情　報　学　基　礎　32－8
（１９９３．　１１．　９）

歴史系テキストデータへの リバースエンジニアリング応用の基礎的研究

藤田茂†、菅原研次†、伊與田光宏†、八重樫純樹††

†千葉工業大学情報工学科
〒275　習志野市津田沼2-17-1

††国立歴史民俗博物館
〒285　佐倉市城内町117

あらまし
フルテキストデータベースは、歴史研究における文献、資料管理などを支援するための有効な情報処理システムであるが、その構築にあたって次の二つの問題を解決する必要がある。一つはフルテキスト内の情報を効率よく検索するために、ハイパーテキスト構造に変換することであり、もう一つはフルテキストやそれに含まれているテキストモジュールを検索するための情報を抽出することである。膨大なフルテキスト集合に対して、この作業を行うことは、データベース構築者にとって大きな負担を伴なう作業であり、これに対する計算機支援が必要である。本研究ではリバースエンジニアリングの方法論に基づいて、フルテキストからハイパーテキストへの変換と、検索情報の抽出を支援する枠組みを提案する。

和文キーワード　歴史研究支援、フルテキストデータベース、ハイパーテキスト、検索情報抽出

Application of Reverse Engineering to Design of Full-text Database for Supporting Historical Research

Shigeru Fujita†, Kenji Sugawara†,Mitsuhiro Iyoda†, Junki Yaegashi††

†Dept. of Computer Science, Chiba Institute of Technology
2-17-1, Tsudanuma, Narashino, Japan 275
††National Museum of Japanese History
117, Jonai-chou, Sakura, Japan 285

Abstract
Full-text database is an effective tool to support historical research utilizing various kinds of texts. The following two problems should be solved when the system is designed. One is to transform a full-text into a hyper-text in order to access an objective information included in it. Another is to extract indexes for retrieve of full-texts and their hyper-text structure from original full-texts.
This task is a heavy work for a designer of full-text database. In this paper, a framework of supporting system for transformation of a full-text into a hyper-text and for extraction of indexes from it, based on the methodology of the reverse engineering.

英文key words　Historical Research Support, Full-Text Database, Hyper-text, Index Extraction

1. はじめに

歴史系研究を支援する方法論のひとつとしてデータベースを中心とした情報システムを構築し、歴史研究者が行う資料管理や資料分析の作業を計算機で支援する方法があげられる。歴史系研究に関係する資料としては、古文書、古記録、絵画史料、埋蔵物などの一次資料と、一次資料を基礎として得られた研究成果、文献などの二次資料がある。一次資料、二次資料を通じて文字列形式の資料は多くの部分を占め、従って文字列形式の資料の管理と利用とを支援することは歴史系研究支援にとって有効である。このような研究アプローチとして、例えば続日本紀のフルテキストデータベースを作成し分析処理を行った事例がみられる[星野92]。

フルテキストデータベースの管理で問題になるのは検索情報の生成と、利用時の問題である。検索情報の抽出については、フルテキストデータベースの構築者、管理者あるいはテキスト作成者の作業になっているが、これはかなり労力を費やす作業であり、自動化技術が望まれている[諸橋84]。

一方利用時の問題とは、検索されたテキストから利用者が必要とする情報のみを見つけ出す方法である。一般にテキストは大量の文字列の集合であり、利用者は検索されたテキストの中から必要な部分を特定する作業を行わなくてはならない。あるいはいくつかのテキストが例えば引用関係などで相互に関連しており、その関係を追跡しなければならない場合もあるであろう。これも利用者に負担を与える作業であり、従ってテキストの検索のみならず、検索したテキストを利用する作業も計算機で支援する必要がある。このためにはテキストにハイパーテキスト構造を導入する必要がある[小川92]。

上記のいずれの問題をも解決しようとすると、蓄積された大量のテキストを分析し、テキスト全体の検索情報、あるいはテキストの内部構造とその検索と利用に関する情報を抽出しなければならない。これは基本的には文字列の集合から上記の問題に関連する意味を抽出する問題であり、現在の情報処理技術を持ってしても、最も困難な問題の一つである。

一方、計算機のプログラムもテキストと同様に文字列であり、これの理解問題や再利用問題は従来からソフトウェア工学の分野で研究されてきた。この自動化もなかなか実用化に至る解決は見られないが、近年知識工学の導入によりプログラムの意味理解を指向したリバースエンジニアリングという技術が注目されている[Har90]。

