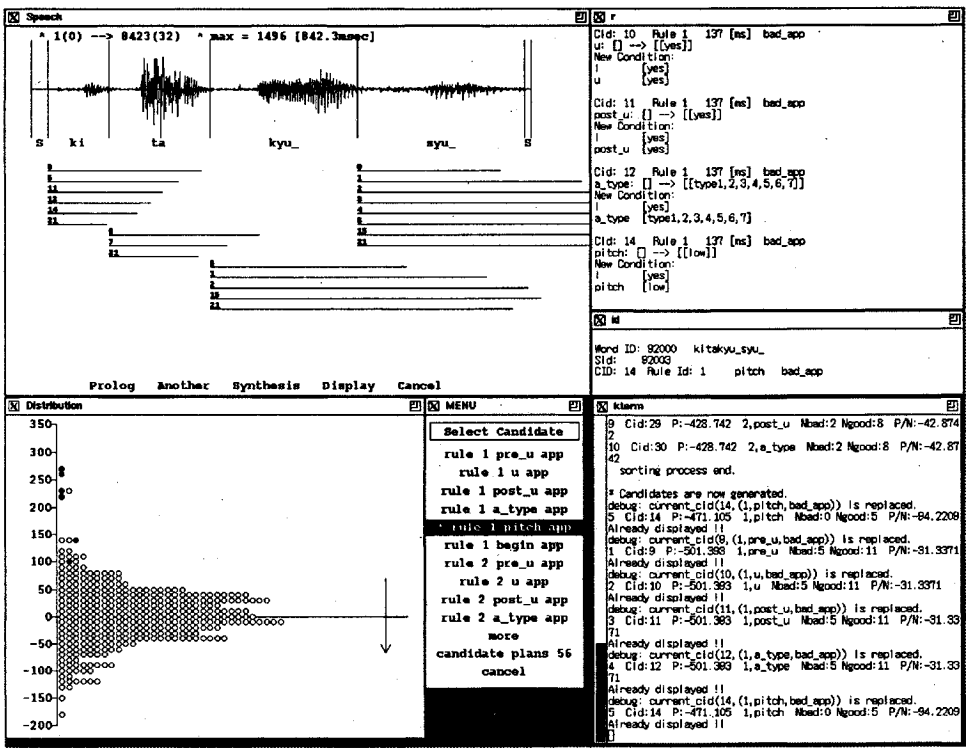


図3.15　SED-SSRB の動作画面例

3.5 規則インタプリタに基づいた音声合成システム

音声合成ルールベースの構築では，一般に合成規則の修正，音声の合成，合成音の聴取，合成規則の修正という処理が頻繁に，繰り返し行なわれる．従って，この作業サイクルにおけるターンアラウンド時間を短縮することが，効率の良い音声合成ルールベースの開発には不可欠である．合成規則の修正を行なった時にその効果をすぐに確認できるようするためには，音声合成システム全体を再構築することなく修正された合成規則を用いた音声合成が行なえなければならない．また，合成規則修正後には繰り返し同一の入力に対して合成音を生成することが多いことから，一度合成した入力テキストから再度音声を合成する (以下，再合成と呼ぶ) 場合の処理時間を短縮することも重要である．

このような観点から，規則インタプリタに基づいた音声合成システム SSRI (Speech Synthesis system based on Rule Interpreter) を開発した．SSRI では，合成規則を蓄える音声合成ルールベースとそれを実行する手続き部を分離し，逐次解釈によって規則が適用されるため，修正されたルールベースを用いた音声合成が容易に行なえる．さらに，発火規則に関する履歴情報を入力ごとに管理することによって，再合成時には条件部のテストを大幅に省略し，極めて短時間で合成規則を適用することができる．

合成音を生成する SSRI はもちろん SED-SSRB の一つのモジュールであるが，SSRI 単独で音声合成ルールベースの開発を行なっていくこともできる．Klatt 合成器のためのホルマント規則などでは，ホルマント周波数を正確に分析することが必ずしも簡単ではないことから，帰納的学習における学習データを用意することも難しく，SSRI 単独での音声合成ルールベースの構築も必要になる．

SSRI の構成を図3.16 に示す．入力された音韻列 (ローマ字表記) は，まずテキスト解析部において音節データと呼ばれる CV 音節ごとの情報に分解される．合成規則インタプリタ RI (Rule Interpreter) は，各音節データに対して条件部が満たされる合成規則を合成規則管理部 RM (Rule Manager) を通じて検索し，得られた全ての規則の実行部を実行する．RI では音節ごとにいくつかの時点における合成パラメータの値が決定され，さらに，パラメータ補間部において音声合成器が必

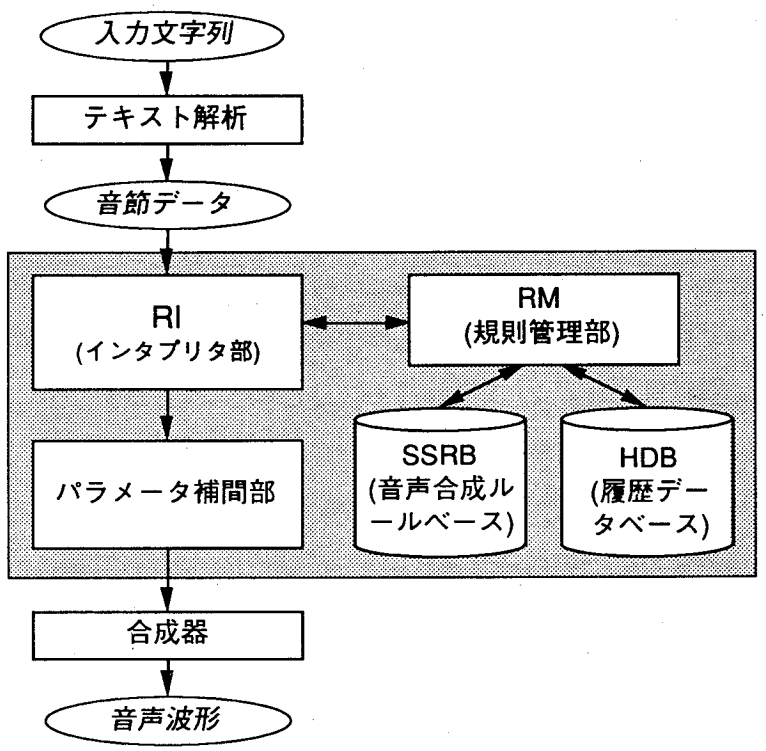


図3.16　SSRI の構成

要とする 5ms ごとのパラメータ時系列に変換される．また，RM は入力テキストごとに発火した規則の履歴を履歴データベースとして保持し，音声合成ルールベースの修正が行なわれるとそれに合わせて規則の発火履歴を更新する．ここで，パラメータ補間部，音声合成器は C 言語で，その他のモジュールは Prolog でインプリメントされており，音声合成器は Klatt 型のホルマント合成器[Klatt80]を用いている．

3.5.1 データの形式

合成規則インタプリタ RI への入力となる音節データ，および合成規則の形式を以下に示す．これらは，Prolog の fact として記述されている．

(1) 音節データ

音節データは，

syllable(*TID*, *SID*, *Env*).

の形式で記述される．*TID*，*SID* はそれぞれ，入力されたテキスト，テキストにおける音節の ID 番号である．*Env* は，その音節の属性，特徴を表すリストで環境と呼ばれる．環境は，

環境 : $[(e_1,v_1),(e_2,v_2),\cdots,(e_n,v_n)]$

のようなリストで記述される．ここで，e_i は環境素，v_i はその属性値である．現在，環境素として以下の 19 種のものを用いている．

音節の子音名，母音名，長音化，無声化，
先行音節の子音名，母音名，長音化，無声化，
後続音節の子音名，母音名，長音化，無声化，
呼気段落の先頭か否か，呼気段落の末尾か否か，
ピッチの高低，ピッチ変化，単語のモーラ数，
単語のアクセント型，呼気段落中のモーラ位置

で，実際に用いられる表記は，それぞれ，

c, v, l, u,
pre_c, pre_v, pre_l, pre_u,
post_c, post_v, post_l, post_u,
begin, end,
pitch, pitch_V, mora,
a_type, mora_p

である．3.3 で示した自動抽出のための例題における環境と比較してみると，アクセント型ごとのピッチパタンを生成するために，"単語のアクセント型" と "呼気段落中のモーラ位置" が追加されている．また，3.3 では単語データだけを扱った

```
syllable ( 1, 1,
        [ ( pre_c, sil ), ( c, ch ), ( post_c, r ),
          ( pre_v, sil ), ( v, i ), ( post_v, i ),
          ( pre_l, no ), ( l, no ), ( post_l, no ),
          ( pre_u, no ), ( u, no ), (post_u, no ),
          ( a_type, 0 ), ( mora_p, 1 ),
          ( pitch, low ), ( pitch_v, # ),
          ( begin, yes ), ( end, no ), ( mora, 6 ) ] ).
syllable ( 1, 2,
        [ ( pre_c, ch ), ( c, r ), ( post_c, n ),
          ( pre_v, i ), ( v, i ), ( post_v, o ),
          ( pre_l, no ), ( l, no ), ( post_l, no ),
          ( pre_u, no ), ( u, no ), (post_u, no ),
          ( a_type, 0 ), ( mora_p, 2 ),
          ( pitch, high ), ( pitch_v, up ),
          ( begin, no ), ( end, no ), ( mora, 6 ) ] ).
syllable ( 1, 3, ...
```

図3.17　音節データの例

が，ここでは文の合成も行なうため，"語頭か否か"，"語尾か否か" はそれぞれ "呼気段落の先頭か否か"，"呼気段落の末尾か否か" に変更されている．環境素は，規則合成を行う場合に入力テキストから得られる音韻名に関する情報及び辞書から得られるアクセント等の情報で，音韻それ自身の組成も含まれる．図3.17に入力 /chirinobeNkyo/ の始めの音節に対する音節データを示す．

(2) 合成規則

SED-SSRB において合成規則は，

合成規則名(合成規則ID, 条件部, 実行部).

の形式で記述される．第 1 引数は，合成規則の ID 番号で合成規則管理部が使用する．条件部，実行部ははそれぞれ，

条件部 : [(e_1,[v_{11},v_{12},…]), (e_2,[v_{21},v_{22},…]), …]

実行部 : [$func_1$, $func_2$, …]

のようなリストで記述されている．ここで，e_iは環境素で，音節データにおけるものと同様である．またv_{ij}は環境素e_iの一つの属性値で，音節データにおける環境と異なり合成規則の条件部では各環境素に対して属性値のリストが記述される．また，実行部には合成パラメータを操作するためのパラメータ操作関数$func_i$が記述され，入力テキストから得られる音節データに対して条件部が適合すると，この実行部の関数が順次実行される．(実際にはPrologの述語として実現されており，関数値をとらない副作用だけを行なう手続きであるが，直観的理解を容易にするために関数と呼ぶ．) SSRIでは，音節データに対して，条件部の適合した全ての規則が発火する．従って，実行可能規則の間での競合解消は必要ない．

(3) パラメータ操作関数

(a) 継続時間長

一般にホルマントなどの合成パラメータは，子音部では細かく変動し，母音部では比較的変化が少ない．また，合成規則はパラメータ値の変化をできる限り簡潔に記述できることが望ましい．しかし，パラメータ制御の自由度が少なすぎると，合成音の品質劣化につながる．これらの事情を考えて音節を7個のセグメントに分割し，通常はこのセグメント境界のパラメータ値を操作し，必要に応じてセグメント内での変化も記述できるようにしている．子音部はパラメータの変動が大きいため4個のセグメント(c0, c1, c2, c3)に分割し，母音部は前後の音素との接続を考え3個のセグメント(v1, v2, v3)に分割する．時間長を設定するために，

絶対操作関数：区間名(時間長)
相対操作関数：区間名(符号，変化量)

の二種類のパラメータ操作関数が用意されており，引数の数で区別される．ここで，関数名として用いられる区間名は7個のセグメント，あるいはcv，c，vのいずれかで，それぞれ，セグメント長，音節全体，子音部，母音部の時間長を設定する．絶対操作関数はその区間の時間長を引数の値にセットし，相対操作関数はそれまでにセットされた値に対して相対的に時間長を増減させる．符号には算

術記号 +, -, * のうち一つが用いられる．図3.18 にこれらの関数を用いて定義される合成規則の例を示す．(a), (b) がそれぞれ継続時間長，パワーに関する合成規則の例である．例えば，(a) の 106 番の規則では音節の時間長を設定する関数 cv が呼ばれ，音節 /ka/ の時間長が 253ms に設定される．また 500 番の規則は「語頭で，子音部がない」，即ち単母音で始まるテキストの先頭の音節では，時間長を 0.75 倍することを示している．

(b) 他の合成パラメータ

ホルマント周波数，バンド幅，エネルギーおよび基本周波数の操作関数は，通常は各区間の始点において各パラメータ値をセットする．(区間は，ある一つのセグメント，あるいは連続したいくつかのセグメントである．) しかし，更に細かくパラメータ値を制御しなければならない場合にも対応できるよう，区間の始点から任意の時間後において，パラメータ値を与えられるような形式も用意されている．ホルマント等に対するパラメータ操作関数は，

絶対操作関数 1: *Para* (区間, 値)
　　2: *Para* (区間, 時間, 値)
相対操作関数 1: *Para* (区間, 符号, 値)
　　2: *Para* (区間, 時間, 符号, 値)

の四種である．ここで，関数名 *Para* は設定するパラメータを表している．符号および絶対操作関数と相対操作関数に関しては，時間長に関するパラメータ操作関数と同様である．それぞれ 1 の型の関数では，区間の始点において引数に応じて値が設定される．また，2 の型の関数には区間の始点からの時間 (ms) が与えられ，一つの区間内での細かいパラメータの変化も実現できる．

(c) 合成素片ファイル

合成パラメータ値が細かく変化する場合には，前項で示した規則の表現形式で合成パラメータの変化を表現することは容易ではない．また，無理にこのような表現形式に従い合成規則を記述することは，後に規則の修正等を行う時にその規則の可読性を低下させる．このため，合成素片ファイルを用い各パラメータ値を

```
def_cv_dur( 106, [(c,[k]),(v,[a])], [cv(253)] ).
def_cv_dur( 107, [(c,[k]),(v,[a])], [cv(259)] ).
mod_cv_dur( 500, [begin,(c,[#])], [cv(*,0.75)] ).
mod_cv_dur( 510, [begin,(pitch,[low])],[cv(*,0.96)] ).
mod_cv_dur( 520, [begin,(pitch,[high])], [cv(*,1.03)] ).
def_c_dur( 210, [], [c0(5),c1(20),c2(20),c3(20)] ).
def_c_dur( 215, [(c,[nn])], [c0(5),c1(15),c2(80),c3(20)] ).
def_c_dur( 220, [(c,[b,d,dh])], [c0(0),c1(25),c2(20),c3(10)] ).
def_c_dur( 230, [(c,[g])], [c0(0),c1(25),c2(25),c3(10)] ).
def_c_dur( 240, [(c,[s,sh])], [c0(0),c1(20),c2(30),c3(30)] ).
def_c_dur( 260, [(c,[k])], [c0(0),c1(20),c2(20),c3(30)] ).
def_c_dur( 270, [(c,[#,q])], [c0(0),c1(0),c2(0),c3(0)] ).
def_v_dur( 310, [], [v1(20),v3(20)] ).
def_v_dur( 311, [(c,[q])], [v1(0),v3(0)] ).
def_v_dur( 315, [(c,[nn])], [v1(5),v3(5)] ).
def_v_dur( 9310, [(c,[f])], [v1(8),v3(9)] ).
def_v_dur( 8310,  [(c,[j])], [v1(70),v3(31)] ).
def_v_dur( 7310, [(c,[k])], [v1(20),v3(20)] ).
```

(a)　継続時間長に関する規則

```
def_energy( 7010, [(c,[nn])], [energy(c,33)] ).
def_energy( 7011, [(c,[uStop])], [energy(c,0)] ).
def_energy( 7012, [(c,[q])], [energy(c0,10),energy(v,0)] ).
def_energy( 60015, [(v,[a])], [energy(v1,40),energy(v23,43)] ).
def_energy( 60025, [(v,[i])], [energy(v1,40),energy(v23,42)] ).
def_energy( 60035, [(v,[u])], [energy(v,33)] ).
def_energy( 60045, [(v,[e])], [energy(v,53)] ).
def_energy( 60055, [(v,[o])], [energy(v,40)] ).
mod_energy( 60350, [begin,(c,[#])], [energy(v1,-,15)] ).
mod_energy( 71010, [begin,(pitch,[low])],  [energy(v,+,1)] ).
mod_energy( 71020, [begin,(pitch,[high])], [energy(v,+,3)] ).
mod_energy( 71070, [end,(pitch,[low]),(pitch_V,[no])],
        [energy(v,-,7)] ).
mod_energy( 71080, [end,(pitch,[high]),(pitch_V,[down])],
        [energy(v,-,3)] ).
```

(b)　パワーに関する規則

図3.18　合成規則の例

決定する規則が用意されている．合成素片ファイルでは，子音部の区間 (主にc2,c3) における全ての合成パラメータの5msごとの値が記述されている．次の関数

file(区間, 合成素片ファイル名)

は特殊なパラメータ操作関数であり，その区間における全ての合成パラメータを合成素片ファイル中のデータで置き換える．ファイル中のデータが区間より長い時には切り捨てられ，短い時には最後のデータから次の区間までがパラメータ補間部によって直線補間される．

3.5.2 合成規則管理部

(1) 履歴データの管理

合成規則管理部RMは，再合成する場合に備えて，入力テキストごとに発火した合成規則IDを履歴データとして保存する．さらに，合成規則の修正が行なわれると，音節データに対する規則の発火状態も一般に変わるためRMは履歴データの修正を行なわなければならない．ここで，履歴データは次の四つのデータ形式によって保持される．

pre_text(*TID*, *Text*).

rule_id(*TID*, *SID*, *Para*, *FiredRidList*).

del_rid(*TID*, *Para*, *DelRidList*).

add_rid(*TID*, *Para*, *AddRidList*).

