自然言語処理 85−12
（1991. 9. 20）

自然言語処理を応用したマニュアル作成支援システム
−マニュアル推敲支援に関して−

納富 一宏† 内山 明彦††
†早稲田大学人間科学部 ††早稲田大学理工学部

概 要

自然言語処理の応用として，パッケージ・ソフトウェアのマニュアル作成支援を文書部品の結合によって実現するシステムについて考察した．文脈表現構造をシステムが直接扱うことで，文書作成を表層的な自然言語文と内部表現構造との相互変換として捉えることができる．そこで，処理対象を限定してシステムの基本動作を検討し，標題のシステムを試作した．本システムの推敲支援部では，文書の整合性評価に，文法，論理，文脈という段階を設け，評価結果からシステムが効率的な校正・推敲作業をプランニングするという方式を採用した．本方式は，意味レベルでの文書データベース構築，あるいは文書の部品化・再利用に対しても有効であると考えられる．

Manual Generating System by Using Natural Language Processing
-Manual Refinement and Proofreading-

Kazuhiro NOTOMI† Akihiko UCHIYAMA††
†School of Human Sciences ††School of Science and Engineering
Waseda University
2-579-15 Mikajima, Tokorozawa-shi 359, Japan

Abstract

In this article, a manual generating system is introduced. This system supports manual generation by connecting document modules. We have examined its fundamental characteristics and implemented the system with a limited processing range. The process may be considered as a mutual transformation of surface representation and deep data-structures, since context structures are directly managed by the system. At a refinement supporting part, it estimates data chains of syntax, semantics, and context levels. From the results of evaluation, it efficiently plans refinement and proofreading the manuals. Our determined methods are also usable for constructing document-database, or composing document modules.

1．はじめに

我々は，文書の記述内容を意味表現レベルで規定し，モジュール化することで，目的文書を部品から再構築するシステムについて研究を進めている．目的文書としては，計算機環境で利用されるアプリケーション・ソフトウェアのマニュアルに限定し，文書の意味や文脈までをおさえた作成支援環境の実現を目指している．

従来の文書作成支援方式では，支援対象となる文書の形態を限定せず，文書の校正・推敲を目的とするものが多く，表記・構文レベルの誤り検出や訂正に関する提案がなされている[1][2][3]．

しかしながら，文書の意味・文脈情報を利用した作成支援に関する提案[4]のうち，部品化・再利用を積極的に考察したものは少ない．

そこで，本稿では，マニュアル作成を「文書モジュール部品の合成」として捉えることで，部品品質の向上と文書合成の自動化という観点から，自然言語処理における文章理解と生成という２つのプロセスを応用する方法について述べる．

2．マニュアル文書の部品化・再利用

ソフトウェア工学からの要請はマニュアルの記述・編集・構成に対する作業全般の効率および信頼性の向上である．これを実現するための一つのアプローチとして，ソフトウェア部品のモジュール結合によるプログラム合成[5]の考え方をマニュアル作成に応用する．

2.1 モジュールとスケルトン

部品化・再利用を実現するための基底構造を，「モジュール(module)」，および「スケルトン(skeleton)」と呼ぶ．モジュールとは部品仕様により定義されるデータ表現であり，スケルトンとはモジュールを連結するアウトライン構造である．

これら２つのデータ表現により部品合成を考える場合，一般に，対象となる部品の内容をシステムが把握した上でモジュール化が行なわれる必要がある．また，モジュール化された部品をスケルトン構造内に配置する際には，部品結合および再配置に関する知識・ルールが必要となる．特に知識ベースシステムとして実現する場合，この点が重要である[6]．

マニュアル文書を部品化・再利用の対象とする場合，処理の入出力過程において扱うデータは自然言語表現である点に注目しなければならない．モジュール化を行なうためには，記述された文書の内容，即ち，文章表現構造をシステムが知る必要がある．これはモジュールが意味や文脈をおさえることのできるデータ構造によって定式化されなければならないことを意味する．

