文書処理とヒューマンインタフェース 20 － 2
（１９８８. ９. ８）

日本語文章作成支援システムＣＯＭＥＴ
－文章解析応用の統合化方式を中心に－

福島 俊一　　大山 裕

日本電気株式会社　Ｃ＆Ｃシステム研究所

日本語の文章を計算機で解析する技術は、かな漢字変換、文章チェッカ、文音声変換、キーワード抽出など、様々な応用に結び付いている。そして、近年、広く普及した日本語ワードプロセッサを発展させ、さらに高度な文章作成環境を実現するためには、それら文章解析応用を統合化することが必要になる。本稿では、その統合化方式を提案し、その方式にもとづいて試作した日本語文章作成支援システムＣＯＭＥＴを紹介する。提案する統合化方式では、文章解析の結果として得られる文章情報（文章の階層構造、形態素情報）を、文章とともに保持する。そして、その文章情報は、文章の編集が行なわれても、局所的再解析により更新される。このとき、各種文章解析応用は、個別に文章解析処理を行なう必要がなく、システム全体として計算効率がよくなる。

A TEXT PROCESSING SYSTEM ARCHITECTURE FOR THE INTEGRATION OF TEXT ANALYSIS APPLICATIONS

Toshikazu FUKUSHIMA and Yutaka OHYAMA
C&C Systems Research Laboratories, NEC Corporation
1-1, Miyazaki 4-chome, Miyamae-ku, Kawasaki-city, Kanagawa 213, Japan

Japanese text analysis technology can be applied to text processing in various ways. Such as well-known applications have been developed as Kana-to-Kanji translation, text critiquing, text-to-speech conversion, keyword extraction and so on. This paper proposes a text processing system architecture for the integration of text analysis applications, and also introduces a Japanese text processing system based on this architecture. The kernel of this architecture is an analysis result storage form, "S-text", which contains the hierarchical text structure and lexical information. "S-text" is not only generated but also modified while text-editing. This architecture eliminates wasteful repetition of the same analysis process for each application.

1　はじめに

日本語の文章を計算機で解析する技術は、日本語情報処理において、様々な応用に結び付いている。例えば、かな漢字変換、文章チェッカ（校正支援）、文音声変換、キーワード抽出、機械翻訳などは、いずれも文章解析技術に支えられた応用である[1][2]。これらの応用は、従来、個別のシステムとして研究開発・実用化されてきたが、筆者らは、それらを統合化することによって、高度な文章作成環境を構築することを試みている[3]-[6]。

この文章解析応用を統合化した文章作成環境は、広く普及した日本語ワードプロセッサの、今後、発展してゆく方向の１つと考えられる。日本語ワードプロセッサは、当初、単に、日本語のエディタであるだけで、大きな価値をもっていた。しかし、計算機を用いた文章作成が一般に浸透するにつれて、単なるエディタの機能にとどまらない、様々な知的支援機能の提供が望まれてきている。そのとき、日本語入力手段としてのかな漢字変換だけでなく、作成した文章を校正するための文章チェッカ、文章を読み上げる文音声変換、データベース化する際のキーワード抽出など、各種の文章解析応用を１つのシステム上に統合化してゆくことが必要になってくる。

さて、計算機による日本語文章の解析処理には、高い計算コストがかかる。したがって、文章解析応用を統合化した場合、応用ごとに、文章解析処理を繰り返すのでは、計算コストの面で非常に効率が悪い（図１参照）。また、応用ごとの処理に要する時間も長くなる。この効率の悪い処理反復を回避するためには、文章解析処理が一度実行されたら、それ以後、各種応用で共通に必要となるような文章解析結果の情報を保持するようにすればよい（図２参照）。そのとき、各種応用は、すでに解析された文章を対象とした、その応用に固有の処理のみを行なえばよく、処理時間も短縮される。

上記の考えを導入したシステムの１つのタイプとして、かな漢字変換入力の際に得られる単語の境界・読み・表記などの情報を、文章ファイル内に保持するものがある[7]。しかし、単語の情報は、文章の編集（特に、単語・文節の途中位置への挿入や、その一部分の削除）によって壊されることがあり、このタイプのシステムでは、文章が編集されるにつれて、単語の情報が失われてゆくことになる。一方、作成された漢字かな混じり文章の一括解析を行ない、得られた単語の表記・読み・品詞などの情報をファイル化するタイプのシステムがある[8]。このタイプのシステムでは、入力・編集の完了した文章が対象に想定されており、文章の一部分にでも修正が加えられた場合には、文章全体の再度の解析が必要となる。