本研究では、歴史研究を支援するための歴史資料を管理するフルテキストデータベースの構築を目的としている。フルテキストはテキスト内部の情報検索が可能となるように、ハイパーテキスト化する必要がある。しかしながら大量のフルテキストに対してハイパーテキストに変換する作業は、データベース構築者にとって大きな負担のかかる作業である。本稿ではフルテキストのハイパー構造とその検索情報の抽出作業を支援するための技術に、リバースエンジニアリングの技術を適用することの有効性を検討することを目的としている。

2. 歴史研究支援のフルテキストデータベース

2.1 歴史研究支援システムの概要

さまざまな歴史研究に使われる古文書や古記録などの資料管理がデータベースを利用して行われている[永村９１]。例えば東京大学資料編纂研究所の史料データベースに見られるように、歴史資料は(1)原史料、(2)画像データ形式で光ディスクなどの様々な記憶媒体に蓄積される史料画像データベースと、テキスト形式のファイルにされたフルテキストデータベース(3)一次資料検索のためのインデクス情報のデータベース(4)資料群管理のデータベースの４層の階層構造のデータベース群で管理される[永村91]。

図１は一般的な歴史資料データベースシステムを利用した歴史系研究の支援システムを文字系資料を中心として見た概念図を示す。支援システムは４層に分かれる。第１階層は古文書、古記録、埋蔵物、絵画、民俗資料などからなる原資料階層である。原資料の研究成果である文献そのものも第１階層の資料と考える。第２階層は古文書や文献などの文字系資料をテキスト形式のデータに変換したフルテキストデータと埋蔵物、絵画など非文字系資料の情報を記述したテキスト形式の２次資料からなるテキスト階層である。第３階層は原資料階層とテキスト階層の資料の管理を行うための管理情報階層であり、資料検索のためのインデクス情報や資料の様々な属性や特性を整理した表形式データから構成される。第４階層はこのシステムを利用する研究者が必要とする各種の歴史資料・情報から研究者の観点から抽象化・整理された情報の階層である。この階層はまた研究のバックデータとなるさまざまな統計データ、分析図表から構成される。第４階層では統計支援、地図情報など様々な情報処理機能が利用される。また原資料の画像管理システムや、画像処理システム、科学分析などの資料分析支援システムも必要である。また高度なヒューマンインタフェース機能も歴史研究者の支援には必須の機能である[菅原91]。

本研究ではこの支援システムのなかのテキスト管理・利用の支援システムを構築することを目的としている。

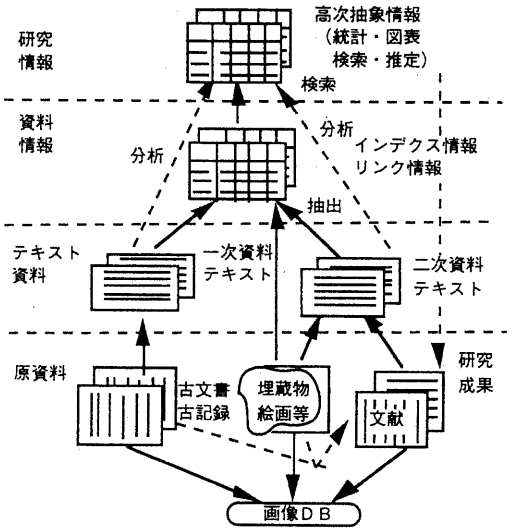

図１　歴史系研究支援システムの概念図

2.2 歴史系研究支援のためのフルテキストデータベース構築の課題

2.2.1 検索情報の自動抽出

大量のフルテキスト集合に対して適性な索引を付けることは、実は極めて熟練的知識が必要な作業であり、かつ手間が大変かかる作業である。この問題に対して自動索引付けの研究が行われてきた。自動索引付けは文献などのフルテキストを分析して、検索のために有効な索引を機械的に抽出することが目的であり、索引は少なくとも次ぎの二つの特性を備えている必要がある[諸橋84]。

(a) 文献あるいは文献集合の内容を的確に表現していること（表現力）

(b) 文献あるいは文献集合間の識別が充分に行えること（識別力）。

索引の抽出は多くの手法では索引語辞書が用意されこれと一致した語がテキスト中に見つかったときそれを索引とする方法である。ところが領域によって索引とすることが不適切な場合もあり、これはある領域に対して索引として不適切な集合（不用語集合）としてシステム管理者が指定することにより索引の抽出を制御している。あるいは辞書を効率よく管理するため形態素解析の知識や文脈理解の知識も導入されている。