2.2 部品化・再利用プロセス

既存の文書部品データベースが存在する場合，目的文書の生成は，モジュールとして規定された意味表現構造のレベルで全ての処理が行なわれる．従って，新たに文書部品を登録したり，既存部品の更新をする場合は，全ての自然言語表現は，一度この意味表現構造へ変換されなければならない．この変換プロセスを「部品化」と呼ぶ．従って，部品化プロセスを自然言語処理の枠組みで捉えると，「文章の理解」というレベルでおさえることができる．

これに対して，モジュール部品を結合することにより得られる共通の内部表現構造は，最終的に目的文書として表層化されなければならない．このプロセスを「再利用」と呼ぶ．再利用プロセスには，①部品検索，②部品結合，③表層化という３つのフェーズが存在する．表層化フェーズは，自然言語処理の枠組みで捉えると，「文章の生成」というレベルでおさえることができる．

3．マニュアル作成支援

マニュアルの種類と要求される計算機支援の関係について考察する．また，文書部品を規定するデータ構造とその処理にも言及する．

3.1 ソフトウェア・マニュアル

マニュアルの形質的な類別から要求される項目には，以下に示す様な違いが認められる．

表１．マニュアルの種類による要求項目の違い

マニュアルの種類	分かりやすさ	正確さ	具体性	表現の自由度	網羅性
イントロダクション	◎	○	○	◎	△
チュートリアル	○	○	◎	○	○
リファレンス	△	◎	△	△	◎

※記号はそれぞれの項目につけられた点数であり，要求の度合いの目安である．
それぞれ（◎高い，○ふつう，△低い）を意味する．

ここで重要な点は，要求項目が異なれば支援方略も異なるということである．即ち，マニュアル作成におけるこれらの性質は表現される文章の内容を限定する（styling form が確定的である）ことを意味する．従って，自然言語処理的アプローチはマニュアルの種類に依存することになる．

表２．マニュアルの類別と特徴

◇リファレンス・マニュアル 最も計算機支援の実現が容易であり，アプリケーション操作における「辞書引き的使用」が前提となる．文脈情報はマニュアルの章・節・項の構成レベルで記述される．表現の自由度は低く，表現の質よりも索引・目次など情報検索の簡便さが要求される．
◇チュートリアル・マニュアル インテリジェント・チュータリング・システム(intelligent tutoring system)やＩＣＡＩ(IntelligentCAI)システムなど，他のアプローチによる計算機支援が期待されている．人間の問題解決の戦略を定式化した文脈情報が必要となる．
◇イントロダクション・マニュアル 最も表現構成の自由度が高く，かつエンドユーザにおいて最初に参照されるドキュメントである．そのため，ドキュメンテーションでの占める割合は小さなものではない．この意味で，自然言語処理による計算機支援のニーズが集中するべき箇所であると考えられる．反面，技術レベルでの文脈情報の記述が容易ではない．

計算機によるマニュアル作成支援では，先の要請に応えるべき戦略をマニュアル文書の部品化・再利用という形態で実現する（図１参照）．

3.2 文書部品のデータ構造

文書の部品化・再利用を行なうためには，モジュール(module)とスケルトン(skeleton)というデータ表現構造が必要であることは既に述べた．部品化・再利用の最小モジュール単位は，以下に述べる「パラグラフ(paragraph)」である．パラグラフはひとつの文脈を規定するプリミティブ(primitive)として表現される．複数のパラグラフを結合してモジュール化した部品を「複合部品」と呼ぶ．

3.2.1 パラグラフ表現構造

モジュールとして文書を規定する形態的な単位を「パラグラフ表現構造」と呼ぶ．パラグラフの構造定義の一例をあげると図２のようになる．パラグラフはトピックセンテンス(topic sentence)とボディセンテンス(body sentences)という単一意味表現のリスト構造として捉えられる．トピックセンテンス[7]はそのパラグラフ全体の文脈ヘッダー(スクリプト・ヘッダー[8])であり，実際の部品検索や部品結合時のインデックス(index)生成に利用される．