そこで、本稿では、一度行なった文章解析の結果の情報を、そのまま保持するだけでなく、文章の編集に応じて更新する機構を導入した、文章解析応用の統合化方式を提案する。この方式では、入力・編集が行なわれても、文章解析結果の情報が常時保持されるので、各種応用処理の計算コストが抑えられるだけでなく、文章の編集の際にも文章解析結果の情報を利用することが可能となる[5]。以下、第２章で保持する文章情報を、第３章で文章情報の生成・更新機構を、それぞれ説明することによって、提案する方式を示す。そして、第４章で方式を具体化した日本語文章作成支援システムＣＯＭＥＴ（Computer Assisted Environment for Japanese Text Creation）[3]-[6]を紹介し、第５章で方式の有効性・課題についてまとめる。

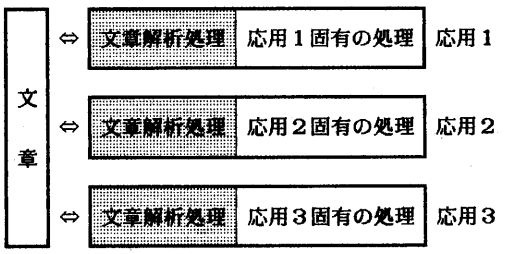

図１　計算効率の悪い統合化

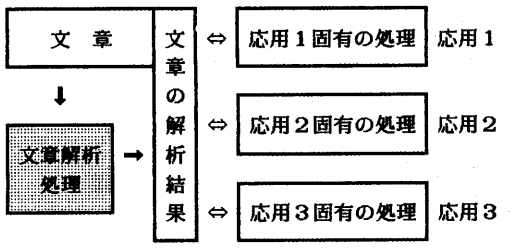

図２　計算効率を改善した統合化

2　Sテキスト

本章では、各種文章解析応用で共通に必要となる文章情報（文章解析の結果）として、どのような情報を取り上げるかを示す。前章で述べたように、この文章情報は、文章解析応用の統合化システムにおいて保持され、各種応用の計算コストを軽減する。この文章情報を伴った文章（テキスト）を、本稿では、Sテキスト（Substantial Text または Structured Text[8]）と呼ぶ。

2－1　文章情報の選出

個々の文章解析応用で必要となる文章情報が、Sテキストに多く収められていればいるほど、その応用自身の処理の計算コストは軽減される。しかし、その分、図2に示したような共有の文章解析処理の、負荷が重くなる。ある特定の応用でのみ必要となるような文章情報の生成を、共有の文章解析処理が多く担うようになると、使用されない文章情報の生成のために計算時間を長く費やすことになり、システム全体として、計算効率が低下する。したがって、Sテキストに収める文章情報は、各種文章解析応用において、できる限り共通に必要とされる情報であることが望ましい。その場合、個々の応用の処理に分散していた同等の処理が、共通化された文章解析処理内に一本化されるので、システム全体として、計算効率がよくなる。

一方、文章の編集に応じてSテキストの内容を更新することを考えると、一箇所の修正が引き起こす変更の範囲が不確定であったり、きわめて広かったりすると都合が悪い。例えば、Sテキスト内に、文章中の代名詞の指示先の情報まで収めていたとすると、ある名詞を削除するような編集が行なわれた場合、その名詞を指していた代名詞を捜して、その代名詞の指示先を判定し直すことまでが必要となる。したがって、Sテキスト内に収める文章情報は、ある限定された範囲の文章から決定できることが望ましい。

以上の検討をまとめると、Sテキスト内に収める文章情報は、次のような条件を満たすことが好ましい。

Ⓐ 各種文章解析応用に共通的に必要とされる。

Ⓑ 限定された範囲の文章から決定できる。

そこで、筆者らは、Sテキストに収める文章情報として、①文章の階層構造と、②形態素情報との2種類の情報を取り上げることにした。

ここでは、文章情報を、単語＜文節＜文＜……といった階層構造（どこからどこまでが、どの階層の単位に対応するかという情報）と、階層構造の個々の単位がもつ情報（文章における役割・意味、他の単位との依存関係）とに分けて考えている。①は前者であり、②は後者の一部である。後者には、②以外に、修飾関係・格関係の情報なども属するが、それらはSテキストに収めない（条件Ⓑによる）。

以下の節では、①②の詳細について述べる。

2－2　文章の階層構造

Sテキストに収める文章の階層構造を、次のように定義する。記述はBNF記法を用いており、A｜BはAまたはBを、{A}はAの1個以上の繰り返し（すなわち、{A}≡A｜AA｜AAA｜……）を、[A]はAが省略可能であることを意味する。

〈文章〉　::＝{〈節〉}
〈節〉　::＝[〈節題〉]〈節本体〉
〈節題〉　::＝〈文〉〈論理デリミタ〉
〈節本体〉::＝{〈節〉}｜〈節文〉
〈節文〉　::＝{〈段落〉}
〈段落〉　::＝{〈文〉}〈論理デリミタ〉
〈文〉　::＝{〈文節〉}
〈文節〉　::＝{〈単語〉}
〈単語〉　::＝{〈文字〉}

ここに示した文章の階層構造は、ＩＳＯの提案する文書モデル[9]における論理構造に対応する。ただし、本稿では一次元の文字列の形態をとる文章を対象としているので、上記の定義では、図表や脚注などを扱っていない。