ここで，*TID*, *SID* はそれぞれ入力されたテキストおよび音節のID番号である．Para は合成パラメータの種類を表しており，dur, fb, file, pitch, energy のいずれかである．また，*FiredRidList*, *DelRidList*, *AddRidList* は合成規則のID番号のリストである．新しくSSRIにテキストが入力されRIでの合成規則の適用が完了すると，RM は pre_text, rule_id のデータを作成し，履歴データベースに格納する．発火した合成規則は，音節およびパラメータごとに保持される．

合成ルールベース修正後の履歴データの更新は，合成規則が修正されるたびに全てのrule_idデータを正しく更新する方法でも実現できるが，その場合には多くのテキストが入力されるにつれて履歴データの更新に時間がかかるようになる．そこで，rule_idのデータはそのままにしておき，そのテキストの合成が行なわれてから以降に削除された規則および追加された規則をそれぞれdel_rid，add_ridによって保持しておく．部分的に変更された規則は，削除されかつ追加されたものとして処理される．したがって，履歴データでは発火すべき規則ではなく，発火する可能性のある規則を保持しており，再合成を行なう段階でその規則に対して条件部のテストを行なう．再合成が行なわれるとそのdel_ridおよびadd_ridは消去される．

(2) 合成規則の分類

履歴データの管理は，再合成の場合に処理時間を短縮できるが，新しいテキストが入力された場合(以下，新規合成と呼ぶ)には何も貢献しない．そこで，合成規則の条件部を用いて予め合成規則を分類したデータを作成し，新規合成時の規則の検索範囲を制限することによって規則の適用時間を短縮する．多くの合成規則，特にホルマントに関する合成規則は主にCV音節ごとに記述されており，また，語頭や語尾に関する条件を含む合成規則も多い．このため，子音，語頭，語尾の三つの環境素に関して，合成規則を次の四種にあらかじめ分類しておく．

(a) 子音を条件に含む合成規則
(b) (a)以外で語頭を条件に含む合成規則
(c) (a)以外で語尾を条件に含む合成規則
(d) (a),(b),(c)以外の合成規則

(a) では，さらに子音名ごとに分類される．規則の適用時には，音節データの子音，語頭，語尾に関して条件の一致する分類の規則に対してのみ条件部をテストする．分類 (d) では全ての規則がテストされる．

(3) 合成規則の検索

RM では，ID 番号や条件による合成規則の検索が行なえる．規則の ID 番号を入力するとその ID 番号の規則の条件部・実行部が表示され，また，条件を (規則の条件部の形式で) 入力すると，その条件を満たす条件部を持つ規則が表示される．

3.5.3 SSRI の評価

(1) SSRI の実行時間

SSRI での実行時間を計測した．RI での実行時間は，規則の数に大きく依存するため，現在の合成ルールベース中の合成規則数を表3.2 に示す．基本周波数の規則が少ないのは，これまで主に単語データを対象として支援環境および音声合成ルールベースの構築を行なってきたためである．

表3.3 は，テキスト (あるいは再合成のコマンド) が入力されてから RI が処理 (規則の条件部をテストし，満たされれば実行部を実行する) を終了するまでに要した時間を示している．なお，括弧内に記した値は，RI での処理に加えてパラメータの補間および合成器での処理時間も含めた，実際に合成音が生成されるまでに要する時間である．表3.3 (a) から合成規則の分類を行なうことによって，新規合成において規則の適用時間を約 1/4 に短縮できていることがわかる．表3.3 (b)

表3.2　音声合成ルールベース中の規則数

パラメータの種類	規則数
時間長	131
ホルマント・バンド幅	148
基本周波数	18
パワー	130
素片ファイル	307

表3.3　合成規則インタプリタでの実行時間の比較

(a) 新規合成における実行時間 [秒]

テキスト番号	規則の分類	
	未使用	使用
1	18.2 (23.4)	4.8 (10.1)
2	22.9 (31.0)	6.0 (14.0)

(b) 再合成における実行時間 [秒]

テキスト番号	新規合成*	再合成1	再合成2
1	4.8 (10.1)	0.7 (4.8)	1.8 (7.1)
2	6.0 (14.0)	0.9 (5.7)	3.3 (11.3)

(* 規則の分類を使用)

テキスト1: /koNnichiwa/ (こんにちは)
テキスト2: /chirinobeNkyo-/ (地理の勉強)

は再合成時の処理時間を示している．ここで，再合成1は合成規則修正の直前に入力されたテキストからの再合成，再合成2は，以前に合成を行なったテキストからの再合成を意味する．RIおよびパラメータ補間部は，記憶容量の都合から直前の処理結果のみを保持するため，再合成2ではすべてのパラメータに対して規則の適用および補間を行なわなければならない．再合成 1 では修正されたパラメータのみに対して処理を行なう．ここでは，合成規則の修正としてパワーの規則の修正 (一箇所) を行なった．表3.3 (b) より，規則修正および同一テキストからの音声合成を繰り返す場合には (再合成1)，RI での処理にはほとんど時間がかからないことがわかる．これらの実行時間は，20MIPS のワークステーション上で計測した．

表3.4　SSRIを用いた都市名単語に対するルールベース修正の内訳

パラメータの種類	削除	変更	追加
時間長	46	52	5
ホルマント・バンド幅	2	29	11
パワー	0	39	0
素片ファイル	22	0	12

(2) SSRIを用いた合成ルールベースの修正

SED-SSRBの中で，REMおよびPOMのモジュールは帰納的学習に基づいているため，あらかじめ正解となるデータを用意しなければならない．時間長規則の場合には，音声データにラベル付けすることにより比較的簡単に学習データを用意することができるが，ホルマントなどのパラメータの場合には簡単ではない．結局，ホルマントを用いたパラメータ導出型の音声規則合成では試行錯誤の処理が必要になってくる．そこで，SSRIを用いて単語音声に対する合成音の品質改善を試みた．対象とした単語は3.4.3で用いた100都市名のうちの80都市名である．単語了解度試験の被験者が関西在住者のため，「市原」などの関西在住者になじみの薄い都市を除いた．

一人の音声合成の専門家が，SSRIを用いて約15時間にわたりの音声合成ルールベースの修正を行なった．行なわれた修正の内容を表3.4に示す．修正作業を評価するため，単語了解度試験およびオピニオン試験を行なった．被験者は，合成音を聴取した経験のない3名である．その結果，規則修正によって単語了解度は78%から93%に，オピニオン平均値は，2.3から3.0にそれぞれ向上した．この結果から，SSRIを用いることにより比較的短時間で効果的な音声合成ルールベースの修正が行なえたことがわかる．

3.6 結言

規則合成方式，特にホルマントを規則によって導出する方式で生成される合成音の品質を向上させるためには，合成規則の集まり，すなわち音声合成ルールベースを充実させていかなければならない．このためには計算機を利用してルールベースの構築を支援することが重要であるという立場から，処理の自動化のための合成規則自動抽出と修正案の提示，および作業サイクルを短縮するためのインタプリタに基づいた音声合成システムについて検討した．これらの機能を統合的にまとめた環境が音声合成ルールベース構築支援環境 SED-SSRB であると言える．

合成規則の自動抽出および修正案の提示では，合成パラメータとして単語中の CV 音節時間長を取り上げて本章で述べた手法の有効性を確認した．自動抽出された合成規則は専門家の知見とも合致し，それを用いることによって非学習データに対しても誤差が大幅に減少した．このような自動抽出を行うことにより，専門家の膨大な労力を必要とすることなく合成規則が得られる．参照したデータに依存しない一般性の高い規則を得るためには，より多くのデータを参照することが必要であり，自動抽出はそのような大量のデータからの規則抽出に有利であると考えられる．また，修正案の提示を利用することによって比較的短時間で，合成規則の修正・追加などを行ない合成ルールベースの性能を改善することができた．人手による規則の修正では，どの規則をどのように修正すればよいかを判断することは容易ではないため，計算機によって適当な修正案を提示することが非常に有効な支援になる．

合成規則インタプリタに基づいた音声合成システムでは，音声合成ルールベースを音声合成における実行部から分離し発火規則を管理することによって，極めて短時間で再合成を実行でき，効率良くルールベースの改善が行なえることを示した．音声合成ルールベースの構築では，ルールベース修正後にその効果をすぐに評価できることが重要であり，修正・評価・修正のサイクルを短縮することにより有効な支援が行なわれた．

自動抽出・修正案の提示の実験ではパラメータの数値の上での評価を行なったが，合成音の品質としてどの程度の改善が達成されているかを知るためには，今後聴取実験などを通して評価していく必要がある．また，パラメータとしては3.3.3(5) でも述べたように，扱いやすさの点からまず CV 音節時間長を用いて実験を行なったが，ピッチやホルマントなど時間長以外のパラメータに関する規則に対しても本手法を適用すると同時に支援機能の拡充を行なう必要があると思われる．

第 4 章

概念表現からの音声合成

4.1 緒言

音声規則合成技術を入力として与えられるデータ形式によって分類すると，2.3.3 でも述べたように，テキストとして書かれた文章などの自然言語表現を音声に変換する TTS (Text-to-speech) と，自然言語表現以外の意味表現を音声に変換する方式である CTS (Concept-to-peech) の二種類に大きく別けることができる．一般に，情報の中には自然言語表現を用いて記述されているものも多く，入力された新聞記事の校閲支援[河合87]などの応用では TTS が重要な役割を果たす．しかしながら，音声対話を実現し問題解決器からの音声出力を行なう場合には，必ずしも問題解決器が自然言語表現で出力を生成する必要がないことから，音声合成システムへの入力として最適なデータ形式について考察する余地が残されている．問題解決器と音声合成システムとの情報伝達形式を核にして，システム論的立場から音声出力システムを再検討することによって，TTS に基づく音声出力方式の問題点が明らかになるとともに，CTS 方式に基づく方式の必要性と利点とを示すことができると思われる．

CTS に関する先駆的研究として Young らによる SSC システム[Young79]があり，国内でもいくつかの試みがなされている[山本85][坂井90][藤崎90]．CTS の利

点の一つは韻律的特徴の生成の容易さにあるが，YoungらのSSCや浅野等のシステム[藤崎90]では入力概念を利用した韻律的特徴の直接的生成が行なわれておらず，CTSの利点が十分に生かされていなかった．山本等の研究では，文生成が行なわれた後で，テキストからの音声合成が行なわれる[山本85]．また，坂井等は機械翻訳における中間表現を利用しており[坂井90]，特定の応用分野での利用が期待できるが，一般の問題解決における音声出力という観点からは適用可能性に問題が残る．このように，従来の研究ではCTSの長所を十分に活用した本格的なシステムは提案されていない．

本章では，従来他の関連分野とは独立に議論される傾向が強かった音声合成技術の問題を，問題解決器との相互作用をも考慮に入れた音声インタフェースの重要な構成要素として捉え[山下89][山下90]，問題解決器からの音声出力を実現するための汎用的なインタフェースシステムの枠組，およびその中でのCTSの設計思想と基本アーキテクチャの設計について述べる．4.2では，機械と自然な対話を実現するための音声インタフェースにおいて，特に，音声出力を実現するための柔軟で汎用性のある枠組について述べ，問題解決器と音声合成システムの観点から汎用インタフェースへ情報を伝達するための記述形式について論じる．4.3では，4.2での考察をもとにインタフェースへの入力となる概念表現を具体的に定義する．さらに4.4では，概念表現を利用した韻律生成メカニズムを持つCTSシステムであるSOCS (Speech Output from Case Structure representation) の基本アーキテクチャについて述べる．

4.2　音声出力インタフェースの設計

4.2.1　音声出力インタフェースの設計思想

「音声による機械との対話」を人間同士の対話のように自然に行なえるようにするためには，多くの問題を解決しなければならない．ここで，対話に関連した処理として，利用者の意図の把握，意図を伝達するための最適な表現の決定，語

句の省略の同定・補充などの問題が挙げられる．さらに，音声理解を助けるための次発話の予測，不可避的に発生する音声認識誤りや合成音の聞き誤りへの対応，出力音声の韻律的特徴の生成など，自然言語など他の媒体には見られない音声特有の問題も解決しなければならない．これらは対話あるいは音声対話に固有の問題であって，問題解決器に依存するものではない．従って，機械との音声対話を行なう時，このような問題は個々の問題解決器内部ではなく，図4.1 に示すような対話管理機能を持つ汎用の音声入出力インタフェースで対処すべきであると考えられる．以下では，特に音声出力について考察し，インタフェースにおいて音声出力を行なう部分を音声出力インタフェースと呼ぶ．

問題解決器から音声出力インタフェースへのメッセージ(出力内容の記述)が自然言語表現で与えられ，TTS によって音声に変換される場合には，

1) 韻律生成に必要な係り受け構造などを抽出するための入力テキストの解析において，解析誤りが合成音の自然性を大きく損ねることがある．

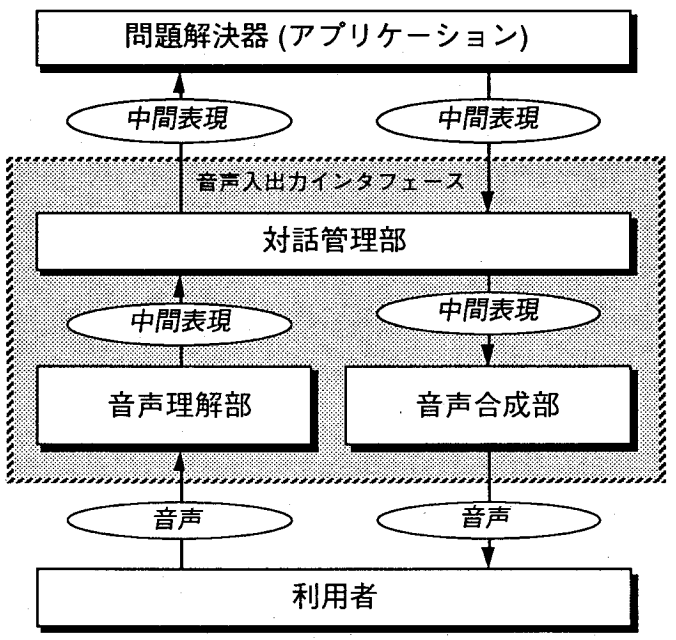


図4.1　音声入出力インタフェース

2) インタフェース側で自然言語表現を操作することは容易ではないために，対話コンテキストに最適な表現の選択や強調などの韻律的特徴の抽出を全て問題解決器が行なわなければならなくなる．