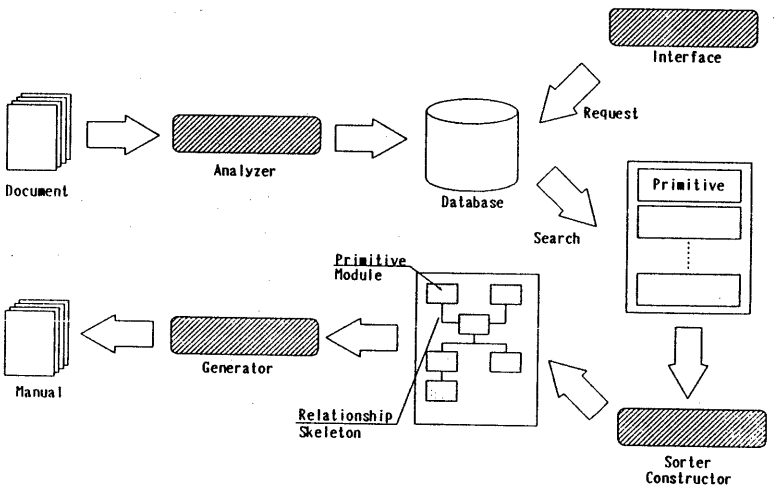

図１．部品化・再利用時のシステム構成

```
Paragraph ::=
   paragraph( TopicSentence, BodySentence, Link )
Link ::= link( Tag, Parents, Childlen )
TopicSentence ::=
   topic_sentence( CaseStructure, Topic )
BodySentence ::= body_sentence([ BodySet | _ ])
BodySet ::= body_set( CaseStructure, Number )
```

```
CaseStructure ::=
   case_structure( LinkageTag, DeepCase )
LinkageTag ::= linkage_tag( Upper, Lower )
DeepCase ::= deep_case([ CaseSet | _ ])
CaseSet ::=
   case_set( SlotName, SurfaceSlot, Level )
SurfaceSlot ::=
   surface_slot( Instance, SemanticMarker )
```

図2．データ構造の定義 I

3.2.2 部品インデックス生成

パラグラフは部品規定プリミティブである．パラグラフ表現構造のヘッダー情報から，部品検索や部品結合の際に参照されるキー(key)を抽出する作業を「インデックス生成」と呼ぶ．インデックスは，部品のカテゴリを示すラベルである．プリミティブの場合，ラベルは単一に指定される．複合部品の場合，各パラグラフからのラベルリストで表現される（図3参照）．

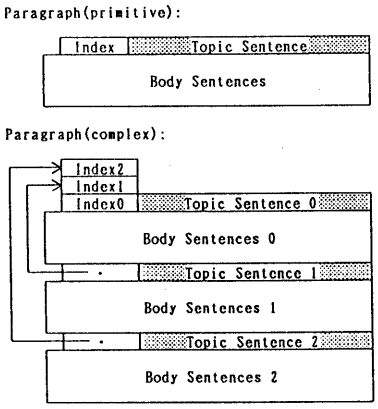

図3．パラグラフ形態とインデックス

インデックス・ラベルは，トピックセンテンスの意味表現に出現する格マーカと意味マーカの共起パターンに依存する．一例を下表に示す．

表3．インデックス・ラベル

Index Label	Co-Occurence Pattern of Case Marker & Meaning Marker
File_Access_Write	ACT(output), INS(file), OBJ(text), SRC(memory)
Print_Out	ACT(output), INS(printer), OBJ(data), SRC(file)
Text_Edit_Read	ACT(input), INS(soft), OBJ(string), SRC(file)

※ Data Structure ::= [CaseMarker(MeaningMarker) | _]