上記の定義における〈論理デリミタ〉は、句読点や括弧などの記号ではなく、通常、改行や空白などで明示されるような、論理的な切れ目を意味する。また、上記の定義における〈節〉は、文章内の単位をさす名称として一般に用いられている「節」よりも、広い意味をもっている。〈節〉は、再帰的な定義によって、一般に「章」や「節」と名付けられるような特定の階層だけでなく、段落よりも上位のすべての階層の単位をさすようになっている。したがって、上記の定義では、段落より上位の階層の数に制限はない。

図3と図4に文章の階層構造の例を示す。図3には

段落以上の階層が示されており、図４には段落以下の　　している。
階層が示されている。なお、↵は論理デリミタを表わ

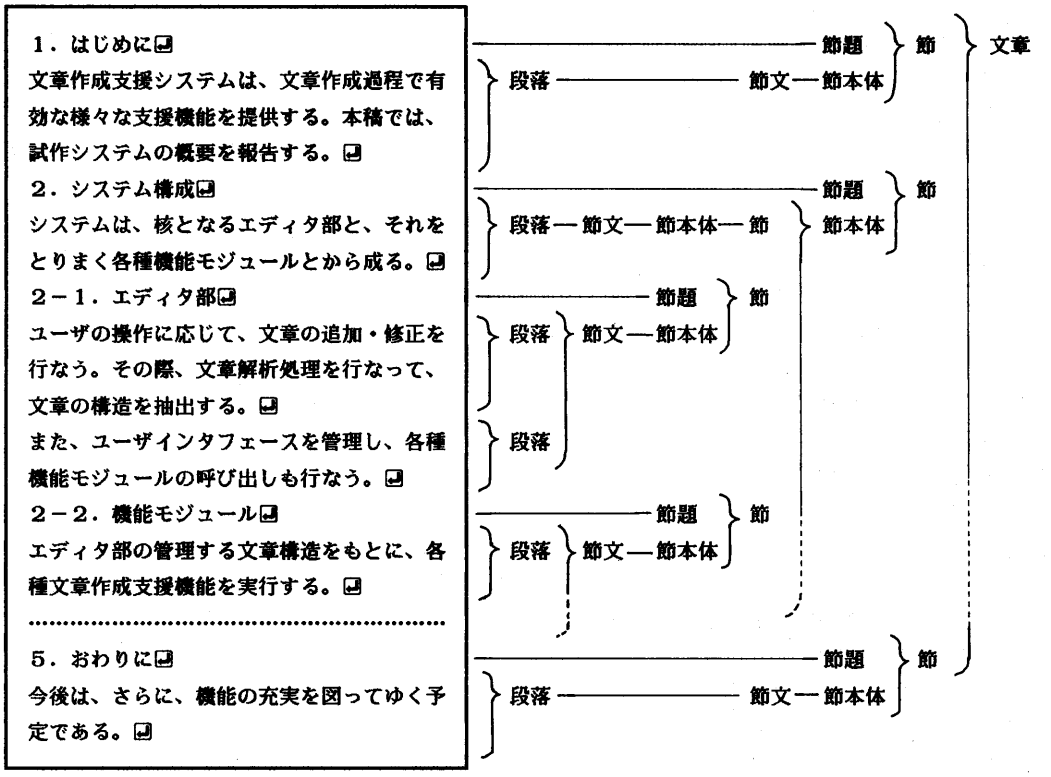

図３　文章の階層構造（段落以上の階層）の例

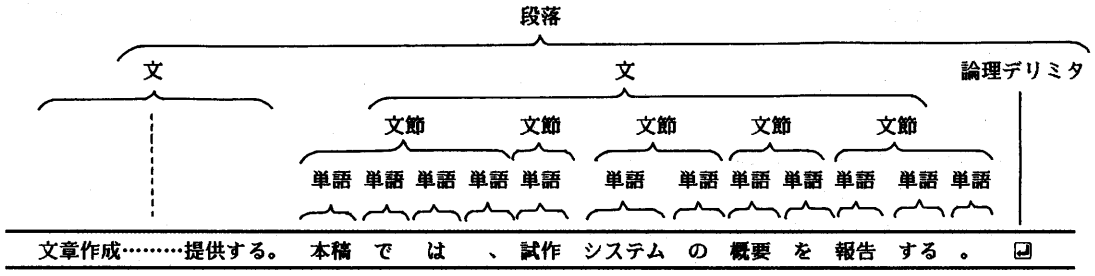

図４　文章の階層構造（段落以下の階層）の例

	単語	単語	単語	単語	単語	単語	単語	単語	単語	単語	単語	単語
表記 →	本稿	で	は	、	試作	システム	の	概要	を	報告	する	。
品詞 →	名詞	助詞	助詞	読点	サ変名詞	名詞	助詞	名詞	助詞	サ変名詞	サ変語尾	句点
読み →	ホンコウ	デ	ハ	、	シサク	システム	ノ	ガイヨウ	ヲ	ホウコク	スル	。