一方語の出現頻度による索引付け手法も研究されている。これは索引語の出現パタンにはある領域固有の傾向があると言う前提にたち、その出現頻度や分布の仕方を分析することにより適正な索引を推定する方法である[長尾76]。

これらの自動索引付け技術は、理論的研究が主体であり、有効性の面からは文献領域の固有の経験的知識を利用した手法が実用的であると言われている。

2.2.2 歴史系研究支援フルテキストデータベースの利用法の課題

フルテキストデータベースにおけるもう一つの問題点は、テキスト利用における利用法の支援である。利用者が目的とするテキストを利用したいと考えた場合、そのテキストフルテキストを常に全て利用することはあまり考えられない。多くの場合テキストが持っている情報のある部分に書かれている情報のみを参照したいとか、あるキーワードに関係した記述がある部分のみを参照したいという利用法が多いであろう。フルテキストデータはその性質上一つ一つが大容量データであり、これを最初から最後までを一通り眺めないと利用できないのであれば、いくら検索の能力が高くても、利用支援にはならない。この問題の解決法としては、テキストの構造化を行ったり、それらの構造間の関係を管理するいわゆるハイパーテキスト化が必要になってくる。

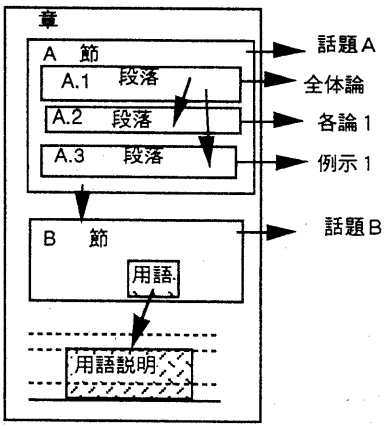

図２　フルテキストにおける
ハイパーリレーション

フルテキストをハイパーテキスト化しておく長所として次ぎのことも考えられる。　一般に資料や文献のテキストデータは共用性が高く、それ自体は独立のデータ集合として組織されており、種々のデータベースから利用されることが想定される。この場合、それぞれのデータベースの利用者の視点が異なるため、そのテキストの中で参照したい部分が異なると思われる。このときテキストが論理的な意味が割り当てられていて、ある検索インデクスにより検索可能な段落の形式で構造化されていれば、一つの文献を様々な用途に利用でき、データベースの共用性も向上すると思われる。従ってフルテキストを図２のような論理的意味単位に

分解してその意味単位の間の関係を見つけ出し、管理することは歴史系研究を支援するフルテキストデータベースにおいて重要な問題である。

以上より歴史系研究支援のフルテキストデータベースの課題として

(1) テキストの構造化
(2) テキストモジュール間の関係の抽出
(3) テキストモジュール検索情報の抽出

が考えられる。(1)と (2)の問題は歴史資料のフルテキストを研究領域知識に依存したハイパーテキストに構成しなおす問題である。(3)の問題は2.2.2.で述べた手法だけではなく、さらに階層構造や関係情報の抽出のため意味理解にむけてもう一歩踏み込んだ処理が必要になる。

3. 歴史系研究支援のフルテキストデータベース構築へのアプローチ

3.1 リバースエンジニアリングの方法論

ソフトウェア開発の効率を上げるために、従来より多くの方法が提案されてきた。リバースエンジニアリングもそのなかの一つであり特に最近脚光を浴びだしたものの一つである。リバースエンジニアリングとは既開発のソフトウェアのプログラムから仕様情報を再生成する技術で、普通のエンジニアリングが仕様書からプログラムが作成される過程とは逆の方向に処理が進むのでリバースという形容詞がついている[Chiko90]。

リバースエンジニアリングが実用的技術に近付いてきたことの理由の一つに、知識工学の導入があげられている。リバースエンジニアリングとは言ってみれば、プログラム分析者がプログラムを認識、理解する作業を支援する技術である。従来は、プログラマやシステムエンジニアと呼ばれる熟練技術者が、マシンコードと呼ばれるメモリの２値情報から、あるいはソースコードと呼ばれるFORTRAN,COBOL,C言語で書かれているプログラムから、それを作成した開発者の意図や思考、行動の過程を読取っていた。人間の分析技術者がこれを行うためには計算機ハードウェア、プログラム言語、処理アルゴリズムなどや対象分野の知識など（例えば財務処理、科学技術計算、歴史学など）、色々な分野における極めて高度で専門的な知識が要求され、かつ膨大な量のプログラムを処理、分析しなければならないので多大な労力と時間が費やされていた。このプログラム理解の過程を機械で支援するためには、熟練技術者の知識をシステムに取込み、これを積極的に利用しながら作業を進めなければならず、従って知識処理の枠組みが稼働することが必要であった。