インデックス生成は，依存関係により完全に規定可能な場合のみ自動生成され，一意に決定できない場合には，ユーザとの対話的な操作により作成される．処理を下図に示す．

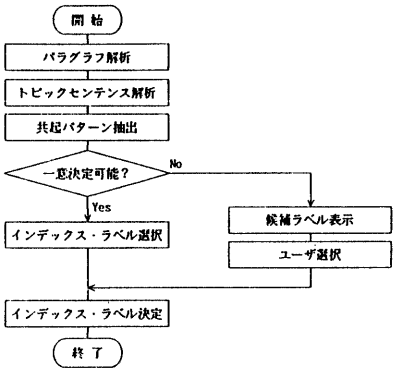

図4．インデックス生成

3.2.3 スケルトン表現構造

パラグラフによって規定されたモジュールを，論理構造に従って連結するためのデータ表現を「スケルトン表現構造」と呼ぶ．スケルトンは，表層化プロセスにおける目的文書の表現形態に対応し，文書構造を規定する．後で述べる編集支援形態のひな型に依存し，システムの提供するプランニング情報を有する．部品レベルでの一連の文脈を表現するものとして扱われる．スケルトンの構造定義の一例を下図に示す．

```
Skeleton ::= skeleton( Packets )
Packets ::= [Packet | _]
Packet ::=
   packet( module(Module), surface(Surface),
           constraint(Constraint), user(User) )
```

図5．データ構造の定義 II

3.3 部品結合方式によるマニュアル作成支援

部品結合方式を採用することによってマニュアル作成を捉えた場合，自然言語処理の手法を用い

る必要があることは既に述べた．マニュアル作成支援では，２つのプロセスが存在し，文書部品データベースへの登録と部品結合がその基本動作である．これを下図にまとめて示す．

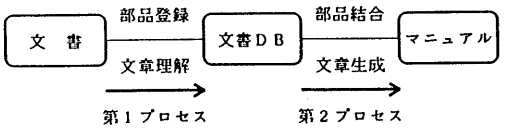

図６．２つの支援プロセスと自然言語処理

3.3.1　自然言語処理から見た部品登録

第１プロセスの部品登録では，自然言語で記述されたマニュアル文書を入力とする．入力された文書は形式パラグラフに切り分けられ，それぞれパラグラフ単位で解析処理に入る．解析処理では，文書のスタイルチェックを行ない，エラーが検出された場合にシステムは推敲支援を提供する．処理ループを抜けると検索情報を付加し，部品構造化を行なって文書部品化が終了する．これを図７に示す．

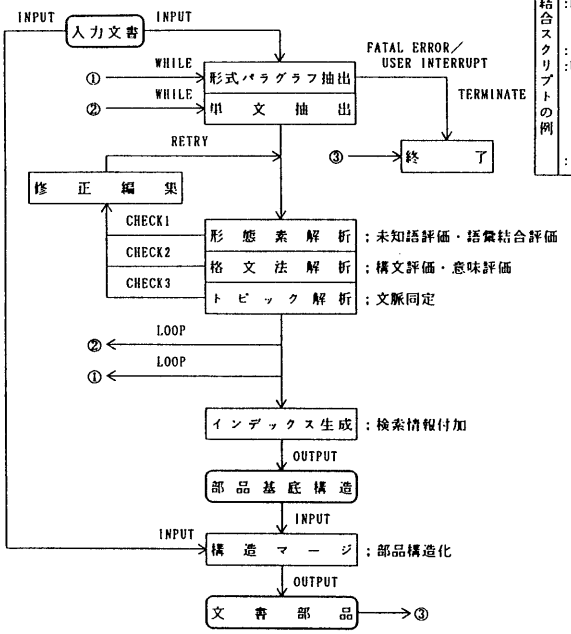

図７．部品化処理

ここで重要な点は，「部品カスタマイズ」である．部品カスタマイズとは，ユーザによる部品表現構造の直接的な修正作業のことであり，マニュアル推敲支援の中心をなす．これについては節をあらためて解説する．

3.3.2　部品検索

部品化プロセスにより構築される文書部品データベースを用いることで，マニュアル作成を行なうためには，最初にユーザによるマニュアルの部品結合仕様定義が必要である．この部品結合仕様書を「結合スクリプト」と呼ぶ．結合スクリプトはいくつかの記述フォーマットを持ち，一部制限された自然言語表現を扱うことができる．この結合スクリプトから，検索フレームが作成され，これに従って実際の検索が実行される．結合スクリプトの例を図８に，検索処理を図９に示す．