図５　基本的な形態素情報の例

２－３　形態素情報

Ｓテキストには、さらに、形態素情報を収める。

この形態素情報は、単語の表記・読み・品詞を基本とする。図５に、基本的な形態素情報の例を示す。表記の部分を連結した一次元の文字列が、漢字かな混じりの文章に対応する。また、この形態素情報における読みは、発音をそのまま記述したものではなく、かな漢字変換の際の入力となるような、かな表記を意味する。したがって、助詞の「は」や「を」の読みとしては、「ハ」や「ヲ」を収め（「ワ」や「オ」とはしない）、句読点や英数字の読みは、表記（、。，．ＡＢＣ……１２３……）と同様とする（「テン」「マル」「エイ」「イチ」などとはしない）。

上記の基本的な形態素情報は、通常、形態素解析の過程で、単語辞書から検索されて得られる。そこで、その際、同時に単語辞書から検索できる情報であれば、拡張した形態素情報として、Ｓテキストに収めてもかまわない。拡張した形態素情報としては、例えば、単語のアクセントをあげることができる（応用の１つに、文音声変換を想定した場合）。

３　Ｓテキストの生成・更新機構

前章で述べたように、Ｓテキストには、文章の階層構造と形態素情報とを収める。本章では、それらの文章情報を、どのようにして生成し、かつ、文章の編集に応じて、どのように更新してゆくかについて述べる。

３－１　システム構成

図６に、Ｓテキストの生成・更新機構を実現するためのシステム構成を示す。この構成において、次の５つのモジュールが、Ｓテキスト内の文章情報（文章の階層構造、形態素情報）を生成・更新する。これらのモジュールは、生成・更新の対象とする文章情報の分担を行なっている。

① かな漢字変換モジュール（読みがな列をもとに、単語・文節の単位を決定し、形態素情報を抽出する）
② 形態素解析モジュール（漢字かな混じり表記をもとに、単語・文節の単位を決定し、形態素情報を抽出する）
③ 文分割モジュール（文の単位を決定する）
④ 段落分割モジュール（段落の単位を決定する）
⑤ 節階層認識モジュール（節題・節文などの単位やそれらの階層関係を決定する）

これら５つのモジュールは文章解析に関する互いに独立な知識源であり、図６は黒板モデル[10]にもとづいた構成と考えることもできる（Ｓテキストが黒板上に置かれていることになる）。モジュール②③④⑤は、Ｓテキストを監視し、その内容にもとづいて動作し、結果をＳテキストに書き込む。モジュール①、文章編集モジュール、各種文章解析応用モジュールは、ユーザのキー操作に応じて起動され、Ｓテキストの内容をもとに、個々の処理を行なう。

３－２　文章解析知識

文章情報を生成・更新する５つのモジュールは、各々が分担する文章情報に応じた文章解析知識をもつ。以下に、各モジュールのもつ知識を簡単に述べる。ただし、①かな漢字変換モジュールと②形態素解析モジュールとは、入力が、読みがな列か、漢字かな混じり表記かが異なるだけで、同じく単語・文節の単位の決定と形態素情報の抽出を行なうので、基本的な知識は共通である。

Ⓐ 形態素解析知識（→モジュール①②）

まず、単語辞書と文字列とを照合することにより、

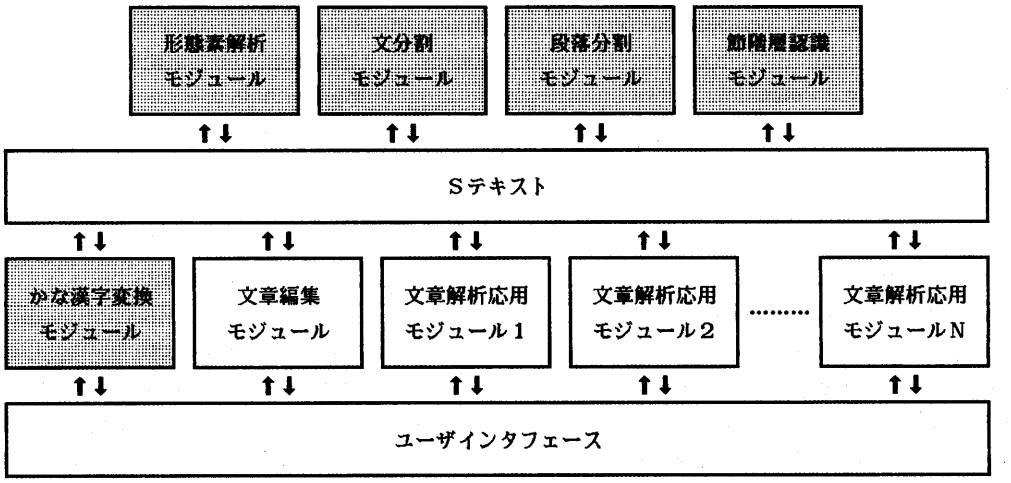

図6　システム構成

文字列中に存在する可能性のある単語の候補を検出する。次に、文節の先頭になり得る単語（品詞）、文節の末尾になり得る単語（品詞）、文節内で接続し得る単語（品詞）の組を記述した接続表により、文節を構成し得る単語の系列を求める。その結果、文字列は単語・文節に分割され、形態素情報が単語辞書より得られる。同一の文字列に対して、複数通りの可能性が生じた場合には、ヒューリスティックスを適用して、最も確からしいものを選択する。