プログラマがプログラムを理解するプロセスを調べると、一般に言えることは

(1) プログラムの中から論理的に意味を持つ最小のかたまりを見つけ出す。これをモジュールという。モジュールを見つけ出すためには、プログラム言語の文法に関する知識が必要である。（構造の識別）
(2) そのモジュールの機能や属性を見つけ出す。（構造の意味理解）
(3) いくつかのモジュールを組み合せて、更に大きいモジュール（上位モジュール）を見つけ出す。このレベルのモジュールを見つけ出すにはソフトウェア構造の知識や様々な設計対象領域の知識が必要になる。（モジュール間構造の識別）
(4) 上位モジュールの機能や属性を見つけ出す。（構造の意味理解）
(5) (3) と (4) を繰り返す。

以上のプロセスにより、文字列としてのプログラムからモジュール構造と機能の認識を行うことができる。

3.2ハイパーテキスト化へのリバースエンジニアリング技術の応用

ソフトウェア工学におけるリバースエンジニアリングの対象であるプログラムソースコードも文献テキストも文字列のファイルである点で類似している。違いは、プログラムが記述している文法規則が機械言語であり、記述内容がアルゴリズムや情報変換などであるのに対して、文献テキストは自然言語により記述され、記述内容が事実、仮説、命題、論証、思想、主張などの論理的意味であることである。これは大きな違いであり、記述メディアが同じだからといって同じ分析手法を適用することは乱暴な話であるが、文献テキストの検索情報を自動抽出する前処理としては有効な一次情報が抽出できることが期待できる。

リバースエンジニアリングにおけるプログラムコードの分析プロセスと、本研究のアプローチである文献などのフルテキストの分析プロセスの対比を図３に示す。分析対象はプログラムも文献テキストも、文字列から構成されるテキストファイルである。

リバースエンジニアリングでは、テキストファイルを構造化するため対象プログラムを記述しているプログラミング言語の文法規則を用いて、関数や手続きや変数などの宣言など意味を形成する可能性のある構成要素を識別する。一般にプログラムの構成要素は階層的構造を取り、変数の参照関係を調べることにより構成要素間の相互関係を解析することができる。構造識別のプロセスの次は仕様抽出のプロセスである。これは識別した構成要素（プログラムモジュール）の階層をボトムアップに解析し、プログラム全体の仕様を再構成していく。最初の段階では他の構成要素を含まない一番小さな構成要素の機能仕様、動作仕様、デー

タ受け渡し仕様などを抽出し、これらの組み合せで上位の構成要素の同様の仕様を抽出する。この仕様抽出にはプログラミングの知識やそのプログラムが実現している応用領域（例えば行列計算）の知識が使わる。応用領域の知識は、機能を理解し再利用情報を生成するうえで重要である。