などの問題点がある．逆に，CTS に基づいた音声出力では，

1) 自らが文生成を行なうため誤りのない統語構造を利用することができる．
2) インタフェースへの入力が意味表現として与えられるため，柔軟に表層文の変更が可能になる．

という利点がある．以上の考察をまとめると，機械との自然な対話を行なうための音声出力インタフェースは，次のような条件を満たすべきであると考えられる．

(a) 汎用的なインタフェース

問題解決器のドメインに依存しない汎用なものであることが必要である．様々な種類の問題解決器に適用できるインタフェースを利用することによって，問題解決器の構築が容易になる．これは問題解決器に依存した知識を使わないということではなく，音声出力インタフェースの中でそのような知識を汎用的な部分と切り分けた枠組にするということを意味する．

(b) 適切な中間表現の設定

問題解決器が「言いたいこと」を記述する表現形式を中間表現と呼ぶことにする．この中間表現はインタフェースへの入力でもあり，中間表現をどのような表現形式にするかはインタフェース内部での処理の容易さや，出力表現の多様性と密接に関わっている．中間表現に関しては 4.2.2 で詳細に検討する．

(c) インタフェース内部での緊密な処理

インタフェース内部では，対話管理部，文生成部，音声合成部などの複数のモジュールが必要となり，各モジュールの適切な役割分担とともに，例えば文生成からの韻律制御情報の獲得など，モジュール間の緊密な情報交換をうまく行なわなければならない．

4.2.2 音声出力インタフェースへの入力表現形式

問題解決器から音声出力インタフェースに与えられる発話内容の表現法には様々な形式が考えられるが，汎用的な音声出力インタフェースを構築するには適切な抽象度レベルの中間表現を定める必要がある．音声出力インタフェースから見ると音韻列や韻律的特徴などの音声表現を入力から決定しやすい方が良く，問題解決器にとっては内部の処理結果から出力を生成しやすい方が都合が良い．このように問題解決器と音声インタフェースとで都合の良い中間表現が異なるため，両者の観点から適切な中間表現を考えてみる必要がある．

(1) 中間表現に対する要求

音声出力インタフェースおよび問題解決器からの中間表現に対する要求として，

＜要求1＞　韻律生成に必要な出力文の構文情報を，容易にかつ誤りなく獲得できること．誤った構文情報から韻律的特徴が生成されると合成音の自然性が大きく損なわれることが多い．

＜要求2＞　文生成が容易に行なえること．

＜要求3＞　対話管理が容易に行なえること．語句の省略や補充，表現の選択・決定，強調や焦点の抽出など，最適な音声出力を行なうための対話管理処理が容易に行なえることが必要である．

＜要求4＞　問題解決器の中間表現生成部をインプリメントする際に，中間表現作成のためのデータを容易に作成できること．

＜要求5＞　少ないデータから多様な表現を生成できること．同じ中間表現から異なる表現や異なる韻律的特徴の音声を生成する機能は，対話管理部において対話のコンテキストに応じた文や韻律的特徴の制御を行なうために不可欠である．このような出力の多様性を音声インタフェースに委ねることによって，問題解決器は「言いたいこと」だけを決めれば良い．

＜要求6＞　問題解決器が推論・処理結果から実行時に容易に中間形式を生

成できること．

などが挙げられ，＜要求1～3＞が音声出力インタフェース側からの要求，＜要求4～6＞が問題解決器側からの要求である．

また，問題解決器側における中間表現生成の容易さは，中間表現の生成メカニズム(すなわち，問題解決器がどのように「言いたいこと」を決定するか)によって異る．これは，大きく分けて，

(i) 問題解決器での推論・処理結果(以下，単に結果)がカテゴリカルに分類され，それに応じて予め用意されているパタンを単に選択する．

(ii) 変数を含む出力パタンが一つ用意されており，結果によってそれが穴埋めされる．

(iii) (i) と (ii) の複合型で，結果によってパタンの選択と穴埋めを行なう．

(iv) 用意されているパタンの選択，穴埋めに加えて，パタンを組み合わせたり，修正を行なう．

の四つに抽象化でき，それぞれのメカニズムによって都合の良い中間表現も異なる．

(2) 様々な中間表現

ここでは，中間表現として図4.2に示すような六つの表現形式を対象として考察する．これらを簡単に説明すると，

REP1-1: 自然言語表現．

REP1-2: 音韻記号と韻律記号．音声合成にとって最も都合の良い表現．

REP2-1: 文パタンに単語を埋めこんだ表現．

REP2-2: 生成文の構文木．ここでは，文脈自由文法が用いられている．

REP3: 格構造表現．一種の意味ネットワークである．

REP4: CD-表現[Schank75]．11種類に抽象化された動詞表現が用いられる．

である．

花子は太郎から車を買った

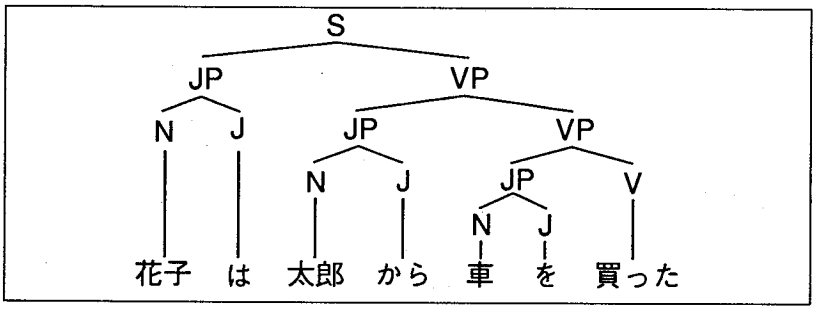
REP1-1: 自然言語表現

ha'nakowa,ta'ro-kara/kurumawo/kaQta

REP1-2: 音韻記号

$buy1(花子, 車, 太郎). $past.

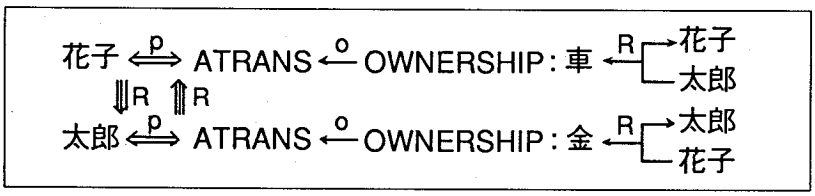
REP2-1: 文パタン



REP2-2: 構文木

買う([$actor(花子),$object(車),$from(太郎),$past]).

REP3: 格構造表現



REP4: CD表現

図4.2　様々な表現形式

(3) 表現形式の比較

(1) で述べた観点から図4.2 に示した六つの中間表現を比較した結果を表4.1 に示す．これは各要求ごとに相対的に比較したもので，要求を満たすことが ◎，○，△，× の順に容易であることを示している．また，＜要求6＞は問題解決器における出力生成メカニズムごとに比較した．

REP1-1, REP1-2 はそれから文生成を行なう必要がない替わりに，対話管理部において問題解決器の生成した中間表現を修正することが難しい．また，REP1-1 ではテキストの構文解析や読みの決定を行なわなければならず，REP1-2 では韻律など音声に関する処理を問題解決器に強いることになる．

REP2-2 も構文解析の必要はないが，対話管理による表現の修正は簡単ではない．REP2-1 は文パタンによって高品質の文，韻律的特徴の生成が期待できる反面，非常に多数の文パタンが必要になる．

REP3, REP4 では文生成を行なう必要があるが，その反面正しい構文情報を利用して韻律パタンを決定することができる．(構文情報は，対話コンテキストなどを利用してインタフェース内部において最終的に決定される．ただし，本論文で定

表4.1　種々の表現形式の比較

要求	表現形式					
	REP1-1 (自然言語)	REP1-2 (発音記号)	REP2-1 (文パタン)	REP2-2 (構文木)	REP3 (格構造)	REP4 (CD表現)
要求1	×	◎	○	○	○	○
要求2	◎	◎	△	○	△	×
要求3	×	×	○	○	◎	◎
要求4	○	△	△	△	△	×
要求5	×	×	△	△	○	◎
要求6-i	○	○	○	○	○	○
要求6-ii,iii	○	×	○	○	○	△
要求6-iv	×	×	○	△	◎	◎

義する概念表現のように入れ子構造を持つ再帰的表現の場合には，それ自体に構文情報が埋め込まれていると言え，インタフェースの外部からも構文情報が一部与えられることになる．) また，これらの表現では意味構造も与えられるため，対話管理における表現修正も比較的容易に行なえる．例えば，REP3, REP4 では「動作主が前の発話と同じだから省略する」といった処理が容易に行なえる．REP4 では文生成時に表層文 (特に，述語表現) に用いる語の選択を行なう必要がある．問題解決器から多言語の音声出力を考える時にはこのような抽象度の高い表現が必要であるが，単一の言語による出力を考える時には，文生成の複雑さが欠点となる．

＜要求6＞に関して比較すると，出力生成メカニズムが (i) である場合 (表4.1 中の＜要求6-i＞) は，あらかじめ用意されているパタンが単に選択されるだけなので，パタンをどのような形式で記述しておいても差はない．(ii) と (iii) では，REP1-2 では穴埋めされる単語によって，ポーズなどの位置を変える処理が必要となる．REP4 では，動詞の穴埋めなど，穴埋めされる表現 (単語) によって用意すべき抽象的表現 (パタン) が異なってくることがある．(iv) の場合，REP1-1, REP1-2, REP2-2 では，パタンの修正，接続が行ないにくいが，REP3, REP4 では，パタンの接続も比較的容易に行なえる．

CTS システムへの入力となる中間表現としてどのような表現が最適かは一概には決められないが，以上の検討の結果 REP3 がよく要求を満たしていることがわかった．中間表現の抽象度レベルの観点から見ると，REP3 はユーザに提示される表層文で用いられる語は決定されているが，その語順は与えられておらず，構文の変更も容易であることが一つの特徴である．そこで，中間表現として，格構造に基づいた表現に一部文パタン (慣用テンプレートと呼ぶ) を合わせて用いる表現 (すなわち，REP3 + REP2-1) を用いることとした．以降ではこれを概念表現と呼ぶことにする．

4.3 概念表現

4.3.1 概念表現を利用した音声出力インタフェース

4.2.2 で述べた要求仕様を満たす汎用的な音声出力インタフェースとして図4.3に示すインタフェースシステムを設計した．インタフェースへの入力は，4.2.2 で決定した概念表現 (詳細は 4.3.2 で述べる) を用いて問題解決器によって記述される．さらにインタフェース内部では，概念表現は対話管理部によって対話のコンテキストを考慮することにより修正され，SOCS (Speech Output from Case Structure representation) システムへの入力となる．この二つの概念表現は基本的には同じ表現形式で記述されるが，次章で述べるように，問題解決器の生成する概念表現では韻律や慣用テンプレートに関する記述が必要ないという点で，SOCS に入力される概念表現とは内容が異なる．この二つを特に区別する場合には，図4.3 に

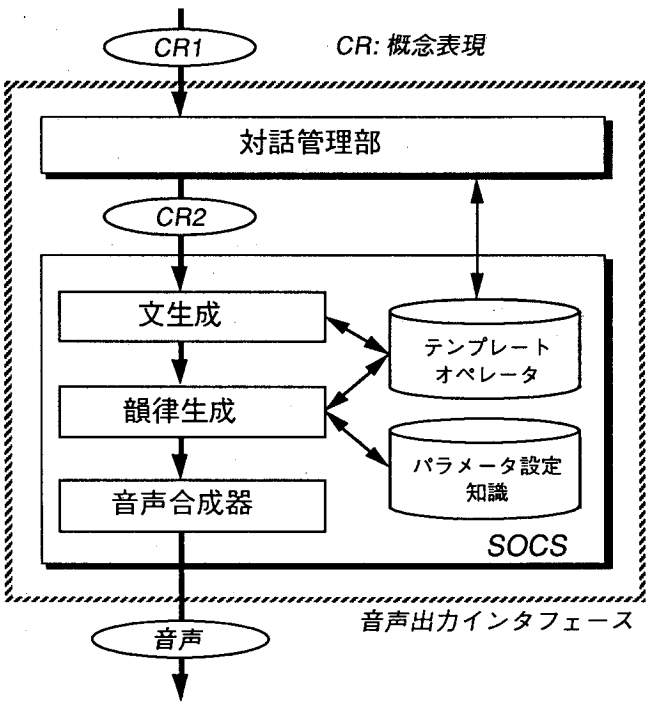


図4.3　SOCS を用いた音声出力インタフェース

示すように問題解決器が生成する表現，SOCSへの入力となる表現をそれぞれ概念表現1 (CR1)，概念表現2 (CR2) と呼ぶ．SOCS は入力された概念表現2 を合成音に変換するCTS システムであり，その基本アーキテクチャは4.4 で述べる．

4.3.2 概念表現の構成要素

4.2.2 での検討結果に基づいて定義された概念表現の構成要素に関して述べる．まず，問題解決器の生成する概念表現1は以下に示す七つの構成要素によって記述される．

(1) アトム

表層文で用いられる名詞，動詞，形容詞，形容動詞，副詞などはアトムと呼ばれ，概念表現中にそのまま記述される．

(2) 係り受けテンプレート

係り受けテンプレート $modify は，単語，句，節などによる修飾関係を表す．これは表4.2 に示すように，第1引数が名詞で終る場合には，助詞「の」をはさんで二つの引数を出力し，活用語で終る第1引数が第2引数の名詞 (句) を修飾する場合には，第1引数の終りを連体形に変え第2引数に接続し，それ以外の場合(例えば，副詞が形容詞を修飾する) にはそのまま二つの引数を接続する．ここで，記号p1 および後述するテンプレートで現われるp2 はポーズマーカーと呼ばれる一種のマーカーで，4.4 で述べるように，ポーズの挿入などに用いられる．以下のテンプレートにおいても記号 p1, p2 の意味は同様である．

表4.2　係り受けテンプレート

$modify(A,B):	[A, の, p1, B]. /Aが名詞で終る時/
	[A*, B]. /Aが活用語で終る時/
	[A, B]. /その他/
	(A*: 末尾の活用語を連体形に変えたもの)

表4.3　格テンプレート

対象関係	$actor(A):	[A, が, p1].
	$object(A):	[A, を, p1].
	$comparison(A):	[A, と, p1].
	$subject(A):	[A, に, p1].
方法関係	$instrument(A):	[A, で, p1].
	$means(A):	[A, に, よって, p1].
	$cause(A):	[A, の, ため, に, p1].
方向関係	$from(A):	[A, から, p1].
	$to(A):	[A, まで, p1].
	$direction(A):	[A, へ, p1].
	$purpose(A):	[A, ため, に, p1].
	$result(A):	[A, に, p1].
時空関係	$place(A):	[A, で, p1].
	$time(A):	[A, p1].
付帯関係	$restriction(A):	[A, で, p1].
	$content(A):	[A, に, ついて, p1].
形容関係	$times(A):	[A].
	$degree(A):	[A].