Ⓑ 文分割知識（→モジュール③）

文の単位は、通常、句点・ピリオド・括弧類・論理デリミタなどで明示される。そこで、それらを検出し、その各々について、次のような判定を行なう。句点の直後や、論理デリミタの前後を、無条件に文の境界とする。ピリオドや括弧類などの直後は、文の境界の候補とするが、小数点や文中の強調・補足などにも用いられるので、最終的には、形態素解析の結果を考慮して決定する。

Ⓒ 段落分割知識（→モジュール④）

段落の単位は、論理デリミタで明示される（2－2の定義を参照）。そこで、論理デリミタを検出して、その直後を段落の境界とする。

Ⓓ 節階層認識知識（→モジュール⑤）

節題は、通常、他の部分（節文）と容易に識別できるような表記法がとられる。また、2－2の定義では、Ⓒで段落とされたもののうち、段落内の文の数が1個のみのものが、節題の候補となる。そこで、その節題の候補を検出して、表記法の特徴（節番号、文末表現、文長など）をもとに、節題の単位を決定する。そして、節題以外の部分は、節文とみなす。次に、節題に付加された節番号（第1章、2－1など）や先頭記号（●、◆など）のタイプの相違と位置関係をもとに、節題の階層レベルを判定する。

3－3　Sテキストの新規生成

Sテキストを新規に生成する際には、かな漢字変換を行なって、キーボードから直接入力する方法と、既に他のワードプロセッサなどで作成された漢字かな混じり文章（一次元の文字列の形式）のファイルを読み込む方法とをとれる。

キーボードから直接入力される場合には、かな漢字変換モジュールが動作し、その変換処理で得られた文節・単語の単位と、形態素情報（単語の読み・表記・品詞）とを、Sテキストに書き込む。それらがSテキストに書き込まれると、文分割モジュール、段落分割モジュール、節階層認識モジュールが、各々独立に動作し、Sテキストに、文・段落・節などの単位、節の階層関係の情報を追加する。

漢字かな混じり文章のファイルが読み込まれる場合には、まず、文章編集モジュールが動作し、Sテキストに表記の文字列を書き込む。Sテキストの表記の文字列が書き込まれると、形態素解析モジュール、文分割モジュール、段落分割モジュール、節階層認識モジュールが、各々独立に動作し、単語・文節・文・段落・節などの単位、節の階層関係の情報、単語の読み・品詞などを生成し、Sテキストに追加する。

かな漢字変換モジュールが動作して、形態素情報などを生成した場合には、形態素解析モジュールは、原則として動作しない（かな漢字変換の際の解析結果を有効利用する）。ただし、漢字かな混じり文章のファイルが読み込まれた場合や、次節で述べるような文章の編集が行なわれた場合には、かな漢字変換モジュールでは解析が行なえず、漢字かな混じり文字列を対象とした形態素解析モジュールが必要となる。

３－４　編集に応じたSテキストの更新

文章の編集が行なわれると、Sテキスト内の文章情報が壊れてしまう（情報が失われるか、または、誤ったものになる）ことが多い。以下に、いくつかの例をあげる。

［例１］　「音楽が聞こえる。」の「楽」１文字が削除される場合、「音楽」という単語（名詞）の情報は無効になり、残る「音」に対する情報が欠落する。

［例２］　「文音声変換を行なう。」の「文」（名詞）と「音声」（名詞）の間に、かな漢字変換により、１文字の「章」（名詞）が挿入される場合、「文章音声変換……」において、「文」と「章」とは別の単語になり、「文章」という１単語（名詞）にならない。

［例３］　句点や論理デリミタの削除・挿入が行なわれた場合には、文や段落の単位が、連結あるいは分断される必要がある。

［例４］　節題である「２－２ 文章の階層構造」の階層レベルは、「２－１ 文章情報の選出」と同じであるが、「２－２」が「２－１－１」に変更された場合、その階層レベルは「２－１ 文章情報の選出」の下位に変更される必要がある。

そこで、そのような文章の編集が行なわれた場合には、Sテキスト中で文章情報の壊れた箇所を再解析して、文章情報を生成し直す必要がある。そのため、形態素解析モジュール、文分割モジュール、段落分割モジュール、節階層認識モジュールの４モジュール（かな漢字変換モジュールは除く）は、Sテキスト内の状態を監視し、必要に応じて再解析を行なう。