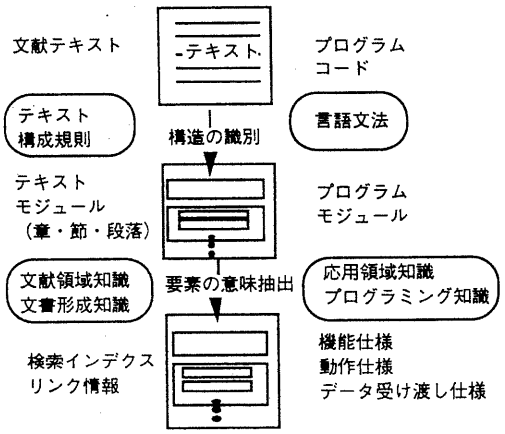

図３　プログラムコード分析とフルテキスト分析

これに対応する文献テキストの分析プロセスを図３の左側に示す。文献テキストの構造を識別するための知識としてテキスト構成規則を用いる。テキスト構成規則とは、「章」、「節」、「段落」などのテキストの構成要素の名前である。これによりテキストの構造を決定する。次のプロセスは識別された構成要素の検索情報と構成要素間の関係情報を抽出するプロセスである。このとき構成要素間の論理関係を調べる手がかりとしてプログラミング知識に対応する作文などの文書形成知識を利用する。文書形成知識とは論理の展開に用いられる三段論法や起承転結といた段落間（段落集合間）の関係の知識である。接続詞などを手がかりにしてこれらのパターンを発見することによりテキストの構成要素間の関係を推定する。各構成要素の情報インデクスを見つける従来の手法は索引辞書を用意して一致する用語をインデクスとする方法であり一般に有効である。しかしながら用語の重要性や意味は解析対象の文献テキストが所属する研究領域により違ってくる。従って各構成要素あるいはテキスト全体の検索情報として引き出すべき検索インデクスは対象とする研究領域の知識に依存しなくてはならない。あるいはその領域の用語の意味（用語間の関係）や、出現頻度、出現規則などを手がかりに、検索インデクスや構成要素の関係情報をより的確に推定できる可能性がある。すなわち文献テキストの分析にはその研究領域の領域知識を利用することが重要であり、本研究では考古学研究、特に土偶の研究[八重樫92]に領域を絞って本アプローチの有効性を検証することを目的としている。

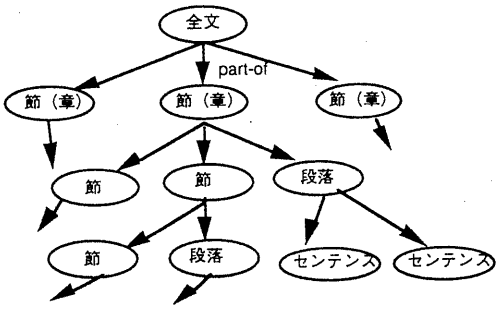

図４　フルテキストの構造木(part-of relation)

4. フルテキストの構造記述

テキストは一次元の文字列の集合である。文字要素は、ひらがな、カタカナ、漢字、数字、アルファベット、特殊文字からなる。改行、空白などは特殊文字に含めることにする。

テキストを分析するために色々な手法が提案されている。例えば分かち書きにする処理を行ったり、分析者の判断で適当な文字列単位で区切り記号を入れる[永村91]、などの前処理が考えられる。これらの処理は文字列の集合にしかすぎないテキストファイルになんらかの構造を定義することを狙いとしている。本稿でもフルテキストに対して図２に示すような構造を導入する。本アプローチでは自然言語理解の技術を現時点では採用せず、フルテキストの最小構成要素をセンテンスとして、その内部の形態素解析などは行わない。センテンスは解析のキーワード集合を含む文字列として定義する。

＜フルテキスト＞は＜節テキスト＞の順序付き集合である。いわゆるテキストの章や節の構造は本定義では全て＜節テキスト＞で抽象化している。すなわち＜節テキスト＞は＜節テキスト＞か＜段落テキスト＞の順序付き集合で表わされる。＜フルテキスト＞、＜節テキスト＞、＜段落テキスト＞の包含関係(part-of 関係)を図４の木構造で表わす。

構成要素の形式的定義を図５に示す。＜フルテキスト＞はフルテキストデータベースで管理され、＜フルテキストＩＤ＞（識別子）が付けられている。さらに＜フルテキスト題目＞やフルテキストを管理する上で必要なそのテキストの分類コードなどの情報が記述されている＜フルテキスト属性＞が付けられる。＜フルテキスト関係＞は、図６に示されるようなテキスト間の関係や、検索情報

や画像ファイルへのポインタなどが書かれている。＜フルテキスト＞の内部構造は＜フルテキスト実体＞で記述される。＜フルテキスト実体＞は＜節テキスト＞の順序付きの集合で表わされる。