凡例（一部）

$部品名	部品名で示されるモジュール部品を検索・結合
#部品機能	部品機能で示される部品の指定
;コメント	行末までをコメントとする
〈アクション〉	結合処理の際のアクション（コマンド）
:ラベル	エンドマーク行までをラベルのスコープ範囲とする
::エンドマーク	ラベル行の終了
@前解読み込み	部品結合時に使用する情報の読み込み

結合スクリプトの例

```
;ワードプロセッサ「ＷＰ」のフロッピーディスクへのインストール方法
;
:head
   @<SPEC(WP)>                  ;ＷＰのスペック情報の読み込み
   @<INDEX(WP)>                 ;ＷＰの解説文インデックスの読み込み
::
:body
   #インストールの概要          ;機能「インストール」を検索
   #新規ディスクの準備          ;機能「新規ディスクの準備」を検索
   #システムディスクの作成      ;機能「システムディスクの作成」を検索
   #インストーラの起動          ;機能「インストーラの起動」を検索
   #ディスクの確認              ;機能「ディスクの確認」を検索
::
```

図８．結合スクリプトの例

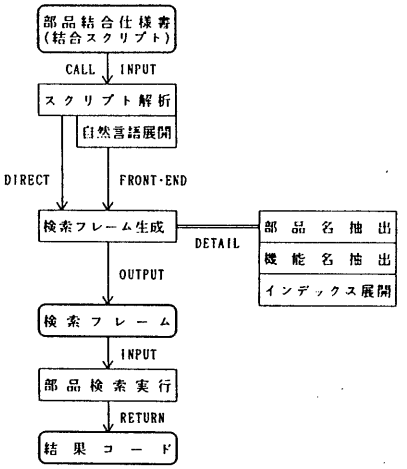

図９．部品検索処理

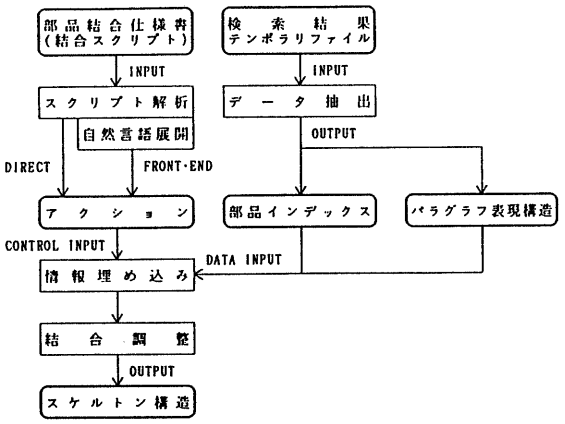

図10. 部品結合処理

3.3.3 部品結合

結合スクリプトに従って，部品検索を行ない，これを結合し，表現構造を得る操作が部品結合処理である．作成されるマニュアルは，部品モジュールがスケルトン構造に貼りついた形で規定される．実際には，表層表現のための調整が行なわれる．これを図１０に示す．

3.3.4 部品カスタマイズ

文書データベースに登録された既存部品を完全なブラックボックスとして扱うのではなく，新規部品作成のひな型として利用する．即ち，実際のマニュアル作成において，求めるモジュール部品が存在しない場合や，部品のバリエーションを追加する場合，形質の類似したモジュールを効率的に利用するために，ユーザが類似モジュールの書き換え作業を行なうことで，これに対処する．この編集操作全体を「部品カスタマイズ」と呼ぶ．