その際、再解析する文章の範囲は、できる限り局所的に抑えることが好ましい。ここで、文章解析知識の各々を考えると、再解析を行なう文章の範囲は、各モジュールの対象とする基本単位（形態素解析モジュールは文節、文分割モジュールは文、段落分割モジュールは段落、節階層認識モジュールは節）の１つ上位の階層の単位内で、該当箇所から前後１個ずつの基本単位をあわせた範囲とすれば十分である。

この局所的再解析の例を、形態素解析モジュールについて、前記の例１と例２を用いて示す。例１では、形態素情報の欠落した文節の「音が」と、その直後の文節の「聞こえる。」とをあわせた範囲（直前は文の先頭なので後方のみ含める）の「音が聞こえる。」を再解析する。例２では、挿入された文節の「章」と、直前の文節の「文」と、直後の文節の「音声」とをあわせた範囲の「文章音声」を再解析する。例１のように１文節の範囲（例１では「音が」）を再解析すれば足りそうな場合もあるが、前後の１文節も含めて再解

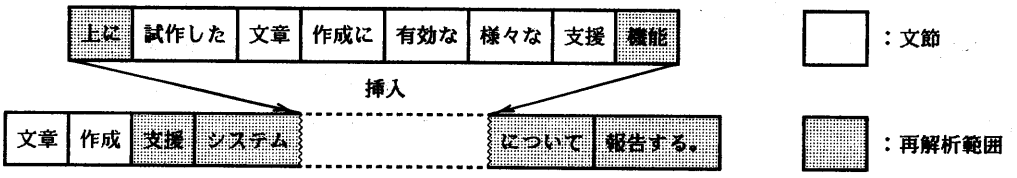

図７　挿入時の再解析範囲の例

析するようにしておけば、前後の文節を考慮した解析候補の選択が行なえる。

また、例２では、かな漢字変換により挿入するのが１文節だけであったため、挿入文字列のすべてが再解析されているが、複数文節を挿入する場合には、挿入先の文章と挿入する文章とのつなぎ目の部分の、前後１文節の範囲のみを再解析する。例えば、「文章作成支援システムについて報告する。」という文字列の「システム」と「について」の間に、かな漢字変換入力を用いて、「上に試作した文章作成に有効な様々な支援機能」という文字列を挿入した場合、形態素解析モジュールが再解析する範囲は、「支援システム上に」と「機能について報告する。」の２箇所である（図７参照）。したがって、「試作した文章作成に有効な様々な支援」の部分については、かな漢字変換の解析結果を、そのまま用いる。複写や移動などの編集操作が実行された場合も同様で、複写や移動の結果として生じたつなぎ目の、前後１文節をあわせた範囲を、再解析する。

以上のようにして、文章の編集に応じたＳテキストの更新が可能となる。

４　日本語文章作成支援システムＣＯＭＥＴ

第２章および第３章で述べたＳテキストとその生成・更新機構は、日本語文章作成支援システムＣＯＭＥＴ[3]-[6]において具体化されている。その結果、ＣＯＭＥＴでは、文章チェッカ、文音声変換、ＫＷＩＣ作成などの文章解析応用が統合化され、文章作成に有効な様々な支援機能が実現されている。

ＣＯＭＥＴは、パーソナルコンピュータＰＣ－９８ＸＡ上（ＣＰＵ：８０２８６）に、Ｃ言語を用いて構築したシステムである。図８にその外観を示した。音声合成用に、音声合成ボード[11]とアンプ付きスピーカが付加されている。

以下では、Ｓテキストの実現形態と、試作されている各種支援機能を紹介する。

４－１　Ｓテキストファイル

ＣＯＭＥＴでは、Ｓテキストを、表記文字列ファイル、単語情報ファイル、読みがなファイル、節階層ファイルという４つのファイルで実現している。

表記文字列ファイルは、単語の表記を連結した一次元の漢字かな混じり文字列である（ＭＳ－ＤＯＳの標準テキストファイルと同等）。単語情報ファイルは、表記文字列ファイルの各文字と１対１に対応するセルの並びである。そして、単語・文節・文・段落の先頭を表わすコードを、該当するセルに収める。さらに、単語の先頭位置に対応するセルには、単語の品詞とアクセントをコード化して収める（アクセントは、文音声変換のための拡張形態素情報である）。読みがなファイルは、単語の読みを、デリミタをはさんで連結した文字列である。節階層ファイルは、段落より上位の木構造を表現したものである。