＜節テキスト＞はフルデータベースで一意に定められている＜節ＩＤ＞、テキスト内で定められている＜節番号＞、

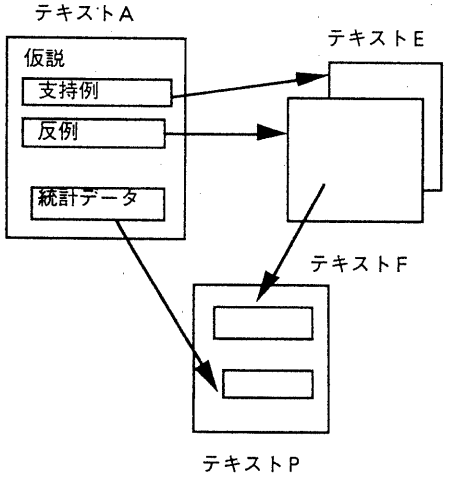

図6　テキスト間のハイパーリレーション

節や段落間の包含関係や意味関係を表わす＜節関係＞、その節の特徴、分類などを記述する＜節属性＞及び＜節テキスト実体＞の記述からなる。＜節テキスト実体＞はその節テキストの内部構造を表わし、＜テキスト実体＞かあるいは＜節テキスト＞の順序付き集合で表わされる。＜テキスト実体＞は＜フルテキスト＞に含まれる実際のセンテンスの集合を表わしこれが検索情報抽出のための解析の対象になる。＜テキスト実体＞は＜段落テキスト＞の順序付き集合である。＜段落テキスト＞はフルテキストの中で一意な＜段落テキストＩＤ＞とそれが含まれる節のなかに占める順番である＜段落番号＞、段落テキストが含まれる＜節テキスト＞へのポインタや他の段落との関係を表わすリンクが記述されている＜段落関係＞、その段落の特性を記述する＜段落属性＞および＜段落テキスト実体＞から構成される。＜段落テキスト実体＞は＜センテンス＞の順序付き集合である。＜センテンス＞は、＜フルテキスト＞のなかで一意な＜センテンスＩＤ＞、段落テキストとの包含関係やセンテンス間の意味関係を記述した＜センテンス関係＞、＜センテンス属性＞および＜センテンス文字列＞からなる。

センテンスは自然言語（日本語）の文章であり、何らかの意味を構成する。フルテキストの意味解析を行うためには、センテンスのレベルの意味解析を基本として、段落テキスト、節テキストと高次の意味集合へと抽象化していく必要がある。しかしながら自然言語理解の研究は、いまだ実用化に至らず現在さまざまな観点から研究が進められている状況である。

本稿ではセンテンスの自然言語理論に基づく意味理解は行わず、センテンスに文字列として包含される用語を手がかりとして、経験的知識に基づいてフルテキストに含まれる上記の構成要素の意味を推定し、フルテキストデータベースを利用する上で必要とする検索情報、利用情報を抽出するアプローチをとる。

```
＜フルテキスト＞=＜＜フルテキストＩＤ＞、＜フルテキスト題目＞、
                    ＜フルテキスト属性＞、＜フルテキスト関係＞、
                    ＜フルテキスト実体＞＞
＜フルテキスト実体＞＝ {＜節テキスト＞}
＜節テキスト＞=＜＜節ＩＤ＞、＜節番号＞、＜節タイトル＞、
                ＜節属性＞、＜節関係＞、＜節テキスト実体＞＞
＜節テキスト実体＞=＜＜テキスト実体＞、＜節テキスト＞*＞
＜テキスト実体＞＝ {＜段落テキスト＞}

＜段落テキスト＞=＜＜段落テキストＩＤ＞、＜段落番号＞、＜段落属性＞、
                    ＜段落関係＞、＜段落テキスト実体＞＞
＜段落テキスト実体＞＝ {＜センテンス＞}
＜センテンス＞=＜＜センテンスＩＤ＞、＜センテンス属性＞、
                    ＜センテンス関係＞、＜センテンス文字列＞、
                    ＜アンカー＞＞
＜アンカー＞=＜＜アンカーＩＤ＞、＜アンカー文字列＞、＜アンカー属性＞
                ＜リンクオブジェクトID＞、＜リンク属性＞＞
＜リンクオブジェクトＩＤ＞=＜＜フルテキストID＞or＜節テキストID＞or
                ＜段落テキストID＞ or＜ センテンスID＞ or＜ アンカーID＞ ＞
*空集合を含む
```

図5　フルテキストの構成要素の定義

5. フルテキストの属性記述

5.1 構造木の記述

文字列のファイルであるフルテキストから、図5のテキストの構造の定義に基づいて、図4のような構造木を生成する作業の自動化は比較的簡単である。改行や空白などの制御文字や、「章」や「節」のタイトルの形式を手がかりとし、テキストを＜節テキスト＞、＜段落テキスト＞に分解する。

現在実験環境として,UNIXワークステーションでプログラミング言語Smalltalkを利用して試作システムを作成している[菅原93]。段落テキスト実

体はSmalltalkオブジェクトとして扱われ、段落テキスト実体内部の構成要素はSmalltalkオブジェクトとして管理されている。

システムは構造木を定義するにあたって、分解基本アルゴリズムに従って図5の構造木を定義するが、最終的な判定は人間が行うこととする。その結果はグラフ形式でウィンドウに表示される。各テキストファイルに対して一つの構造情報ファイルが生成される。土偶の研究領域の研究論文[稲野92]の構造分析を行った結果の表示ウインドウの例を図7に示す。