部品カスタマイズにおけるシステムの提供する編集支援項目には表４に示すものがある．

4．マニュアル推敲のシステム構成と全体処理

ここでは，文書部品結合によるマニュアル作成支援環境のうち，推敲支援部分のシステム構成，および処理の内容について述べる．

4.1 推敲システム概要

推敲支援では，マニュアル文書の評価を行ない，その評価結果に基づく編集のための知識を用いて

表４．編集支援のレベル

編集レベル	編集形態	概要
1	通常編集 ＋ 文書チェッカ	パラグラフ単位でのテキスト編集を行なう．機能および操作方法は通常のワードプロセッサに準ずる．この通常編集に続く処理で文書評価[1]を行なうのが「文書チェッカ」である．文書チェッカはデータベースへの部品登録時のインデックス生成を行なうほか，エラー検出時の編集レベルの選択・移行を受け持つ．部品登録時に自動起動される．
2	定型フォーム編集	既存部品をひな型として用いてた編集を行なう．書き換えにはレベルが設定されており，再編集範囲に制限を設けることができる．パラメータの埋め込みによる書き換え操作が中心となる．
3	アウトライン編集	パラグラフの構成編集を行なう．スケルトンの直接編集を目的とし，階層構造レベルで実行できる．スケルトンに貼り付けられたパラグラフ構造の操作にも対応する．
4	自動パラフレーズ	システムに与えられた知識を用いて，既存部品のデータ構造から，表層表現を生成する．ユーザが指定したパラメータに従って，①表現の詳細度の変更，②箇条書き文(タイトル)の生成，③常体・丁寧体の変更等を行なうことができる．

1) 文書評価項目を表５に示す．

表５．文書評価項目

評価レベル	概要
構文	形態素解析による語の連結を構文規則によりチェックする．
意味	格文法解析により述語の格フレームをチェックする．
文脈	論理構造と事象連鎖をチェックする．

プランニングを行なった後，文書編集を行なう．

評価としては，①表層整合チェック（文書の文法的な誤りの検出），②深層整合チェック（文書の意味的な誤りの検出）という２つの基本フェーズを想定している．

具体的には，最初に，システムはマニュアル文書から文脈表現構造[9][15]を抽出し，この構造の状態により編集操作を開始させる．この際に，システムは編集操作のプランニング情報から編集形態（styling form）を選択する．ユーザはシステムの提示したプランに沿った編集を行なう．

最も基本的な表現エラーチェックは，この文脈表現構造の抽出過程において評価可能である．また，使用される名詞概念，および動作・状態概念はシステムにアプリオリに与えられるものとする．マニュアル推敲支援という観点から眺めると，エラー検出に続くプランニングとそれに基づく修正段階が重要である．図１１にシステムの構成ブロック図を示す．

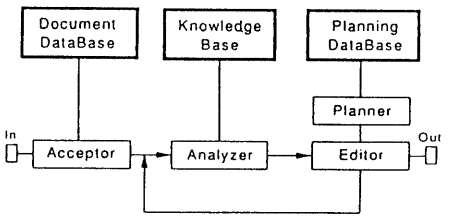

図１１．システム構成

4.2 推敲支援プランニングと制約

マニュアルという限定された対象を部品レベルで扱うため，個々のモジュール単位での構文・意味解析は小規模なものにとどめ，主として編集操作環境の起動性と文書評価・およびプランニングの安定性とを重視している．

先の文脈情報を利用して作成文書の表層的な整合性(文法・表記レベル)，および深層的な整合性(意味・論理レベル)をシステムが評価し，その結果によりプランニングを行なうが，プランニングでは，ユーザによる編集操作に対して制約条件を付加することが目的である．ユーザはこの制約の中で編集作業を行なうことができる．

文書作成における制約とは，「対象となる文章の文法構造・意味構造・文脈構造を(編集操作によって)破壊しないための条件や制限」として捉え，①システムの機能制限，②ユーザへの操作負荷（打鍵数など）の軽減，の２つの戦略として実現した．

5．文脈情報の利用

編集操作対象となる文書から，自然言語処理手法によって得られるデータ構造のうち，①文法構造，②意味構造，③文脈構造（論理連鎖，プランニング連鎖）を総括して「文脈情報(Contextual Data-Structures)」と呼ぶ．文脈情報とは，表現における論理構造の連鎖，もしくは一連の因果関係による構造を意味する．一つの状態と他の状態との関係記述である．マニュアルのパラグラフ単位の連続状態を区別すると，表層情報から得られる意味の連鎖として捉えることができる．

単文の規定には格文法による意味表現構造を利用する．実際には，パラグラフ表現構造，およびスケルトン表現構造を用いることは既に述べた（3.2節参照）．

パラグラフ間の関係記述およびその表示例を，それぞれ表６および図１２に示す．

表６．パラグラフ間の関係記述

関係種別	表　示	意　味
1) 連　接	(P1, P2, P3, ..., Pn)	順序依存関係
2) 並　立	[P1, P2, P3, ..., Pn]	可　換　関　係
3) 因　果	Pi->Pj	原　因　・　結　果
4) 暗　黙	!P	マ　ス　キ　ン　グ