表4.4　法オペレータ

相	$progressive:	～ている
	$perfect:	～てしまう
時制	$past:	～た
	$future:	～だろう
判断・態度	$negative:	～ない
	$possible:	～できる
	$desirable:	～たい
	$imperative:	～なさい
	$interrogative:	～か
	$permissive:	～てよい
	$persuasive:	～う
	$state:	～ている
	$passive:	～れる、られる

(3) 格テンプレート

格テンプレートは格情報を表し，概念表現として与えられる述語の引数として記述される．表4.3 に示すように 18 個の格テンプレートが定義されている．格テンプレートでは，文生成時に用いられる助詞等がポーズマーカーと共に記述されている．

(4) 法オペレータ

法オペレータは述語の法情報を表し，格テンプレートと同様に述語の引数として記述される．表4.4 に示すように 13 個の法オペレータが定義されている．文生成時には，ここで記述される法情報に従って動詞等の語尾が変更される．

(5) 接続テンプレート

概念表現は一文ごとに記述される．そこで，直前の文とのつながりを表現するために接続テンプレートを用いることができる．このため，このテンプレートには表4.5 に示されるように接続詞が記述されている．

(6) 慣用テンプレート

一般に対話によく現れる慣用的な表現や問題解決器によく現れる表現などは，表4.6 に示すような慣用テンプレートを用いて記述することができる．慣用テンプレートには，出力される単語とポーズマーカーの他に韻律的な特徴を制御するために韻律修正関数が記述される．$reasonテンプレートの $mod_acc や $why テンプレートの $mod_dur，$mod_bpit が韻律修正関数である．4.4 で述べるように，韻律修正関数は慣用テンプレートの出力リストの引数に対し，時間長，話調の開

表4.5　接続テンプレート

$conversion(A):	[それでは, p2, A].
$addition(A):	[ですから, p2, A].
$pos_cond(A):	[従って, p2, A].
$neg_cond(A):	[しかし, p2, A].

表4.6　慣用テンプレート

$is(A,B,OPs):	[A, は, p1, B, です].
$because(A,B,OPs):	[A, から, p2, B], $mod_bpit(4,190).
$reason(A,B,OPs):	[A, は, p2, B, から, です], $mod_acc(4,+,15).
$and(A,B,OPs):	[A, と, p1, B]. /Aが名詞で終る時/
	[A*, p2, B], $mod_bpit(3,-,10). /Aが活用語で終る時/
	(A*: 末尾の活用語を連用形に変えたもの)
$why(A,OPs):	[なぜ, p2, A, の, です, か],
	$mod_dur(1,+,10), $mod_bpit(3,190).
$where(A,OPs):	[A1, p2, どこ, で, p1, A2, の, です, か], $mod_dur(3,+,10),
	$mod_acc(3,+,15), $mod_bpit(3,200).
	(A1:Aの格要素, A2:Aの述語)

始ピッチ周波数，アクセントの大きさを決定する．このように独立のテンプレートを用意することにより，よく用いられる表現における韻律パラメータを容易に生成することができる．また，最後の引数OPsには，必要に応じて疑問などの韻律オペレータが記述される．次章で述べるように，対話管理部においてテンプレートを用いた表現への書換えを行なうため，このテンプレートは必ずしも概念表現1で用いられる必要はない．

(7) 強調オペレータ

問題解決器が強調したい意味内容を含むメッセージを生成したい場合に$emphasis(A)を用いて，それを記述する．

さらに，対話管理部による修正後の概念表現2では，概念表現1で用いられる構成要素に加えて，次の韻律オペレータが用いられる．

(8) 韻律オペレータ

韻律オペレータは，述語の引数や他のテンプレートの引数など概念表現の一部として韻律的特徴を指定するために用いられる．　$p_prominent(A),

$p_interrogative, $p_speed(A,S) の三種類のオペレータが定義されており，4.4 で述べるように韻律的特徴が生成される．

以上で定義した各テンプレートおよびオペレータは，慣用テンプレートを除いて，文の構成や韻律的特徴の指示において基本的に必要なものばかりで，問題解決器に依存していない．また，慣用テンプレートのうち対話によく現われる慣用的な表現を表すものも対話に固有であり問題解決器には依存していない．(表4.6にしたテンプレートは全てこのようなテンプレートである．) 慣用テンプレートとして，特定の問題解決器でよく用いられる表現を記述しておくことができるが，このような慣用テンプレートと語彙に関するアトムが問題解決器に依存している．また，問題解決器は多様な内容を出力すると考えられ，概念表現はこれらの出力内容を全てカバーしなければならない．現在までに，案内をタスクとした比較的短い発話から構成される対話においては，ほとんどの発話が上で述べた構成要素で記述できることを確認している．しかしながら，複雑な情報伝達を行なうような対話 (例えば，教育など) では，必ずしも十分ではないと思われる．このような対話では，新たな慣用テンプレートを定義することが必要となろう．

4.3.3 概念表現例

図4.4 は概念表現の例で，(a) の s1 から s6 までが概念表現1 を表している．(b) は，それに対して対話管理部によって韻律に関するオペレータや慣用テンプレートの追加などが行なわれた概念表現2を表している．s1, s3 に関しては，対話管理部での修正を受けずに SOCS へ入力される．また，s1 に現れる lesson1 や s3 に現れる region1，next はラベルで，任意に用いることができる．

すでに述べたように，問題解決器では基本的には「言いたいこと」のみを決定すべきであるから，対話処理によって獲得できる情報や韻律に関する操作は概念表現1に記述しなくても良いような枠組が必要である．このような観点から，音声出力インタフェース内での対話処理について次章で考察している．

```
s1( 始める([$object(lesson1),$persuasive]) ).
    lesson1( $modify(地理,勉強) ).

s2( 生育する([$actor(米),$place(北京),$interrogative]) ).

s3( 選ぶ([$object(region1),$from(next),$imperative]) ).
    region1( $modify(生育する([$actor(米)]),地域) ).
    next( $modify(次,中) ).

s4( 生育する([$actor(米),$place($modify(温かい,所))]) ).

s5( $because(soviet1,生育する([$actor(米),$negative]),[]) ).
    soviet1( $is(ソ連,$modify($emphasis(寒い),国)) ).

s6( 生育する([$actor(米),$place(イラン),$cause(なぜ),$interrogative]) ).
```

(a) 概念表現1 (CR1)

```
s2'( 生育する([$actor(米),$place(北京),$interrogative,$p_interrogative]) ).

s4'( 生育する([$actor(米),$place($p_prominent($modify(温かい,所)))]) ).

s5'( $because(soviet1,生育する([$actor(米),$negative]),[]) ).
    soviet1( $is(ソ連,$modify($p_prominent(寒い),国)) ).

s6'( $why(生育する([$actor(米),$place(イラン)]),[$p_interrogative]) ).
```

(b) 概念表現2 (CR2)

図4.4　概念表現例

4.4 SOCSシステム

SOCSシステムは図4.3に示した構成をとり，概念表現2が入力されると文生成および韻律生成を行ない，それを音声に変換する．SOCSには種々のテンプレート・オペレータに関する知識が蓄えられており，慣用テンプレートなど一部は対話管理部と共有されている．このような知識と文生成・韻律生成のメカニズムによってSOCSの基本アーキテクチャが構成される．また，実際に韻律生成を行なうには，ポーズ挿入のしきい値や強調表現におけるパラメータなど，SOCS運用上の知識がさらに必要になる．本論文では，文生成および韻律生成のメカニズムについて主に述べることとし，具体的な慣用テンプレートの構成や運用上の知識の詳細な検討に関しては今後の課題としたい．

4.4.1 文生成

一般に自然言語生成の過程は，waht-to-say決定部とhow-to-say決定部に大きく分けることができ，高度な自然言語生成を行なうためには，この二つの決定部の相互作用の重要性が指摘されている[徳永91]．しかしながら，汎用な音声出力インタフェースでは，音声言語生成においてどのような処理が問題解決器に依存せずに行なえるかを考えると同時に，問題解決器とインタフェース部での処理の切りわけを行なう必要があることから，本インタフェース部ではwhat-to-sayの決定は対象外となる．一般に表層文における語の選択はhow-to-say決定部で行なうべき重要な問題の一つであるが，これは問題解決器の知識ベースに依存した問題であることから，本インタフェースの枠組では概念表現において既に語の選択が行なわれている．さらに，文生成に必要な相・時制・様態などの法情報も問題解決器によって概念表現中に記述されている．従って，インタフェース部においては，適切な語順や構文の決定を行なう必要があるが，これらの処理は対話管理部で行なわれる．また，表層文における語の選択は問題解決器によって行なわれるが，名詞の代名詞化など対話コンテキストを利用して処理されるものは対話管理部で行なわれる[松山92]．SOCSにおける文生成部は，格文法とテンプレート，

オペレータを適用することによって，概念表現2に記述された情報に従って文生成を行なう．

概念表現は動詞の格構造およびテンプレートが入れ子状に組み合わされて構成されている．文生成では，外側の動詞あるいはテンプレートから順に再帰的に処理される．各テンプレートには，出力すべき文／句パタンが予め記述されており，それに従って表層文が出力される．動詞に関しては，引数として記述されている格情報を出力し，さらに指定されている付加情報に従って付属語を付加すると同時に文体が「です」「ます」調になるように語尾を変更する．動詞に対する格情報は予め定められた順序に従って出力される．ここで，テンプレート中に記述されているポーズマーカーは単語と同様に文の中に埋め込まれて出力される．

図4.4(b) の s2' を例にとり，具体的に説明する．s2' は動詞 "生育する" の格構造から成っている．まず，その格情報である "$actor(米)" と "$place(北京)" が予め定義されている格情報の順序に従って処理される．先に "$actor" が格テンプレートの定義に基づいて処理されて "米が p1" が出力され，続いて同様に "北京で p1" が出力される．次に動詞 "生育する" が処理される．これには法オペレータ $interrogative が指定されていることから，「です」「ます」調への変換と合わせて疑問助詞「か」が追加され "生育しますか" が出力される．このようにして s2' からポーズマーカーを含んだ

s2:　米が p1 北京で p1 生育しますか

が出力される．

また概念表現は，テンプレートや動詞の入れ子構造として記述されるため，単語およびポーズマーカーが生成される入れ子の深さが異なるが，後述するポーズ挿入などの処理のためにこの深さも記憶しておく．

図4.4 に示したその他の概念表現からは，上に示した s2 の他，それぞれ，

s1:　地理の p1 勉強を p1 始めましょう

s3:　次の p1 中から p1 米が p1 生育する地域を p1 選びなさい

s4:　米が p1 暖かい所で p1 生育します

s5: ソ連は p1 寒い国ですから p2 米が p1 生育しません

s6: なぜ p2 米が p1 イランで p1 生育するのですか

のような文が生成される．

4.4.2 韻律生成

SOCS システムでは，テンプレートに記述されたポーズマーカー，慣用テンプレートに記述される韻律修正関数，概念表現に示される韻律オペレータの三つのメカニズムによって韻律が生成される．ポーズと話調成分の立直し位置がポーズマーカーによって決定され，継続時間長，話調成分の始端周波数，アクセント成分の大きさが韻律修正関数によって決定される．以下にこれらによる韻律生成の基本方式について述べる．

(1) ポーズの挿入

人が発声する場合に一息で発声できる声の長さには限りがあるため，合成音にも適当にポーズを挿入することが合成音の自然性には不可欠である．SOCS では，PS1，PS2 の二種類のポーズが用いられる．PS2 が長いポーズ，PS1 が短いポーズである．表4.2，4.3，4.5，4.6 のテンプレートで記述されている p1，p2 のポーズマーカーは，それぞれ，

p1: PS1 のポーズが挿入される可能性がある．

p2: PS2 のポーズを挿入する．

を示している．表4.6 の "why(A,OPs)" テンプレートの "なぜ" の後のように常にポーズを挿入した方が良いと思われる場所には，p2 をテンプレートに挿入しておく．

記号 p1 に関しては，一つの呼気段落のモーラ長がしきい値を越えている場合には p1 を PS1 に置き換え呼気段落を二つに分割する処理を行なう．分割された呼気段落もそのモーラ長がしきい値以内になるまで，この処理を再帰的に行なう．概念表現からの文生成では，テンプレートの呼び出しと述語に対する格情報

の付加が入れ子状に行われるため，一文中の単語およびポーズマーカーは異なる入れ子の深さで生成される．p1 の PS1 への置き換えは，より外側のテンプレートによって生成された，言い換えると，より浅い深さで生成された韻律記号から順に行われる．

図4.4 の s3 に対するポーズ挿入の例を図4.5 に示す．概念表現から文が生成された段階 (図4.5(a)) では，全体が一つの呼気段落と仮定される．この例でモーラ長がしきい値を越えているとすると，ポーズマーカー p1 のうちの一つがポーズ PS1 に置き換えられる．文生成によって得られた単語およびポーズマーカーをそれが生成されたテンプレートの深さと合わせて表現すると，図4.5(b) のようになる．ここで，"中から p1 " と "地域を p1 " の p1 が最も浅い位置で生成された (述語「選ぶ」の格として生成された) ポーズマーカーである．このように複数のポーズマーカーが候補になる場合には，呼気段落をなるべく等分するようなポーズマーカーが選択される．その結果，"中から p1 " の p1 を PS1 に置き換え図4.5(c) を得る．モーラ長がしきい値を越える呼気段落がまだ存在する場合には，同様の処理を再帰的に行なう．残された p1 は最終的には削除されるが，次項 (2) で述べるように，ポーズマーカーは話調成分の立直し位置の決定にも用いられるため，残りの p1 はここではまだ削除されない．

一般に発話に現われるポーズには，呼気の制約から息を継ぐためのものと意味的な切れ目を表現するものがあると考えられる．ここで述べたポーズ生成の枠組

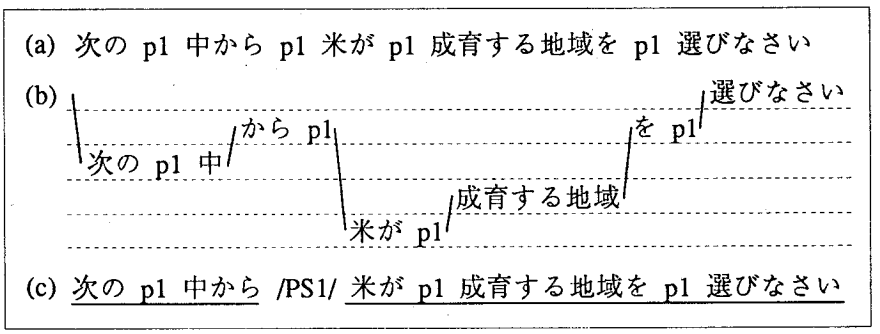


図4.5　ポーズ挿入例

では，p2, p1 がそれぞれ意味的な切れ目によるポーズ，呼気の制約から生じるポーズを生成し，p1 に関しても意味的な切れ目の大きさを考慮している．ここで，p2 は常にポーズに変換されるため，これが多く文中に現われると，ポーズが挿入され過ぎ不自然になることも考えられる．しかし，p2 は主に慣用テンプレートによって生成され，多数の慣用テンプレートが一文中に入れ子状に用いられることはほとんどないことから，p2 によるポーズの過生成は起こらないと考えられる．また，各ポーズの時間長とポーズ挿入のためのモーラ長のしきい値は調整可能であり，実際の運用に即して最適化を行なうことによって自然なポーズの挿入がなされる．

(2) 話調成分の立直し位置の決定

基本周波数の決定は，図4.6 に示す箱田らによって提案されたモデル[箱田80]，すなわち話調成分にアクセント成分を重ね合わせて表現するモデルを用いて行なう．このため，話調成分の立直し位置を決定する必要がある．話調成分の立直し位置は，ポーズ挿入と同様にポーズマーカーを用いて決定する．ただし，ポーズ挿入の場合よりもモーラ長のしきい値を小さくするため，ポーズの挿入される位置は必ず話調成分の立直し位置となる．

(3) 韻律オペレータの機能

4.3.2 で述べたように SOCS では三つの韻律オペレータが定義されている．

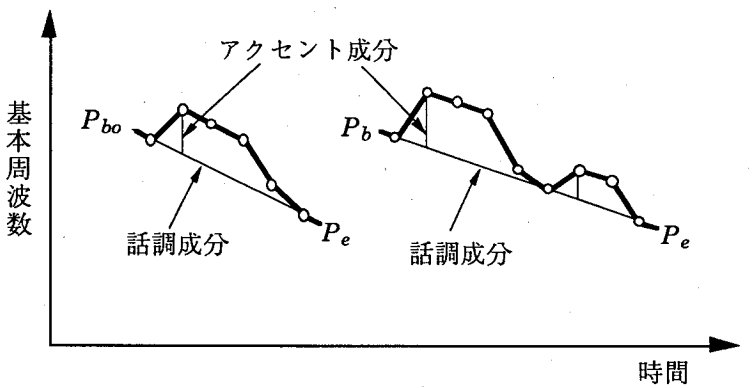


図4.6　基本周波数モデル

$p_prominent(A) は強調する語句を引数にとり，基本周波数および時間長を増加させることによって，韻律的な強調を表現する．$p_interrogative は疑問文において指定され，文末の基本周波数を上昇させる．(ここで，法オペレータの $interogative は，単に疑問助詞「か」を文末に付加するだけの処理を行なう．疑問助詞「か」を伴わない疑問文を生成できるようにオペレータを分離している．) $p_speed(A,S) は第1引数で示される語句に対して時間長を変化させる．変化の程度は % で第2引数として指定される．$p_speed(A,S) は，音響的に類似した語の存在する語や耳慣れない語を聞きやすくするために，文中の一部の語の発話速度を遅くするなどの目的に用いられる．参考までに，運用知識として SOCS のプロトタイプシステムで用いている $p_prominent(A) のパラメータ修正操作を図4.7 に示しておく．実際の運用においてどのようにパラメータを操作し強調を表現するかに関しては，ポーズを挿入するなど種々のパラメータを組み合わせて実現することが検討されている[武田91].