5.2 テキストモジュールの属性記述

フルテキストの属性を記述するファイルであるフルテキスト属性スクリプトは以下のスクリプトから構成される。

(A)フルテキスト属性
　分類（古文書、古記録、研究文献等）
　シソーラス分類、検索キーワード、
　メトリクス等、参照テキスト情報、
(B)節テキスト属性
　文構成分類（章、節、序論、結論、理論、資料等）シソーラス分類、検索キーワード、
　節メトリクス等
(C)段落テキスト属性
　検索キーワード、段落メトリクス等
(D)リンク属性
　リンク分類、リンクキーワード等

メトリクスとはテキスト（モジュール）の特徴を表わす、サイズ、用語頻度などの統計量であり、そのフルテキストが属する領域に依存したある意味を持つ。

スクリプトはフレームモデルで記述されており、抽象情報を記述するクラススクリプトと、具体的なフルテキストの特性を記述するインスタンススクリプトから構成される。

6. 属性スクリプト抽出支援プロセス

フルテキストから属性スクリプトを抽出する作業を支援するプロセスの概要を図8に示す。プロセスの目的はフルテキストの検索のためのハイパーテキスト情報を抽出することである。プロセスは図8では楕円で表わされ、処理の順番がふられている。

作業の最初は構造スクリプトを抽出するプロセスである。これは図7に示すように比較的自動化が簡単なプロセスである。この構造を基に、各テキストモジュールのメトリクスをボトムアップに計測する。計測するメトリクスは領域知識に依存し、基本データである領域辞書などや統計情報などは動的に変える。

得られたメトリクスをもとに、フルテキストや章などの大きなテキストモジュールの所属領域（ドメイン）の推定を行う。これをもとにメトリクスパターンライブラリを類似度に基づいて検索し、対応する属性スクリプトのテンプレートを検索する。各テキストモジュールの属性スクリプト

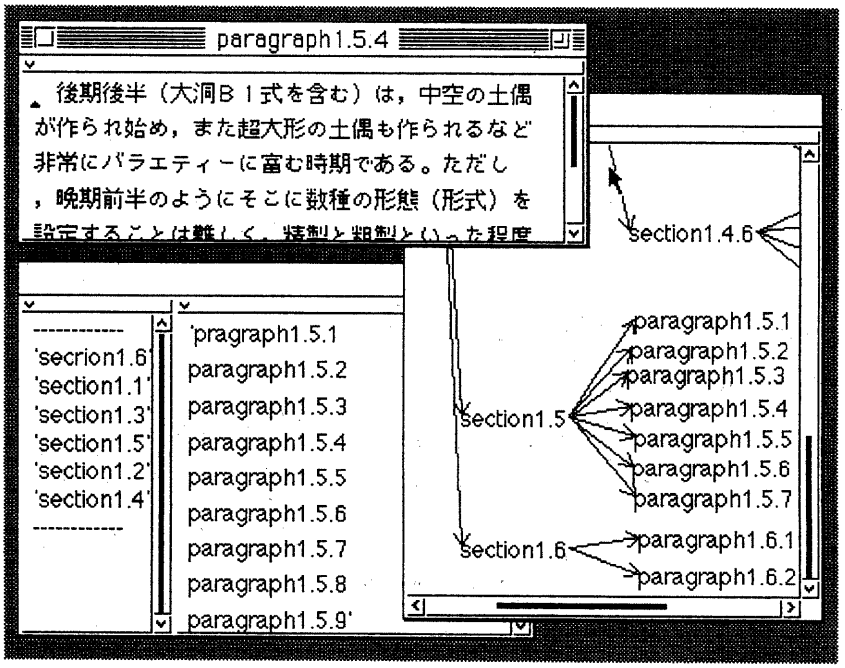

図7　構造スクリプトの表示ウィンドウ

は、構造スクリプトや領域知識の情報を用いて、フルテキスト属性スクリプトの可能な仮説を生成する。この仮説は検証プロセスにより仮説の妥当性の検証がおこなわれ、検証できない場合はドメインを変更するなどしてあらたな仮説を生成する。