例．(A, [B, C], D, !E, F->[G, H, I], J->K)

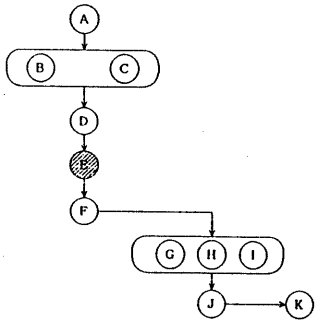

図12．パラグラフツリー

6．推敲支援の知識処理

6.1 文書推敲

推敲作業を開始するためには，対象となる文書の評価を定量的に行なうことが必要である．従来の研究ではマニュアル推敲支援を目的とした文書評価手法[10]，およびシステム[11][12]が提案されている．

これに対し，我々は文書部品という比較的小規模な対象を評価し，表現構造を完全に規定した上で推敲作業を行なうという立場をとる．即ち，文脈情報が存在する部品を対象とし，自然言語表現の練り直し作業を目的とするならば，これは表現構造からの文生成というレベルで捉えることが可能である．表現文の書き換えを，「文書パラフレーズ」と呼ぶ．我々は，技術文書など，文章の結合状態（因果関係）が明確な文章向けの文生成手法について検討してきた[13][14]．

6.2 推敲知識

マニュアル文書の記述内容の練り直しをするのに必要な知識を「推敲知識」と呼ぶ．マニュアルに要求されることは，その表現記述の正確さとわかりやすさである．従って，これらに関するヒューリスティックスをルールベースとして統合管理し利用できる環境を実現することが必要である．

推敲作業とは，新たな文章表現を次々に得ることで，その中から最適な表現を選択することを意味すると考えることができる．

推敲知識は以下の書式により表現される．

<No>	<目的>	<制約>	<アクション>

図13．推敲知識の表現書式

<NO>とはその知識の識別番号を示し，<目的>とは評価文をどのように書き換えるかその種類を示す．また，<制約>とは書き換えのできる範囲や程度，自由度に対する制限などを示し，<アクション>とは自動的に書き換えを行なうバッチ命令を示す．

ここで，アクションには以下のようなものを用意している．表7．パラフレーズ・アクションの分類

アクションの種類	概　説
① modifier制御関係	連体・連用修飾語句の表示・非表示や折込文にするなどを制御する
② 態 変 換 関 係	態変換に関して文全体を書き換える
③ 同義語変換関係	用語・術語の統一や連続文での文末の重なりなどを書き換える
④ 接続語制御関係	連続文の因果関係により接続語句の書き換えを制御する
⑤ 格 制 御 関 係	格の交換や表示・非表示を制御する

パラフレーズ・アクションにより，いくつかの書き換え作業が選択される．ユーザは，選択されたそれぞれの編集環境において，レベル設定とパラメータ指定を行なうことで，新規表現を生成させることができる．従って，この作業の繰り返しにより，推敲作業を行なうことができる．

7．動作例

文節移動による表層書換え，および格フレームによる不定表層格の追加編集の動作例を示す．

表現交換機能（パラフレーズ機能）：エディタ動作	
システムディスクを反転初期化します．	<ESC><M>
反転初期化 システムディスクを作成します．	<←>
システムディスクの作成を反転初期化行ないます．	<→><→>
システムディスクの作成を最初に行ないます．▌	<ESC>
注目点は反転表示で示される．注目点をカーソルキーで移動できる．この際構文チェック機能が働き，ひな型に従って文が自動的に書き変わる．	