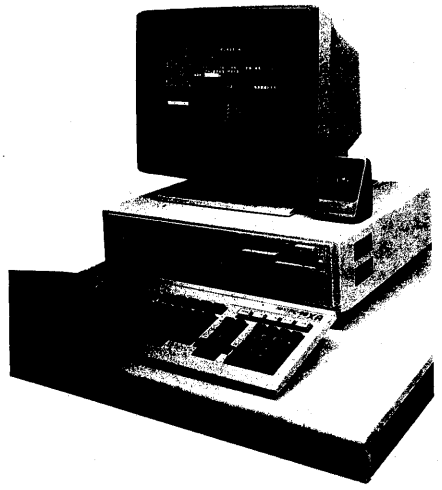

図８　ＣＯＭＥＴの外観

このように文章情報に応じてファイルを分離することによって、他のシステムとのデータ交換を容易にしている。

４－２　文章作成支援機能

ＣＯＭＥＴでは、Ｓテキストに収められた文章情報を利用して、以下のような、様々な文章作成支援機能が実現されている。これらの機能の多くは、これまでに文献[3]-[6]などで報告しているので、ここでは、詳細を省略する。

(1) 文章構造を利用した表示・編集機能

◆ アウトライン表示（図９参照）

◆ アウトライン編集（節単位の削除・移動・複写）

◆ 段落・文・文節・単語単位のカーソル移動
◆ 段落・文・文節・単語単位の削除・移動・複写
◆ 品詞・活用を考慮した単語検索
◆ 段落の省略表示（段落の先頭文以外を省略）
◆ 文の省略表示（文頭・文末以外を省略）
◆ かな漢字変換における再変換

(2) 誤字・文法的誤りの検出機能
◆ 形態素解析失敗箇所の検出
◆ 誤用辞書に登録した語の検出
◆ 呼応していない呼応の副詞の検出

(3) スタイルの検査・書き換え機能
◆ 長すぎる文の検出
◆ 読点間隔の長すぎる箇所の検出
◆ 同一字種の長く連続した箇所の検出
◆ 長すぎる段落の検出
◆ 文体（常体、敬体）の不統一箇所の検出
◆ 文体の相互書き換え（常体⇄敬体）
◆ 動詞の敬語表現への書き換え（図１０参照）
◆ 句読点のゆれの検出・統一
◆ 正しく対応していない括弧類の検出
◆ 節番号の不正箇所の検出
◆ 平均文長・漢字含有率の評価

(4) ＫＷＩＣの表示機能、表記のゆれの検出機能
◆ 自立語をキーワードとしたＫＷＩＣの表示
◆ 自立語表記のゆれの検出（図１１参照）
◆ カタカナ列をキーワードとしたＫＷＩＣの表示
◆ カタカナ表記のゆれの検出
◆ 英字列をキーワードとしたＫＷＩＣの表示
◆ 英字表記のゆれの検出

(5) 注意を要する箇所の検出機能
◆ 指定した品詞のマーキング
◆ 受身の助動詞「れる」「られる」の検出
◆ 接続助詞の「が」の検出
◆ 指示語の検出
◆ 否定語の検出
◆ かな書き優先語の検出
◆ 数字列・特殊記号などの検出
◆ 曖昧になりやすい品詞の並びの検出

(6) 合成音声による読み合わせ機能

5　おわりに

以上、様々な文章解析応用を１つのシステム上に統

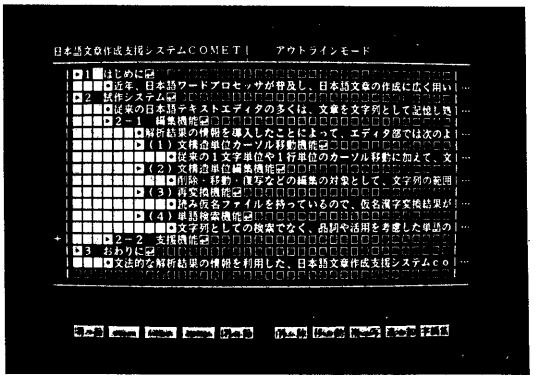

図９　アウトライン表示画面

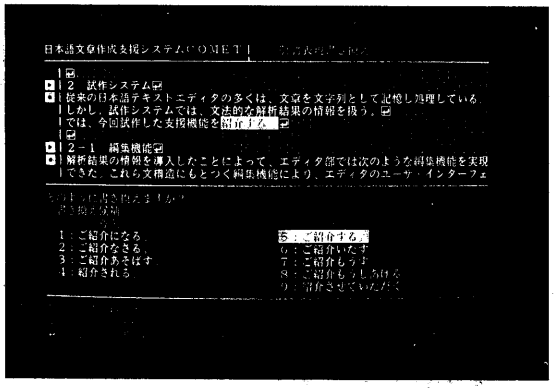

図１０　動詞の敬語表現への書き換え画面

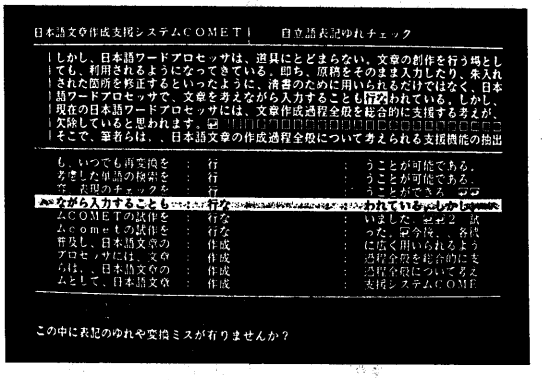

図１１　自立語表記のゆれ検出画面

合化するために、Ｓテキストを生成・更新する機構を導入することを提案した。このＳテキスト内には、文章が編集されても、常時、文章の階層構造や形態素情報などの文章情報が保持される。その結果、各種文章