(4) 韻律修正関数の機能

韻律オペレータが概念表現の一部として用いられるのに対して，韻律修正関数は，慣用テンプレートの中でのみ用いられる．表4.7 に示すように三種類の韻律修正関数 $mod_dur，$mod_bpit，$mod_acc が用意されており，それぞれ，時間長，話調成分の開始周波数，アクセント成分の設定／修正を行う．(基本周波数モデルとして文献[箱田80]のモデルを用いており，このモデルにおけるアクセント成分

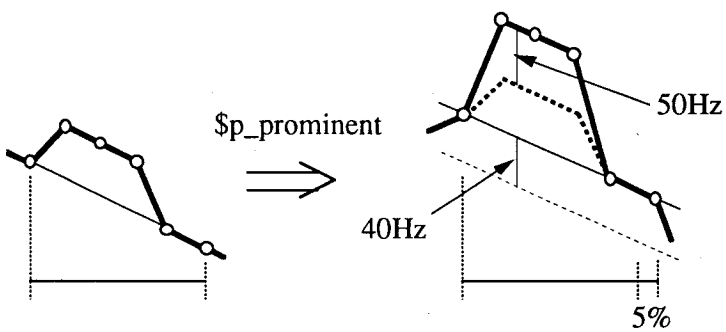


図4.7　"$p_prominent" オペレータによる韻律操作

表4.7　韻律修正関数

継続時間長	$mod_dur(pos,sign,duration)
話調の開始周波数	$mod_bpit(pos,pitch) $mod_bpit(pos,sign.pitch)
アクセント成分	$mod_acc(pos,accentLevel) $mod_acc(pos,sign,accentLevel)

が韻律修正関数によって操作される．）いずれも，3引数の関数は相対的な値の増減を行ない，第2引数は +，あるいは - の符合である．2引数の関数は値の設定を行う．また，第1引数 pos が慣用テンプレートの出力リストの何番目の語句に対して処理を行うかを指定する．時間長を操作する $mod_dur では，相対的な値の増減を行なう関数のみが定義されており，第3引数は時間長増減の程度を % で指定する．例えば，表4.6 の $reason テンプレートに現れる $mod_acc(4,+,15) は，$reason の2番目の引数 B のアクセント成分の大きさを 15Hz 増加させる．

(5) 韻律生成の例

図4.4(b) の s5' から生成された基本周波数とポーズのパタンを図4.8 に示す．この例では，慣用テンプレート $because によって長いポーズ (SOCS のプロトタイプシステムでは長いポーズは 300ms に設定してある) が挿入され，その韻律修正

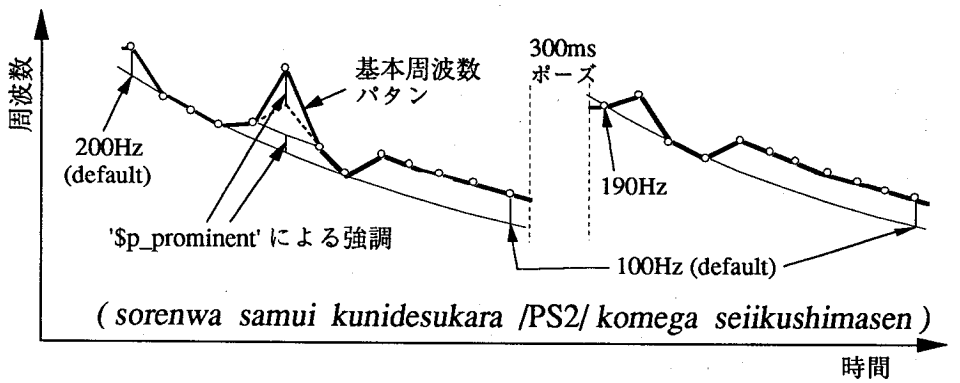


図4.8　基本周波数パタンとポーズの生成例
（図4.4 の s5 に対して）

関数によってポーズ後の開始基本周波数が190Hzに設定される．話調成分終端の基本周波数は常に一定の値(プロトタイプシステムでは100Hz)に設定される．また，概念表現中に「寒い」の強調が記述されているため，そこでは基本周波数が上昇している．

4.5 結言

種々の問題解決器からの高品質な音声出力を容易に実現するには，問題解決器と音声出力インタフェースを統合的に捉え，これら二つのシステムの相互作用，役割分担をシステム論的に考察する必要がある．このような観点から，本章では，概念表現に基づいた汎用音声出力インタフェースの枠組の設計思想と基本アーキテクチャの設計について述べた．これまでにUNIXワークステーション上でC-Prologを用いてSOCSのプロトタイプシステムを実現し，筆者の所属する研究室で開発を行なっている知的CAIシステムと接続し，本枠組の実現可能性を検証した．しかし，各パラメータの設定などの詳細なシステムの検討およびその定量的な評価は，今後の課題として残されている．また，対話を行なう上では，伝達する意味内容が同じでも対話のコンテキストなどによって最適な表現が異なってくる．例えば，ある語句を強調するためには語順を変更するとか，繰り返して発話するなどの文表現も用いられる．このような強調などを含めて，文生成レベルでの最適な表現の生成を行なうには，対話管理部で扱うべき情報を整理し，文生成・音声合成部との相互作用について更に検討していく必要があると思われる．

第5章

対話音声合成における対話管理

5.1 緒言

音声研究の目的の一つとして，計算機上に実現された問題解決器と音声を用いて情報交換を行なう，音声対話システムの構築が挙げられる．このような音声対話システムを実現するためには，音声認識や音声合成の音声信号処理に加えて対話管理が重要な問題となる．音声認識での曖昧性の解消や音声合成のための韻律的特徴の決定において，対話に関する知識の利用が有効であることがこれまでにも指摘されている[Hauptmann88][山岡90][Hirschberg90]．ここで対話管理の役割について考えてみると，音声入力のための対話管理では，対話構造の理解，ユーザ発話の予測，省略語の補充などの処理を行い，音声出力では，対話コンテキストにあった表現の決定，音声合成に必要な情報の抽出などを行う．さらに，入力音声の認識誤りやユーザの合成音に対する聴き誤りなどの問題も音声対話では避けることができず，対話管理部で扱うべき問題となる．これらの問題は問題解決の領域には依存しない対話に固有な問題であることから，いくつかの対話処理は問題解決器と分離して考えることができる．従って，このような対話に関する知識を利用して音声対話を効率良く実現するには，対話管理の機能を取り込んだ汎用のインタフェースシステムを構築し音声処理を行なうべきである．このような観

点から，4章では汎用音声インタフェースの概念設計を行なった．

汎用の音声インタフェースでは，すでに指摘したように，問題解決器とインタフェース部での処理の分担が重要な問題となる．対話音声合成という観点から，4章で述べた概念表現を用いた音声合成において処理の分担がどのように行なわれるかを考えてみる．問題解決器は概念表現を生成することによって「言いたいこと」を決定するが，ユーザに提示される音声言語表現の文構造や韻律的特徴に関しては注意を払わない．一般に，同じ意味内容でも対話コンテキストによって，すなわちそれ以前にどのような発話が行なわれたかによって最適な表現形式が異なってくる．そこでインタフェース部の対話管理部では対話コンテキストを利用して文表現の変更や韻律的特徴の付与を行なう．本章では，このような対話音声合成における対話管理について述べる．5.2ではまず，対話コンテキストを利用するための対話構造モデルについて述べ，5.3ではそのモデルの妥当性を模擬対話を使って検証する．さらに5.4では概念表現と対話構造モデルに基づいた具体的な対話処理を示す．

5.2　対話構造モデル

対話コンテキストを用いた処理を有効に行なうためには，単に対話の履歴を保存しておくだけでなく，対話を理解するための対話構造モデルを持つ必要がある．一般に，ある目的を持って行なわれる目的指向対話[飯田88]では，図5.1に示すように少なくとも二種類の対話構造，すなわち発話の対に基づく構造 (発話対[柏岡89][Yamamoto91]) と意味的なまとまりに基づく構造 (対話セグメント[Grosz86]) が存在する．協調的に行なわれる目的指向対話では，相手の発話を無視することなく話を合わせて対話が進められる．このため「質問すれば答える」といった発話の「やり取り」から対話が構成され，時にはそれが入れ子構造となる．また，このような対話では，対話の内容が関連性のない話題に突然変わることはない．一つの対話セグメントは発話の意味的なまとまりであり，その中では同じ話題のもとで対話が進む．意味的まとまりの大きさ (抽象度) の見方を変えることで

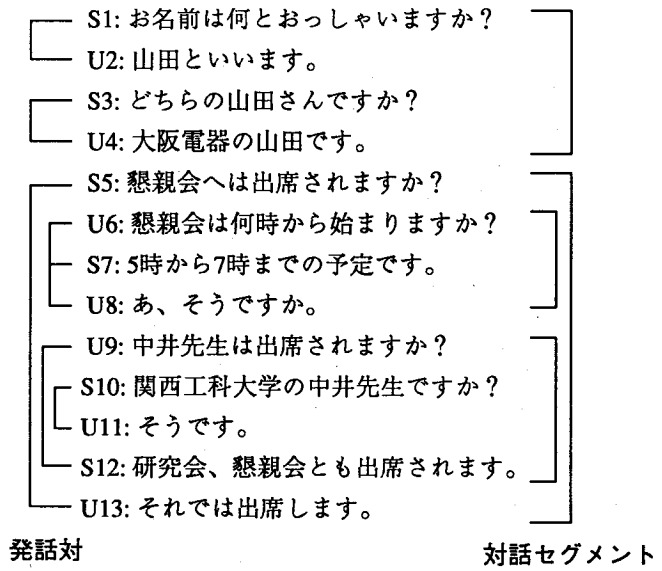


図5.1　二種類の対話構造

対話セグメントも一般に入れ子構造を示し，話題における階層性と対応する．

このような観点から，対話管理部では，発話対および対話セグメントに対してモデルを持たせている．すなわち，発話対による対話構造は山本らによって提案されているSR-プランによって表現され[Yamamoto91]，対話セグメントは筆者が提案するTPN (Topic Packet Network) によって表現される．このようなモデルと概念表現を利用することによって対話音声合成のための対話処理が行なわれる．本節では，まずSP-プランについて簡単にふれた後，TPN (Topic Packet Network) について詳しく述べる．

5.2.1 SR-プラン

問題解決器とその利用者の間で行なわれるような目的指向の音声対話では，一回の発話の中にそのときの目的を達成するために相手に働きかける発話(要求)，および相手の働きかけに対して返答する発話(応答)のどちらか一方，あるいは両方が明示的に，または暗黙に必ず含まれていると考えられる．そこでの対話は，

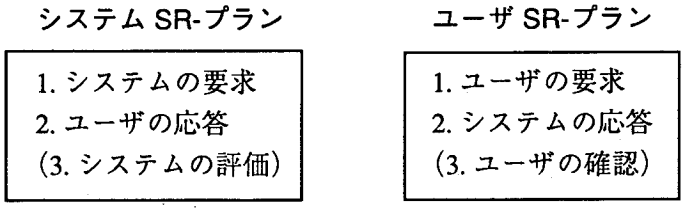


図5.2　SR-プランにおけるやり取りの形式

お互いが要求と応答のやり取りを行なうことによって進められる．そこで，このやり取りの系列を一つのプラン[Schank77]として捉え，このプランが要求(Stimulus) と応答 (Response) から成っていることから，これを SR-プランと呼ぶ[Yamamoto91].

SR-プランは，要求の発話をシステムが行なうシステムSR-プランと，ユーザが行なうユーザSR-プランに大きく分類できる．(ここでは，問題解決器を「システム」，その利用者を「ユーザ」と呼ぶことにする．) 図5.2 に SR-プランにおけるやり取りの形式を示す．システムSR-プラン，ユーザSR-プランでは，要求・応答のやり取りの後にそれぞれシステムの評価，ユーザの確認の発話が行なわれることがあるが，やり取りによってはこれらは省略されることもある．さらに，要求・応答のやり取りを分類整理することによって，SR-プランは図5.3 に示すように 17 個に分類されている．

5.2.2 TPN

(1) TP による話題の局所的遷移の記述

一般に，目的指向対話における話題は一種の階層構造をなすと考えられる．言い換えると，対話が開始された時点での話題はおおまかな話題であり，対話が進むにつれて必要に応じてそれがより詳細な話題へと遷移し，さらに詳細化される以前の話題へと話題がしばしば復帰する．ここで，一つの話題はある限られた話題の集合へと詳細化でき，この集合の中の話題は相互に非常に関連性の高いもの

S-QUIZ	<問題>
1. SP-ASK-COMPONENT	動作の対象物、動作を行う位置などを尋ねる
2. SP-ASK-REASON	動作の理由を尋ねる
3. SP-ASK-FACT	ある事実が正しいかどうかを尋ねる
S-COMMAND	<命令>
4. SP-COMMAND	何か仕事を要求する
S-EXPLAIN	<説明+要求>
5. SP-AFTER-EXPLAIN	説明してから理解したかどうか確認する
S-DEMAND-U.SPEECH	<ユーザの発話に関する要求>
6. SP-CONFIRM	システムの理解が正しいかどうか確認する
7. SP-SUPPLEMENT	ユーザの発話の中で不足している情報を尋ねる
8. SP-DECIDE	ユーザの発話の中で曖昧な情報について尋ねる

(a) システム SR-プラン

U-DEMAND-K.BASE	<システムの知識ベース関する要求>
9. UP-ASK-DEFINITION	言葉の意味(定義)を尋ねる
10.UP-ASK-COMPONENT	動作の対象物、動作を行う位置などを尋ねる
11.UP-ASK-REASON	動作の理由を尋ねる
12.UP-ASK-WAY	動作のやり方を尋ねる
13.UP-ASK-FACT	ある事実が正しいかどうかを尋ねる
U-COMMAND	<命令>
14.UP-COMMAND	何か仕事を要求する
U-DEMAND-S.SPEECH	<システムの発話に関する要求>
15.UP-ASK-UTTERANCE	システムが何と言ったのか聞き返す
16.UP-CONFIRM-UTTERANCE	システムが言ったことを確認する
17.UP-ASK-MEAN-SENTENCE	システムの言った内容の意味を尋ねる

(b) ユーザ SR-プラン

図5.3　SR-プランの分類

であると考えられる．また，「親」の話題と詳細化されて現われる「子」の話題集合の関係は，個々の対話に依らない．そこで，このような互いに関連の深い話題の集合を TP (Topic Packet) と呼ぶ．