図14-(a)．動作例 I

共起パターンチェック機能	
メモリチェックを反転初期化．	<ESC><C>
反転初期化 ▌	<↑><↑><↑>
反転初期化 _____	
反転初期化 _____	
メモリチェックを反転初期化．	
反転初期化 システム起動時に	<システム起動時に><RET>
反転初期化 _____	<RET>
反転初期化 デバイスの初期化要請により	<デバイスの‥より><RET>
メモリチェックを反転初期化．	
システム起動時にデバイスの初期化要請により メモリチェックを行なう．▌	<ESC>
注目述語は反転で示される．注目述語の格文法・共起パターンから省略項目を検出して入力状態となる．カーソルを項目に移し必要な部分を記述する．記述された新たな情報により文を自動的に書き換える．指定された述語の共起パターンに省略がない場合は，格分解された形で修正可能状態となる．	

図14-(b)．動作例 II

8．おわりに

自然言語処理の応用システムとして，パッケージ・ソフトウェアのマニュアル作成支援を考察した．文脈表現構造をシステムが直接扱うことで，文書作成を表層的な自然言語文とデータ構造の間での相互変換として捉えることができる．

本システムの推敲支援では，①文書の文法的整合性評価，②文書の論理的整合性評価，③文書の文脈的抽象性評価という段階を設け，これによりシステムがプランニングした推敲仕様に従って，ユーザに編集作業を効率的に行なわせる手法について述べた．

アプリケーション・ソフトウェアの機能や操作方法の統一化が図られた場合，外部仕様の共有というレベルでドキュメンテーションを考えることができる．このことから，本システムは，限定された対象を扱う場合に対して，意味レベルでの文書データベースの構築，あるいは文書の部品化・再利用を積極的に実施する際に有効である．但し，実支援環境レベルでは，知識ベースの管理・保守が重要となる．現在，以下の問題点に関して検討を進めている．

①文書パラフレーズの意味レベル制御の安定化
②表層化処理の表現における「なめらかさ」の定量化
③推敲支援のプランニング規則および制約条件記述の効率化

【参考文献】

[1]野村直之，村木一至：言語の構造単位を保持した文書執筆支援システム，情処研報，Vol.91，No.25，82-10（1991）．
[2]堀川博史，伊藤俊之，高野彰：ソフトウェア文書のためのエディターSpec，情処学論，Vol.31，No.3，pp.390-396（1990）．
[3]香沼明，倉田昌典，牛島和夫：日本語文章推敲支援ツール『推敲』における否定表現の抽出法，情処学論，Vol.31，No.6，pp.792-800（1990）．
[4]清水富門，他：意味情報を用いた校正支援，第37回情処全大 6B-7，pp.1018-1019（1988）．
[5]小高知宏：ソフトウェア部品化・再利用向きプログラム言語環境について，信学論，Vol.J73-D-I，No.1，pp.18-27（1990）．
[6]坂本寧 編：ソフトウェア工学ハンドブック，オーム社，pp.223-259（1986）．
[7]木下是雄：理科系の作文技術，中公新書624，pp.58-74（1981）．
[8]田中幸吉，淵一博 監訳：人工知能ハンドブック第I巻，共立出版，pp.389-401（1983）．
[9]納富一宏，内山明彦：自然言語における定型文の解析，早大情科センター紀要，Vol.9，pp.44-46（1989）．
[10]高橋善文，牛島和夫：計算機マニュアルの分かりやすさの定量的評価方法，情処学論，Vol.32，No.4，pp.460-469（1991）．
[11]高橋善文，吉田哲三：計算機マニュアル推敲・査読支援システムMAPLEの開発と運用，情処学論，Vol.31，No.7，pp.1051-1062（1990）．
[12]林良彦，菊井玄一郎：日本文推敲支援システムにおける書換え支援機能の実現方式，情処学論，Vol.32，No.8，pp.962-970（1991）．
[13]納富一宏，内山明彦：文脈処理としての日本語文生成－科学技術論文作成支援システム開発への試み－，第41回情処全大（1990）．
[14]納富一宏，小高知宏，内山明彦：自然言語によるプログラムドキュメント自動生成への一考察，第35回情処全大，3X-1，pp.997-998（1987）．
[15]納富一宏，内山明彦：文脈理解への一考察－内部検証空間を用いた日本語非定型構文の文脈解析－，情処研報，自然言語処理72-5（1989）．