各プロセスはシステムとフルテキストデータベース構築者の協調作業により実現される。現在は分析知識の獲得が行われていないため多くの作業が人間の負担になるが、このプロセスを運用していく過程でその知識を獲得し蓄積していくことにより、機械により支援可能な作業の割合をふやしていく。

7. まとめ

歴史研究を支援するための歴史資料を管理するフルテキストデータベースの構築するうえで必要な、フルテキストをハイパーテキストに変換する作業を支援する枠組みの検討を行った。テキスト構造の定義とその構成要素となるテキストモジュールの属性を抽出するための分析支援システムをSmalltalkを用いて開発している。今後は、領域知識を利用することによりフルテキストの分析と情報抽出の支援の能力を向上させることを目指している。

参考文献

[長尾76]長尾、水谷、池田、" 日本語文献における重要後の自動抽出" 、情報処理, Vol.17,No.2,1976

[諸橋84]諸橋、" 自動索引付け研究の動向" 情報処理、vol.25,no.9,1984

[稲野92]稲野、金子、熊谷、中村、" 岩手県の土偶　ー縄文時代後・晩期を中心にー、" 国立歴史民俗博物館研究報告、第３７集、１９９２

[小川92]小川、菊地、高橋、" フルテキスト・データベースの技術動向、" 情報処理、vol.33,no.4, 1992

[永村91]永村、" 日本史史料データベースとデータ処理に関する研究、" 国立歴史民俗博物館研究報告、第３０集、１９９１

[星野92]星野、" 日本史データベース、" 情報処理、vol.33,no.18, 1992

[八重樫92]八重樫、小林、" 土偶資料を例とした資料情報化研究（１）ーコンセプトと研究経緯、その課題ー、" 国立歴史民俗博物館研究報告、第３７集、１９９２

[菅原91]菅原、伊與田、福島、八重樫、" パーソナルコンピュータを用いた静止画像データベースの歴史資料管理への応用例" 、国立歴史民俗博物館研究報告、第２９集、１９９１

[菅原93]菅原、伊與田、八重樫" 全文テキストデータベースとリバースエンジニアリング" 、国立歴史民俗博物館研究報告、第５３集、１９９３

[Chiko90] E.J. Chikofsky and J.H. Grosss Jr.,"Reverse Engineering and Design Recovery: A Taxonomy," IEEE Software, vol7. no.1, Jan. 1990

[Har90] M.T. Harandi and J.Q. Ning,"Knowledge-Based Program Analysis," IEEE Software,Vol.7, No.1, pp.74-81, Jan. 1990

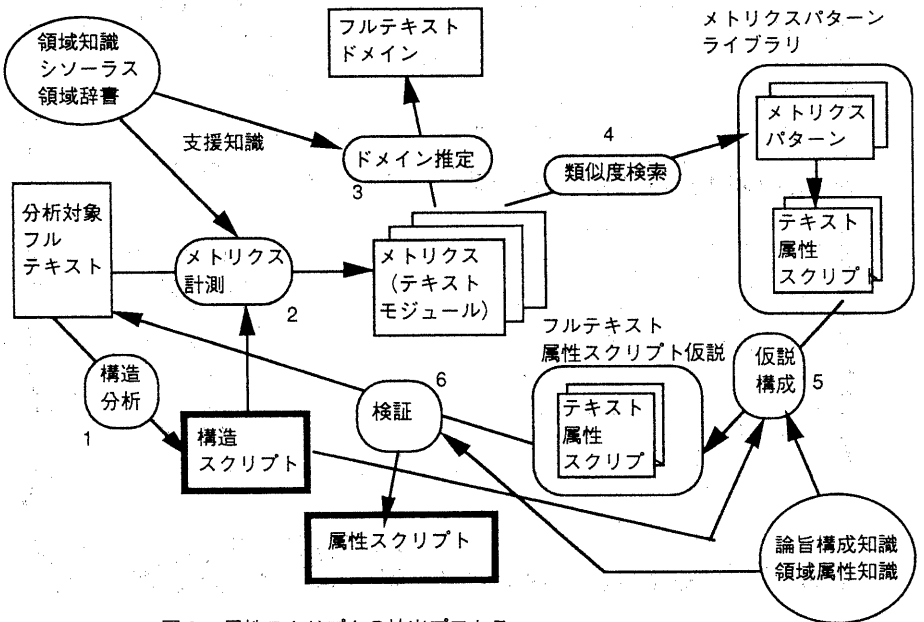

図８　属性スクリプトの抽出プロセス