図5.4 にこの TP の例を示す．例えば，車の設計に関する対話を行なっている時には，"吸気" が話題になると，それが詳細化することによって "キャブレタ"，"エアクリーナ"，"過吸器"，"バルブシステム" などが話題になることがある．図5.4(a) はこのような話題の関係について記述したものである．この話題遷移は車の設計だけでなく，車の診断においても同じように当てはまる．同様に，図5.4(b) の TP は観光案内においても地理案内においても利用可能な話題遷移情報である．このように，TP は対話における局所的な話題の遷移を表現したものであることから，個々の対話に依存することなく記述できる．

(2) TPN による話題の大局的遷移の記述

TP は図5.4 にも示したようにそれぞれ上位話題 (super topic) を持っている．対話が進むにつれて話題が詳細化されると，もとの話題を上位話題として持つような TP が活性化され，その中の一つの話題が新しい話題となる．話題の詳細化が繰り返されると次々に TP が活性化していき，話題が終了すると上位の TP 中へと話題が遷移する．このように，対話における話題の遷移は，いくつかの TP を

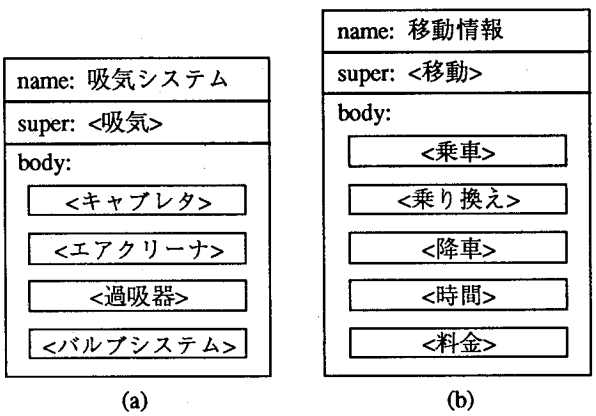


図5.4　TP の例

上位話題を介して接続されるTPのネットワーク上で表現される．このTPのネットワークをTPN (Topic Packet Network) と呼ぶ．

一般に，対話における話題は，話題が継続する場合と遷移する場合に分けられるが，話題が遷移する場合は，さらに，

1. 話題の詳細化
2. 関連した話題への遷移
3. 話題の終了

の3種類に分類できると考えられる．TPNのモデルでは，図5.5に示すように，3種類の遷移によって活性化される話題は，それぞれ，一つ下位のTP中への話題，現在の話題を含むTP中のその他の話題，一つ上位のTPの話題としてモデル化される．

このように話題の遷移をネットワークによって表現する場合，全体が話題から直接構成される大きいネットワークになっていると，問題解決器ごとにそれを適応させることは非常に困難になる．そこで，TPNではTPを問題解決器と独立し

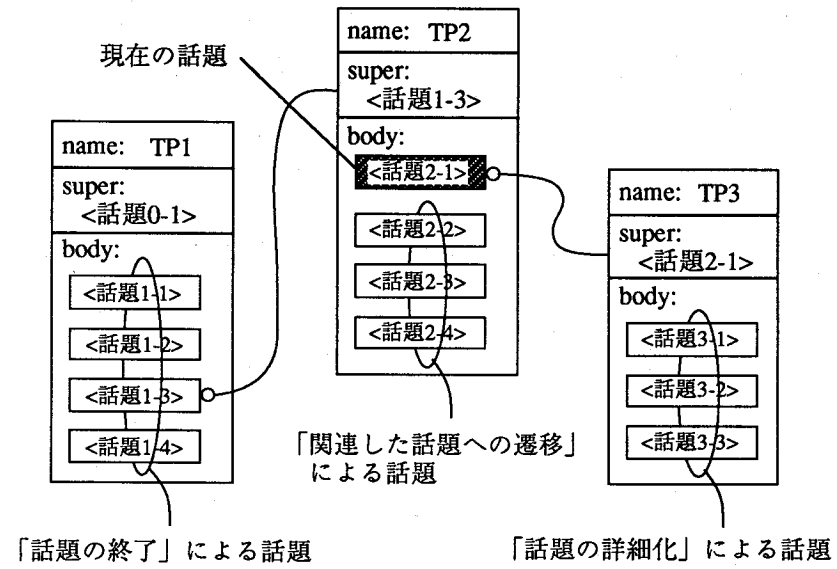


図5.5　三種類の話題の遷移

た部品として用意し，それを接続することによってネットワークを構成することで，再利用性を高めている．さらに，TPを単位として話題の遷移範囲を限定するモデルを与えることで，対話音声理解におけるユーザの次発話予測にも有効に用いることができる[平松92]．

(3) 従来の研究との比較

これまでに自然言語理解の研究において，人間の行動を記述した文章を理解することを目的として，Schank らによって MOP (Memory Organization Packet) [Schank82] が提案されている．MOPはいくつかの抽象化された場面の系列とゴールから成っており，多くの実際の「できごと」に適応できる．TPはMOPにおいて場面を話題に置き換えたものに似ているが，MOPが人間の行動を説明するのに対して，TPは話題を列挙しただけで人間がとる行動とは直接結び付かないものも含んでいる．また，Kellermann らは初対面の形式張らない会話という限られた状況において，どのような話題 (verbal action と呼んでいる) が会話に現われるかを調べている[Kellermann89]．そこでは，MOP の枠組で話題の遷移を説明しており，これを会話 MOP (conversation MOP) と呼んでいる．

また，対話のモデルとしてはこれまでにもプランを用いた手法が多く提案されている[Litman87][飯田90]．これらの方法では対話構造の特徴を全てプランで捉えようとしているため，プランの正確な記述が必要であり，複雑な問題解決においては記述すべきプランの量もかなり多くなる．反面，対話音声理解，発話予測といった観点からは，プランを強力な制約として利用できる．本論文で述べる対話管理部では，SR-プランとTPNを用いて対話構造のモデル化を行なっている．プランは局所的な発話対のモデルとしてのみ用いられ，大局的な対話構造の特徴はTPN によって与えられる．TPはその中での話題の出現順序などに関しては規定しないことなどから，TPNはプランに比べると大まかな構造を記述していると言える．この点で，本モデルはプランのみによるモデルと比べると，制約を与える力は劣るが，柔軟性に富んでいると言える．

5.3 模擬対話における話題の遷移

話題に関する対話構造のモデルとして前節で述べたTPNがどの程度妥当なものであるかを検証するために，模擬対話における話題遷移について調べた．

5.3.1 模擬対話の収録

音声対話処理の研究では，対話構造の解析やモデル化，発話に現われる表現の分類，対話音声における韻律的特徴の分析などを行なうために，対話音声データの収集が重要な研究課題になる．このような認識から，対話音声の収録がいくつかの研究機関で行なわれている[田中92][菊池92][速水91]．本研究においても，対話における話題の遷移を調べTPNの妥当性を検証するために，模擬対話の収録を行なった．ここで，模擬対話収録の目的には，話題に関する分析の他，発話における表層表現の収集や，対話音声の韻律的特徴の分析なども含まれている．

模擬対話の収録は以下のような要領で行なった．

非対面の対話

一般に対話システムが，音声のみを媒体として構築されるべきであるとは限らない．しかしながら，音声による対話の特徴に注目するには，他の媒体による情報伝達がないような状況の方が都合が良いと考えられる．そこで，対話は非対面で行なった．具体的には，二人の実験協力者 (対話者) が別の部屋におり，内線電話を用いて対話を行なった．ただし，録音に関しては，それぞれの対話者の前にマイクを設置し，一つのレコーダの左右のチャンネルに録音した．従って，対話者の音声は完全に二つのチャンネルに分離されるため，割り込み発話などでも音声の重なりがなく，韻律的特徴の分析においては都合が良い．

システム役とユーザ役への指示

菊池らの報告[菊池92]でも指摘されているように，ある目的を達成するために行なう対話を模擬するには，システム役 (回答者) が回答するために必

要な情報を予め整理しておくことが重要である．模擬対話は「地理案内」をタスクとして行ない，案内すべき地理に関する情報として，駅名，所要時間，料金などを紙にまとめ，システム役に熟知させた．ここで，ユーザ役にとって案内される地理が一部既知であると対話に現われる話題が制限されることが考えられるため，架空の地名も用いた．(しかし，架空の地名は対話の現実感を欠くことになり，その使用の是非は検討課題として残される．)ユーザ役には，「～の状況にある．～へ行けるように地理案内システムに教えてもらう」というようなゴールを与え，電話を (システム役へ) かけてもらい，模擬対話が開始される．システム役は，ユーザ役のゴールを知っており，予め用意してある情報に従って回答していく．

また，システム役の回答方針 (どの程度の内容を一度に回答するか) によって対話の進行が大きく影響される．そのため，システム役にはある程度回答方針に関して指示を与えた．例えば，「必要最低限のことだけを答える」などである．

5.3.2 模擬対話の分析

上述した要領で 10 個の模擬対話を収録した．このうちの 4 対話と日本音響学会のデータベースから 2 対話[小林92]，合計 6 個の模擬対話を用いて話題に関する分析を行い TPN を抽出した．図5.6 に結果を示す．対話においてある意味的なまとまりの中の話題が，一つのパケットに含まれるように TP を手作業で抽出し，12個の TP を得た．"時間TP"，"料金TP" などには一つの話題しか含まれていないが，他の領域の対話ではこのような TP が (地理案内では現われなかった他の話題も含んで) 存在すると考え，ここでは独立した TP とした．

TPN における話題遷移のモデルでは，話題の遷移範囲が同じ TP，一つ上位の TP，一つ下位の TP と限定され，話題の遷移は，

・話題の継続

・話題の詳細化

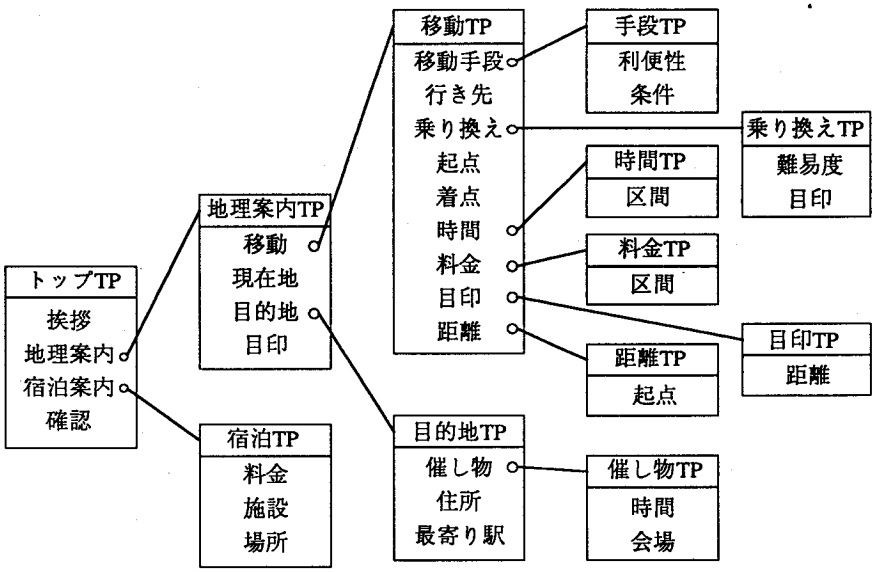


図5.6　対話例から抽出された TPN

表5.1　対話例における話題の遷移

対話例	osa0008d	osa0009d	osa0010d	osa0013d	osa0003r	etl0008r	合計
(a) 発話数	43	75	152	168	30	77	545
(b) 話題が決定できない発話数	16	30	27	61	4	7	145
(c) 話題が決定できる発話数 (b-a)	27	45	125	107	26	70	400
(d) 話題の継続	12	25	71	56	14	27	205
(e) 話題の詳細化	3	3	11	12	2	11	42
(f) 次の話題への遷移	11	10	29	22	5	17	94
(g) 話題の終了(上位話題)	0	1	8	8	0	8	25
(h) 話題の終了(上位の別の話題)	0	1	2	3	1	2	9
(i) e+f+g+h	26	40	121	101	22	65	375
(j) その他の遷移	1	5	4	6	4	5	25

・関連した話題への遷移

・話題の終了

のいずれかに，TPに基づいて発話ごとに分類される．表5.1は，それぞれの対話において，話題の遷移が図5.6のTPNでどのように捉えられるかを示したものである．対話において，「はい」(合槌も含む)や「もう一度お願いします」のような発話は話題の継続を意味しており，発話単独では話題を決定できない．このような発話が(b)である．話題の終了に関しては，上位の話題への遷移(g)と上位の話題と同じTPの話題への遷移(h)に分けて示した．このように，TPNでのモデルの範囲外へ話題が遷移したもの(例えば二つ上位のTPへの話題の復帰など)は非常にわずかで(25/545=4.5%)，TPNによって話題の遷移がうまくモデル化できることがわかった．

5.4　概念表現に基づいた対話処理

図4.3で示した概念表現を用いた対話音声合成の枠組では，問題解決器が生成した概念表現1を対話管理部が対話コンテキストを利用して概念表現2に修正し，概念表現からの音声合成システムSOCSがそれを合成音に変換する．本節では，概念表現と対話構造モデルに基づいて行なう対話処理として，四つの具体的な処理を述べる．

5.4.1　強調語句の抽出

対話では発話の中である語句が，文表現としてあるいは韻律的特徴として強調されることがある．同じ意味内容が対話の中でなく単独で発話された時にはその語句が強調されないような場合では，その強調は問題解決の結果としてではなく，むしろ対話によって引き起こされたと考えるべきである．従って，本論文で述べる汎用音声出力インタフェースの枠組では，このような対話コンテキストによって，すなわちそれ以前にどのような発話がやり取りされたかによって決定さ

れる語句の強調は，問題解決器ではなくインタフェース内部の対話管理部で処理すべきである．そこでまず，対話コンテキストによって決定される語句の強調を分類し，具体的な強調語句抽出メカニズムを示す．

(1) 語句の置き換え

例えば，ユーザの発話「A の B に C する」の後でのシステム (すなわち，問題解決器) の発話「A' の B に C する」のように，よく似た表現の一部が別の語句に置き換えられることがある．このような場合には，置換えた語句 A' をはっきりと提示するためにそれを強調すべきである．図5.7(a) はこのような対話の例で，"代々木" が強調すべき語句となる．ここで，U, u, s, s', S は，それぞれユーザの発話，ユーザの発話の理解結果，システムが生成した概念表現1，対話管理部が修正した概念表現2，ユーザに提示されるシステムの発話を表している．また，u で示されているように，ユーザの発話は音声入力処理によって同じ形式の概念表現に変換されることを仮定しており．対話管理部は，このように概念表現のレベルで直前のユーザの発話(u)とシステムの発話(s)を比較し，良く似た概念表現パタンの中で一つだけ置き換えられた単語を強調すべき語として抽出し，$p_prominent オペレータを付加して(s') SOCS に渡す．

(2) 類似語句の選択

並列の関係にある語句は一般に意味的に類似した概念を表すと考えられ，概念表現では図5.7(b) の u のように慣用テンプレート $and あるいは $or を用いて表現される．このような意味的に類似した概念が複数提示された後で，その一つについてのみ言及する発話ではその語句を強調すべきであると考える．そこで，(b) の対話例のように，ユーザ発話に $and あるいは $or が含まれており，その後のシステム発話においてそのテンプレートの構成単語の内の一つが現われた場合には，それを強調すべき語句として抽出する．図5.7(b) の例では，"京都" が強調される．

U: 新宿は渋谷の次の駅ですか？
u: $is(新宿, $modify($modify(渋谷,次),駅), [$interrogative])
s: $is(新宿, $modify($modify(代々木,次),駅), [])
s': $is(新宿, $modify($modify($p_prominent(代々木),次),駅),[])
S: 新宿は**代々木**の次の駅です