解析応用は、個々に文章解析処理を行なう必要がなくなり、文章情報をもとにした処理を実行するだけでよいので、システム全体として、計算効率がよくなる。

日本語文章作成支援システムＣＯＭＥＴは、そのような方式にもとづいて構築されており、文章チェッカ、文音声変換、ＫＷＩＣ作成などの応用を中心として、様々な文章作成支援機能が、その上に実現されている。また、Ｓテキストによって、文字や行といった物理的な単位を対象とした編集処理でなく、単語・文節などの論理的な単位を対象とした編集が可能となっている。英語文では、単語が空白で区切られるのに対して、日本語文は、べた書きされる点が、従来、論理的な単位をもとにした編集にとって不利であったが、Ｓテキストは、それを解消する。

しかし、本方式にも、いくつかの課題・問題点が残されている。その第一は、文章の編集が行なわれるのに伴って文章解析処理（局所的再解析処理）が実行されるので、編集時の応答速度が、単なる文字列編集の場合に比べて遅くなることである。ＣＯＭＥＴの形態素解析モジュールの処理速度は２０文字／秒程度であり、編集時の再解析範囲は、通常、１０～２０文字であるから、その応答速度は１秒以内であるが、編集時の応答速度は高速であるほど望ましい。

第二の問題点は、文章構造を利用しない編集が頻繁に行なわれる場合には、計算効率は、必ずしもよいとは言えないことである。利用されない文章構造を生成するために、文章解析処理を行なうことになるからである。文章構造が必要となった時点ではじめて、壊れた箇所の再解析を行なう方が、計算効率はよい。ただし、その場合には、一回に再解析する箇所の数が増える（あるいは、再解析範囲が広くなる）ことになり、再解析処理の一回当りの処理時間が長くなる（編集のたびに再解析を行なう場合は、再解析処理の回数は増えるが、処理時間が分散されて、一回の再解析処理当りの処理時間は短くなる）。

第三に、計算機による文章解析処理では、解析誤りが避けられないことである。つまり、Ｓテキスト内の文章情報には誤りが含まれ、それが各種応用に好ましくない影響を及ぼす可能性がある。また、かな漢字変換モジュールと、形態素解析モジュールとでは、解析結果に相違が生ずる可能性もある。

今後、文章作成支援機能の拡張を行なってゆくだけでなく、上記の問題点・課題に対処するため、文章解析処理の高速化・高精度化も図ってゆく必要がある。なお、Ｓテキストは、ＣＯＭＥＴに閉じたものではなく、他のシステムでも利用できるような標準的な文章データであると考えている。

最後に、ＣＯＭＥＴの試作にご協力いただいた大竹嬢、柳生氏、鮫島嬢をはじめ、有益な助言を与えてくださった関係者各位に深謝する。

参考文献

[1] 長尾（監修）、“日本語情報処理”、電子通信学会、1984

[2] T.Johnson, “急速に立ち上がりつつある自然言語処理市場”、日経コンピュータ、1986.3.17号

[3] 福島・大竹・大山・首藤、“日本語文章作成支援システムＣＯＭＥＴ”、信学技報、OS86-21、1986

[4] 福島・大竹・大山、“日本語文章作成支援システムＣＯＭＥＴにおける文章書き換え機能”、情処34全大、6X-6、1987

[5] 福島・大山、“日本語文章作成支援システムＣＯＭＥＴ－文章構造処理－”、情処35全大、4S-3、1987

[6] 大山・福島・大竹、“日本語文章作成支援システムＣＯＭＥＴ－文章書換機能とそのユーザインタフェース－”、情処36全大、5U-7、1988

[7] 藤崎・諸橋、“「ことだま」の文書処理”、ｂｉｔ別冊「ワープロと日本語処理」、1985

[8] 武田・鈴木・藤崎、“日本語文書校正支援システムＣＲＩＴＡＣのユーザ・インタフェース”、信学技報、OS86-22、1986

[9] ISO DP 8613/2, "Information Processing - Text Preparation and Interchange - Text Structures Part 2: Office Document Architecture"

[10] L.D.Erman, F.Hays-Roth, V.R.Lesser and D.R.Reddy, "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty", ACM Compt. Surv., Vol.12, No.2, 1980

[11] 三留・伏木田、“ホルマント、ＣＶ－ＶＣ型規則合成”、音響学会音声研究会資料、S85-31、1985