(a) 語句の置き換え

U: 大阪と京都のどちらが東京に近いですか？
u: $is($or(大阪,京都), 近い, [$subject(東京),$interrogative])
s: $is(京都, 近い, [$subject(東京)])
s': $is($p_prominent(京都), 近い, [$subject(東京)])
S: **京都**が東京に近いです

(b) 類似語句の選択

U: 交差点からどちらへ進むのですか？
u: 進む([$direction(どちら),$from(交差点),$interrogative])
s: 進む([$direction(西),$imperative])
s': 進む([$direction($p_prominent(西)),$imperative])
S: **西**へ進んでください

(c) 回答の中心

図5.7　強調すべき語句の抽出

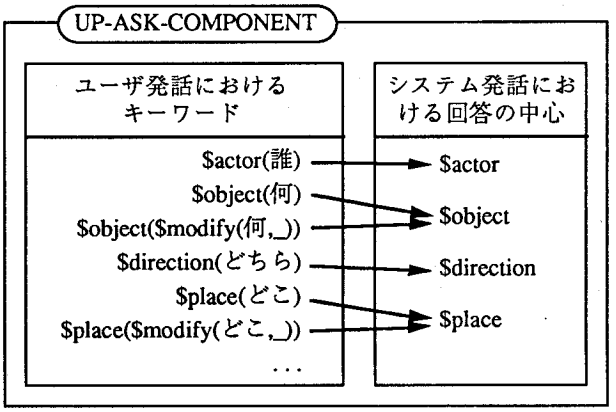


図5.8　システム回答の中心を同定するためのキーワード

(3) ユーザの質問に対する回答の中心

ユーザの質問に対するシステムの発話において回答の中心となる語句を強調する．図5.7(c) においてユーザの発話は方向を尋ねており，次のシステムの発話では方向である "西" が強調される．上述した二種類の強調語句の抽出は，概念表現レベルでユーザ発話とシステム発話を比較することによって実現できるが，ここでは回答の中心を決定するためにユーザが何を尋ねているのか，またシステム発話がどのユーザ発話に対する回答なのかを対話管理部において決定しなければならない．本対話管理部では，SR-プランを用いてこの処理を行なう．図5.8 に示すようにユーザSR-プランの一つである UP-ASK-COMPONENT に質問に現われるキーワードを登録しておくことによって，システム発話における回答の中心がどこにあるかを同定する．例えば，図5.7(c) の例のようにユーザ発話の概念表現に "$direction(どちら)" が記述されていると，次のシステム発話において $direction の格情報が回答の中心になると判断する．先に示した図4.4 (a)の s4 もそれに先行するユーザの発話が「どんな所で米が生育しますか」というような質問であった場合には，(b) s4' のように変換される．

以上のように概念表現を利用することによって，対話コンテキストに依存した

U: 新宿からどのように行けばいいのですか？
u: 行く([$from(新宿),$means(どのように),$interrogative])
s: 乗る([$obj($modify(地下鉄,$emphasis(丸ノ内線))),$imperative])
s': 乗る([$obj($modify(地下鉄,$p_prominent(丸ノ内線))),$imperative])
S: 地下鉄の**丸ノ内線**に乗ってください。

図5.9　ドメイン知識による語句の強調

強調の抽出が容易に行なえる．しかし，語句の強調はこのように対話コンテキストによって引き起こされるものだけではなく，図5.9に示す例のように，問題解決上の知識から語句が強調されることもある．　この例でのシステム発話における"丸ノ内線"は，「新宿を通る地下鉄が複数ある」という問題解決器固有の知識によって強調した方が良いと判断される．このようなドメイン知識は汎用インタフェースでは利用できないため，問題解決器が概念表現1に"丸ノ内線"を強調するための記述を与えておく必要がある．4.3.2で述べた$emphasisはこのために用いられる．

5.4.2 入力音声の認識結果の提示

計算機上に実現されたシステムとユーザとの情報交換で最も重要な点は，一般に通信の問題でもそうであるように，誤りのない情報交換が行なわれることである．また，現在の音声認識技術では発話を全て正しく認識することは困難で，将来においてもある程度の認識誤りは避けることができないと思われる．例えば，図5.10のような対話を考えてみる．この対話において，仮にユーザ発話中の"病院"が"美容院"に誤って認識されていたとすると，システムは"美容院"の場所

ユーザ：病院はどこですか？
システム：この先の交差点の右側です．

図5.10　対話例

を案内していることになり，しかもユーザはそのことに気がつかず，誤った情報を得ることになってしまう．従って，システムへの入出力が音声で行なわれる音声対話システムにおいて誤りのない情報交換を行なうためには，ユーザ発話の音声認識結果を何らかの形でユーザに提示する必要がある．

音声認識結果を確認するには，対話管理部が語句を聞き返す直接的な提示とシステム発話に語句を埋め込む間接的な提示のいずれかを行ない，音声認識結果における1位の候補単語の認識スコアと1位と2位の候補単語の認識スコアの差に基づいて，

1) 1位候補の認識スコアが低い場合
   直接節的な提示を行なう．
2) 1位候補のスコアは高いが，2位候補のスコアとの差が小さい場合
   間接的な提示を行なう．
3) 1位候補のスコアが高く，2位候補のスコアとの差も大きい場合
   特に，認識結果の掲示を行なわない．

のように使い分ける．ここで，認識スコアの高さやスコアの差の大きさはしきい値を設定して判断する．以下，具体的な提示方法について述べる．

まず直接的な提示では，認識スコアの低い1位候補の単語を直接聞き返す．図5.10の対話例で"病院"の認識スコアが低い場合には，このユーザ発話の直後に，

システム：病院ですか？

という発話を行ないユーザの発話内容を確認した後，結果をシステムに送る．この直接的な提示はシステム発話としてユーザに提示されるが，実際にこれを生成するのは対話管理部である．

次に間接的な提示では，2位の候補との認識スコアが近い1位の候補単語をシステム発話に埋め込むことを行なう．例えば図5.10の対話例において，"病院"の認識スコアが2位候補の"美容院"と近かった場合，

システム：病院はこの先の交差点の右にあります．

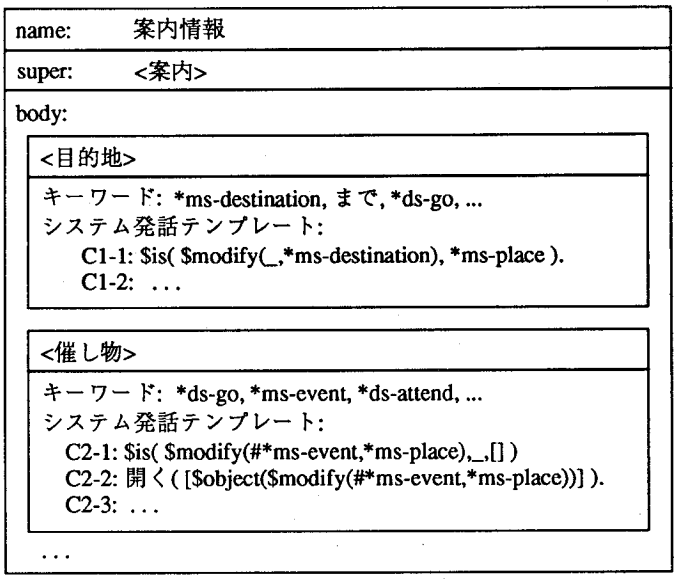


図5.11　TPにおけるシステム発話テンプレート

のように音声認識の結果がユーザに示される．音声出力時に対話管理部へ入力される概念表現に，直前のユーザ発話の情報がどの程度記述されているかはその問題解決器に依存するが，汎用な対話管理部ではその記述量に依らずその対話コンテキストに最も適した表現を決定する必要がある．従って，不必要な情報は削除し，必要な情報が欠落している場合にはそれを補うことが必要になる．このような観点から間接的な提示を行なうために，システム発話に語句を補充するメカニズムについて述べる．

語句の補充は，システム発話がユーザ発話に対する応答か，新たな要求かによって異なる手法で行なわれる．システム発話が応答である場合は，ユーザの要求とシステムの応答を概念表現レベルで比較することによって，欠落している語句を見つけだし，それをシステム発話に容易に補うことができる．ここで要求－応答の発話対の同定・管理はSR-プランによって行なわれる．また，システム発話が新たな要求である場合には，TPを用いて語句の補充を行なう．TPには，システム発話における話題を決定するために，図5.11に示すようにシステム発話テンプレートが話題ごとに記述されている．このテンプレートに語句の欠落のパタ

U: 音声研究会へ行きたいのですが
u: 行く([$obj(音声研究会), $wish]).
s: $is(会場, どこ, [$interrogative]).
s': $is($modify($focus(音声研究会),会場), どこ, [$interrogative]).
S: 音声研究会の会場はどこですか

図5.12　語句の補充例

ンを埋め込んでおく．この図のC1-1，C1-2などがシステム発話テンプレートで，"#" のついた意味カテゴリは欠落する可能性があることを表している．図5.12の対話例では，システム発話 s が C2-1 とマッチし，対話管理部は *ms-event が欠落していると判断する．その具体的な語句が何であるかは，ユーザ発話 u の中で意味カテゴリ *ms-event とマッチする語を探すことによって，"音声研究会" であることがわかる．ここで，ユーザ発話の認識において "音声研究会" のスコアが2位の候補のスコアと近い場合には，この図の s' に示すように "音声研究会" を補充し，間接的に認識結果を提示する．

5.4.3 慣用テンプレートへの書き換え

SOCSではよく用いられる表現に対して4.3.2で述べたように慣用テンプレートを用いることができる．慣用テンプレートには文・韻律のパタンが記述されているため，これを用いた方がうまく文生成や韻律的特徴の決定が行なえる．ここで，問題解決器は慣用テンプレートとしてどのようなものが用意されているかを必ずしも知っている必要はない．問題解決器によって生成された概念表現がSOCSで定義されている慣用テンプレートに変換可能であるときには，対話管理部で表現を変換する．例えば，

開く([$object(音声研究会),$place(どこ),$passive,$interrogative]).

は，

$where(開く ([$object(音声研究会),$passive])).

に変換する．図4.4 (a), (b) における s6 から s6' への書き換えもこの例である．

5.4.4 韻律オペレータの追加

問題解決器は音声固有の韻律的特徴に関する処理を行なう必要がないようにするために，4.3.2 で述べたように概念表現1 には韻律オペレータは記述されない．必要に応じて対話管理部が韻律オペレータを概念表現2 に追加する．

質問のシステム発話では，概念表現1 に法オペレータ $interrogative が記述されており，これは 4.3.2 でも述べたように，SOCS の中では疑問助詞「か」の生成を行なう．対話管理部はこの場合に $p_intterogative を追加し，韻律的特徴を SOCS に伝える．また，問題解決の結果強調すべき語句がある場合には，5.4.1 で述べたように $emphasis が用いられている．　この場合には対話管理部が $emphasis を $p_prominent に置き変える．図4.4 における s2' と s6' の $p_interrogative，s5' の $p_prominent もこの処理によって追加される．

5.5 結言

概念表現に基づいた対話音声合成の枠組では，問題解決器は表層的な文表現や韻律的特徴は決定しない．問題解決器と独立した汎用音声出力インタフェースにおける対話管理部が，対話コンテキストを用いて必要な情報の抽出や概念表現の書き換えを行なう．本章では，このような汎用な枠組みに基づく対話管理の問題について述べた．

対話コンテキストを利用するには，対話管理部が何らかの対話のモデルを持つ必要がある．本システムでは SR-プランと TPN によって対話の構造をモデル化している．TPN は話題の遷移に関する情報を一種のネットワークとして与えたもので，模擬対話からボトムアップ的に抽出した TPN を用いて，その妥当性を検証した．TPN の構成要素としての TP は問題解決の領域独立性が考慮されているもの

の，TPN全体としてはある領域の対話に固有のものとなる．また，幅広い領域をカバーするのに必要な全てのTPを予め用意しておくことも容易なことではない．もちろん模擬対話などの対話例が与えられればそこからTPNを抽出することは可能であるが，TPNを適用する問題解決ごとに模擬対話を行なうことは現実的ではない．従って，TPNの構成法に関しては，問題解決や授受される情報，あるいは問題解決上に現われる行為などの観点から，話題を発生させる要因をトップダウン的に整理していく必要があると思われる．

対話管理部で行なう対話処理として四つの具体的な処理について述べ，概念表現を用いることで対話処理が容易に行なえることを示した．本論文で述べた枠組では，対話管理部と概念表現からの音声合成部は独立したモジュールとなっており，対話管理部での処理結果は概念表現の変更やオペレータの追加などとして音声合成部へ渡される．さらに高品質の文生成・韻律生成を行なうためには，対話管理部と音声合成部の間でより緊密な情報交換や融合した処理が必要になると思われる．また，本章で述べた対話処理は，対話管理部で行なうべき処理の一部にすぎない．今後，対話における音声言語表現の対話コンテキストによる影響を調べることも大きな課題になるであろう．

第6章

結論

本章では，本研究において得られた成果を総括し，今後に残された問題について検討する．

第一に，音声規則合成において合成規則の作成・修正を計算機を用いて支援する手法について検討し，合成ルールベース構築支援 SED-SSRB (a Support Environment for Development of the Speech Synthesis Rule Base) を開発した．音声規則合成システムは，合成規則の集まりを合成ルールベースとして捉えることにより，一種の知識ベースシステムとみなすことができる．(1) 処理の自動化，(2) 情報管理による処理作業サイクルの短縮化の二点において合成ルールベースの構築支援を行なうという立場から，SED-SSRB は，合成規則の自動抽出，合成規則の修正案の提示，規則インタプリタに基づいた音声合成システム SSRI (Speech Synthesis system based on Rule Interpreter) を提供する．

まず，帰納的学習に基づいて多数の音声データから合成規則を自動抽出する手法を示した．それぞれの合成パラメータについて，規則によって決定された値と人間が発声した音声データの分析値との差の記述を例題として用い，音声学に関する専門知識に基づいて条件部を一般化・特殊化することによって規則の自動抽出を行なった．CV 音節継続時間長に関する規則を対象として合成規則の自動抽出を行ない，非学習データにも有効で，専門家によっても解釈可能な規則が抽出された．次に，合成規則の修正に関して，規則の条件部の一般化・特殊化を行な

い，副作用をデータベース中の音声データを用いて定量的に評価することにより有効な合成規則修正案の提示を行なえることを示した．自動抽出によって得られた 20 個の CV 音節時間長規則の修正では，支援環境を用いることによって，一人の専門家が約 75 分間に 21 回のルールベース修正操作を実行し，二乗平均誤差を 80ms から 36ms に減少させることができた．さらに，SSRI では，音声合成時に発火した規則を入力ごとに管理することによって，ルールベース修正後に同じ入力に対して再合成を行なう場合の処理時間を大幅に短縮できることを示した．再合成時の規則適用に要する時間は，新しい入力に対する場合に比べて 1/7 程度に減少する．SSRI を用いた合成ルールベース修正実験では，約 15 時間で 80 単語に対する単語明瞭度が 78% から 93% に，オピニオン平均値が 2.3 から 3.0 に改善され，短時間で効果的な合成ルールベースの修正を行なうことができた．合成ルールベースの改善では，合成規則の修正・評価・修正のサイクルが繰り返されるが，SED-SSRB では評価に関しての支援機能は提供されておらず，専門家の聴取に依らざるをえない．合成音の聴取では，繰り返し聴取すると品質の評価が変わってくる (しだいに良い合成音に聞こえてくるなど) "慣れ" の問題もあり，合成音の評価は音声合成の研究における一つの "ボトルネック" になっていると言え，評価に対する計算機の支援は今後の大きな課題である．

第二に，計算機上に実現された種々の問題解決器から自然で高品質な対話音声出力を実現するために，汎用音声出力インタフェースの枠組を提案し，インタフェースへの入力である概念表現を音声に変換する概念表現からの音声合成システム SOCS (Speech Output from Case Structure representation) の基本アーキテクチャにを示した．従来の TTS (Text-to-speech) に基づいた対話音声出力の問題点を指摘し，CTS (Concept-to-speech) に基づいた汎用音声インタフェースの設計を行なった．インタフェースは，(1) インタフェースの汎用性，(2) 適切な中間表現の設定，(3) インタフェース内部での緊密な処理に留意して設計されている．

まず，インタフェースへの入力として最適な表現形式を問題解決器・音声合成システム・対話管理部の観点から検討し，格構造に基づいた表現と文のパタンを組合わせた表現を概念表現として定義した．概念表現は，問題解決器に依存しな

い構成要素からなっており，具体的には，名詞概念などを表すアトムと，格情報などの表す4種類のテンプレート，法情報などを表す3種類のオペレータから構成される．対話でよく用いられる表現に対して慣用テンプレートを用いることができる点が特徴である．次に，概念表現からの韻律生成の手法を示した．CTSに基づいた音声合成では韻律的特徴を容易に設定できることが利点の一つであり，SOCSにおいても(1)概念表現中に記述される韻律オペレータによる韻律の操作，(2)文生成時にテンプレートから出力されるポーズマーカーによるポーズと話調の立直し位置の決定，(3)慣用テンプレート中の韻律修正関数の三つの手段によって概念表現を用いた韻律操作が行なわれる．さらに，UNIXワークステーション上でC-Prologを用いてSOCSのプロトタイプシステムを実現し，筆者の所属する研究室で開発を行なっている知的CAIシステムと接続し，本枠組の実現可能性を検証した．しかし，各パラメータの設定などの詳細なシステムの検討およびその定量的な評価は，今後の課題として残されている．

第三に，汎用音声出力インタフェースにおける対話管理のための対話構造モデルを提案し，概念表現と対話構造モデルに基づいた対話処理手法を示した．SOCSに基づいた対話音声合成の枠組では，問題解決器は出力の意味内容だけを決定し，音声出力インタフェースにおける対話管理部が対話コンテキストを考慮して最適な音声言語表現を決定する．このように対話コンテキストを用いた処理を有効に行なうためには，単に対話の履歴を保存しておくだけでなく対話を理解するための何らかの知識を持つ必要がある．目的を持って行なわれる目的指向対話では，一般に発話対と対話セグメントの二種類の構造が存在することに注目し，それぞれをSR-プランとTPN (Topic Packet Network)によってモデル化を行なった．SR-プランは応答と要求のやり取りを対話の領域に依存しないように抽象化し整理したもので，TPNは基本的に個々の対話に関わらず存在すると考えられる局所的な話題の遷移パタン(TP)から構成される．このように特定の対話や問題解決に依存しないモデルで対話構造を表現することによって，対話管理部は種々の問題解決器に容易に適用可能な柔軟なシステムとなる．また，TPNに関しては，収録された模擬対話における話題の遷移を調べた結果，その約95%がTPNの表現す

る話題遷移でカバーされており，このモデルが妥当であることが示された．

また，対話管理部における対話処理は，この二つの対話構造モデルとSOCSにおけるテンプレートなどに関する知識を利用しながら，概念表現を修正することによって行なわれる．具体的な対話処理として，(1)概念表現の比較あるいはSR-プランを用いることによって行なう強調語句の抽出，(2) 音声対話システムではユーザ発話の認識結果をユーザに確認する必要があるという立場から必要な認識結果提示のための語句の補充，(3)慣用テンプレートを用いた表現への概念表現の書き換え，(4) 韻律オペレータの追加の例を挙げ，概念表現を用いることにより対話処理が容易に行なえることを示した．しかし，これらの対話処理は，汎用なインタフェースにおいて行なうべき対話処理の一部を拾い挙げたに過ぎない．今後，語順や韻律パラメータの変更など，その他の対話処理についても検討すると共に，汎用インタフェースで扱うべき処理を明確にしていくことが必要となる．

以上，本研究では高品質な対話音声合成を実現するための基礎的な研究について述べた．対話音声合成はもちろんそれだけで一つのインタフェース技術になり得るが，やはり音声対話システムを実現するために，対話音声認識と組合せた音声インタフェースとして確立すべきで技術である．本論文で述べた対話構造のモデルは，音声出力だけにしか利用できないものではなく，音声入力においてもユーザ発話の次発話予測などに有効である[Yamamoto90][平松92]．従来は，音声認識と音声合成がそれぞれ別の研究として扱われてきたが，今後音声対話の実現を目指してこれらを融合した研究が必要になるであろう．

謝　辞

本研究の全過程を通じ，直接懇切なる御指導・御鞭撻を賜った大阪大学産業科学研究所溝口理一郎教授，ならびに龍谷大学理工学部角所収教授に衷心より感謝の意を表する．

本研究をまとめるに当たり，貴重な御助言と御教示を賜った大阪大学工学部電子工学科児玉慎三教授，情報システム工学科寺田浩詔教授，白川功教授，ならびに貴重な御助言と御指摘を賜った電子工学科吉野勝美教授，裏克己教授，濱口智尋教授，西原浩教授，尾浦憲治郎教授，情報システム工学科藤岡弘教授に厚くお礼申し上げる．

本研究の遂行に当たり，終始適切な御助言と御鞭撻を頂いた郵政省通信総合研究所関西先端研究センター知覚機構研究室柳田益造室長，ならびに有益な御助言を頂いた大阪大学産業科学研究所豊田順一教授，北橋忠宏教授，神戸大学工学部上原邦昭助教授，静岡大学工学部山口高平助教授，摂南大学国際言語文化学部中嶋鴻毅講師，大阪大学産業科学研究所池田満助手に厚く感謝の意を表する．

本研究の遂行に当たり，音声データの提供など多大なご援助を頂き，かつ熱心に御討論して頂いたシャープ株式会社中央研究所部長藤本好司氏 (現在複写機事業部)，主任研究員鬼頭淳悟氏(現在情報技術開発センター)，海木延佳氏(現在情報技術開発センター)，福田尚行氏(現在IC開発センター)，谷口賢一氏(現在九州松下電器株式会社)，水谷直樹氏 (現在大阪大学経済学部在学中) に心から感謝する．

また，本研究の遂行には，日本音響学会の研究用連続音声データベースの一部を使用した．さらに，大阪大学産業科学研究所溝口研究室，ならびに関西大学工学部電子工学科情報工学研究室の諸氏には，模擬対話の収録など種々の面でお世話になった．ここに記して深く感謝する．

参考文献

[阿部81] 阿部芳春, 今井聖: "CV 音節のケプストラムパラメータからの音声合成", 電子通信学会論文誌, J64-D, 9, pp.861-868 (1981).

[尼子91] 尼子喜健, 太田義一, 山下洋一, 溝口理一郎: "決定木を用いた長い名詞句に対するピッチパタン制御", 電子情報通信学会技術報告, SP91-77, pp.7-14 (1991).

[飯田88] 飯田仁: "自然言語対話の言語運用特性と対話処理の研究課題", 人工知能学会誌, 3, 4, pp.445-452 (1988).

[飯田90] 飯田仁, 有田英一: "4階層プラン認識モデルを使った対話の理解", 情報処理学会論文誌, 31, 6, pp.810-821 (1990).

[石井86] 石井直樹: "音声合成研究の問題点と今後の方向", 日本音響学会誌, 42, 12, pp.930-935 (1986).

[大泉72] 大泉充郎, 藤村靖: "音声科学", 東京大学出版会 (1972).

[海木92] 海木延佳, 武田一哉, 匂坂芳典: "言語情報を利用した母音継続時間長の制御", 電子情報通信学会論文誌, J75-A, 3, pp.467-473 (1992).

[柏岡89] 柏岡秀紀, 平井誠, 北橋忠宏: "発話対に基づく対話構造のモデルと知識の利用", 1989年度人工知能学会全国大会論文集, 8-1, pp.415-418 (1989).

[河合87] 河合徹: "機械が抑揚を持って話す文章・音声変換装置が実用化へ", 日経コンピュータ, 8月17日号, pp.163-170 (1987).

[菊池92] 菊池英明, 小林哲則, 白井克彦: "自然な模擬対話を収録するために", 科研費総合研究(A)『音声対話』成果報告書, pp.109-118 (1992).

[小林92] 小林哲則, 板橋秀一, 速水悟, 竹澤寿幸: "日本音響学会研究用連続音声データベース", 日本音響学会誌, 48, 12, pp.888-893 (1992).

[斎藤81] 斎藤収三, 中田和男: "音声情報処理の基礎", オーム社 (1981).

[坂井90] 坂井信輔, 村木一至, 岩田和彦: "中間言語からの韻律情報の生成", 日本

音響学会春季講演論文集, 2-4-9, pp.243-244 (1990).

[匂坂85] 匂坂芳典, 佐藤大和: "テキスト・音声変換技術の現状と課題", 日本音響学会誌, 41, 12, pp.901-905 (1985).

[匂坂89] 匂坂芳典: "統語構造に基づくF0パタン概形の制御", 日本音響学会秋季講演論文集, 3-P-13, pp.301-302 (1989).

[佐藤78] 佐藤大和: "PARCOR-VCV連鎖を用いた音声合成方式", 電子通信学会論文誌, J61-D, 11, pp.858-865 (1978).

[管村81] 管村昇, 板倉文忠: "線スペクトル対(LSP)音声分析合成方式による音声情報圧縮", 電子通信学会論文誌, J64-A, 8, pp.599-606 (1981).

[武田90] 武田一哉, 安部勝雄, 匂坂芳典: "選択的に合成単位を用いる規則音声合成", 電子情報通信学会論文誌, J73-D-II, 12, pp.1945-1951 (1990).

[武田91] 武田昌一, 市川熹: "日本語文音声におけるプロミネンスの韻律的特徴の解析", 日本音響学会誌, 47, 6, pp.386-396 (1991).

[田中92] 田中和世 他: "音声の知的処理に関する調査研究", 第4部, (財)日本情報処理開発協会 (1992).

[辻野89] 辻野克彦, 竹之内正一郎, 桜井徹, 千種俊輔, 野村康雄, 溝口理一郎, 角所収: "適応的ルールインダクションシステム: ARIS", 電子情報通信学会論文誌, J72-D-II, 1, pp.121-131 (1989).

[徳永91] 徳永健伸, 乾健太郎: "1980年代の自然言語生成 --1--", 人工知能学会誌, 6, 3, pp.380-387 (1991).

[中嶌89] 中嶌信弥, 浜田洋: "音韻環境に基づくクラスタリングによる規則合成法", 電子情報通信学会論文誌, J72-D-II, 8, pp.1174-1179 (1989).

[中津90] 中津良平: "音声認識・合成技術の製品化および市場動向", 電子情報通信学会技術報告, SP89-103, pp.39-46 (1990).

[箱田80] 箱田和雄, 佐藤大和: "文音声合成における音調規則", 電子通信学会論文誌, J63-D, 9, pp.715-722 (1980).

[速水91] 速水悟, 伊藤克亘, 田中和世: "音声対話システムの構築とそれを用いた会話音声収集", 電子情報通信学会技術報告, SP91-101, pp.79-86 (1991).

[樋口89] 樋口宜男, 山本誠一, 清水徹: "パラメータ導出型日本語音声規則合成方式における調音制御", 日本音響学会誌, 45, 6, pp.426-433 (1989).

[平松92] 平松敬史, 吉田英昭, 野村康雄, 山下洋一, 溝口理一郎: "音声対話理解のための話題知識の利用", 電子情報通信学会音声研究会資料, SP92-110, pp.55-62 (1992).

[広川88] 広川智久: "波形辞書を用いた規則合成法", 電子情報通信学会技術報告, SP88-9, pp.65-72 (1988).

[広瀬92] 広瀬啓吉: "音声合成の研究の現状と将来", 日本音響学会誌, 48, 1, pp.39-45 (1992).

[藤崎90] 藤崎博也, 広瀬啓吉, 浅野康治: "知識表現からの文章音声合成システム", 日本音響学会秋季講演論文集, 2-6-6, p.231-232 (1990).

[松山92] 松山広嗣, 田島慶一, 野村康雄, 山下洋一, 溝口理一郎: "音声対話におけるシステム発話の対話コンテキスト依存性", 平成4年電気関係学会関西支部連合大会講演論文集, G16-6, G430 (1992).

[三村91] 三村克彦, 海木延佳, 匂坂芳典: "統計的手法を用いた音声パワーの分析と動特性の制御", 電子情報通信学会技術報告, SP91-4, pp.25-32 (1991).

[山岡90] 山岡孝行, 飯田仁: "文脈を考慮した音声認識結果絞り込み手法", 情報処理学会自然言語処理研究会報告, 78-16, pp.121-128 (1990).

[山下89] 山下洋一, 溝口理一郎: "概念表現を利用した合成音声出力", 日本音響学会秋季講演論文集, 3-P-22, pp.319-320 (1989).

[山下90] 山下洋一, 水谷直樹, 溝口理一郎: "合成音出力における概念表現の利用", 電子情報通信学会音声研究会資料, SP89-115, pp.41-48 (1990).

[山本85] 山本誠一, 樋口宜男, 松崎一博: "概念からの音声合成のための実験システム", 日本音響学会秋季講演論文集, 2-3-16, pp.185-186 (1985).

[Allen87] J.Allen, M.S.Hunnicutt, D.Klatt: "From Text to Speech: The MITalk System", Cambridge University Press (1987).

[Grosz86] B.J.Grosz and C.L.Sidner: "Attention, Intentions, and the Structure of Discourse", Comp. Linguist., 12, 3, pp.175-204 91986).

[Hauptmann88] A.G.Hauptmann, S.R.Young and W.H.Ward: "Using Dialog-Level Knowledge Sources to Improve Speech Recognition", Proc. of AAAI '88, pp.729-733 (1988).

[Hertz85] S.R.Hertz, J.Kadin and K.J.Karplus: "The Delta Rule Development System for Speech Synthesis from Text", Proc. of the IEEE, 73, 11, pp.1589-1602

(1985).

[Hirschberg90] J.Hirschberg: "Accent and Discourse Context: Assigning Pitch Accent in Synthetic Speech", Proc. of AAAI '90, pp.952-957 (1990).

[Kellermann89] K.Kellermann, S.Broetzmann, T.S.Lim and K.Kitao: "The Conversation Mop: Scenes in the Stream of Discourse," Discourse Processes, 12, pp.27-61 (1989).

[Klatt80] D.H.Klatt: "Software for a Cascade/Parallel Formant Synthesizer", J. Acoust. Aoc. Am., 67, 3, pp.971-995 (1980).

[Klatt87] D.H.Klatt: "Review of Text-to-speech Conversion for English", J. Acoust. Aoc. Am., 82, 3, pp.737-793 (1987).

[Litman87] D.J.Litman and J.F.Allen: "A Plan Recognition Model for Subdialogues in Conversations," Cognitive Science, 11, pp.163-200 (1987).

[Riley89] R.D.Riley: "Tree-Based Modelling for Speech Synthesis", Proc. of the ESCA Workshop on Speech Synthesis, Autrans (France), pp.229-232 (1990).

[Schank75] R.C.Schank: "Conceptual Information Processing", Volume 3 of Fundamental Studies in Computer Science, North-Holland (1975).

[Schank77] R.C.Schank and R.P.Abelson: "Scripts, Plans, Goals, and Understanding", Hillsdale, N.J.: Lawrence Erlbaum (1977).

[Schank82] R.C.Schank: "Dynamic memory: A Theory of Reminding and Learning in Computers and People", Cambridge University Press (1982).

[Streeter88] L.A.Streeter: "Applying Speech Synthesis to User Interfaces", Handbook of Human-Computer Interaction, Elsevier Science Publishers B.V. (North-Holland), pp.321-343 (1988).

[Yamamoto91] T.Yamamoto, Y.Ohta, Y.Yamashita, O.Kakusho and R.Mizoguchi: "MASCOTS: Dialog Management System for Speech Understanding System", IEICE trans., E74, 7, pp.1881-1888 (1991).

[Young79] S.J.Young and F.Fallside: "Speech Synthesis from Concept: A Method for Speech Output from Information Systems", J.Acoust.Soc.Am., 66, 3, pp.685-695 (1979).