データベースシステム 104－4
（1995. 7. 18）

# 文書データベース管理システムXebecの検索について

中津山 恒
hisashi@rsl.crl.fujixerox.co.jp

沼田 賢一
numata@rsl.crl.fujixerox.co.jp

富士ゼロックス(株)システム・コミュニケーション研究所
〒259-01 神奈川県足柄上郡中井町境430 グリーンテクなかい

本稿では，文書データベース管理システムXebecにおける文書検索の機能と検索方式の概略について述べる．Xebecにおける文書検索の特徴は，構造化文書の文脈情報に基づく検索である．Xebecのデータモデルでは，文書の内容は構造化され，有向順序木で表わされる．構造化された文書の内容を論理構造という．論理構造中のノードに関する情報と，ノード間の接続関係を組合せた情報を文脈情報と呼ぶ．文脈情報に基づいて検索条件を指定することにより，文書スキーマを特定用途に偏向させずに検索機能を高めることができる．

# Document Retrieval in the Xebec Document Database Management System

Hisashi Nakatsuyama　　Kenichi Numata

Systems and Communications Lab., Fuji Xerox Co.,Ltd.
430 Sakai, Nakai-machi, Ashigarakami-gun, Kanagawa, 259-01 Japan

This paper describes document retrieval in the Xebec document database management system. The most prominent feature of document retrieval in Xebec is to utilize "contextual information" of structured documents. In the Xebec data model, contents of documents are represented as ordered trees called logical structures. Contextual information is the mixture of information within nodes and structural relationship between nodes in logical structures. Specifying queries based on contextual information, we can describe precise conditions, while keeping document schemata generic: schemata need not to be changed for specific purposes.

# 1 はじめに

ワードプロセッサやデスクトップパブリッシングシステムの普及により, 効率的に文書が作成できるようになった. 文書が電子化されたため, 文書データを「切り貼り」でき, 定型的な文書の作成がとくに効率化された. 電子化された文書データは, 二次記憶に保管できる. また, オンラインまたはオフラインで簡単に交換できる.

しかし, 文書が再利用できるのは, 使っているシステムがその文書データを解釈できる場合に限られる. 再利用したい文書があっても, システムがその文書を解釈できなければ, 再利用するどころか表示さえできない. したがって, その文書の内容を再度打ち込んだり, コンバータでデータを変換したり, ハードコピーを文字認識で読み取らせるなどの作業が必要である.

このような問題を避けるひとつのアプローチは, オフィスで文書作成に用いるシステムまたは作成した文書のデータを統一することである. しかし, そのオフィスの外, たとえば取り引き先で用いる文書データ/システムまで統一する訳にはいかないので, 一般にはオフィスで扱う文書データを統一できない.

もうひとつの問題は, 膨大な文書の中からいかにして所望の文書を探し出すかである. 目的の文書を保管してあることが分かっていても, 探し出すことができなければ, それはないに等しい.

これらの問題を解決するため, 異種混合文書データベースを管理できる文書データベース管理システムが待ち望まれる.

我々は, このような問題意識から, 異種混合文書データベースの管理機能と, 構造化文書がもつ情報を複合的に利用した高度な検索機能をもつ文書データベース管理システム Xebec を作成した.

本稿では, 最初のプロトタイプである Xebec V1.0 の文書検索機能と検索処理の概要について述べる.

# 2 Xebec の概要

Xebec V1.0(以下 Xebec と略す)[10] の主な特徴を列挙する:

- 異種混合データベースの管理.
- 高度な検索機能.
- グラフィカルユーザインタフェース.
- サーバ/クライアントアーキテクチャ.

以下, これらの特徴について順次説明する.

文書を表現する情報構造のことを文書アーキテクチャという. 文書の生成規則を文書クラスという. 文書クラスは, 文書の雛型であるとも言える. 国際規格である ODA[3] は文書アーキテクチャであり, ODA の共通論理構造は文書クラスである. 文書アーキテクチャによっては, 明示的な文書クラスが存在しない. このような文書アーキテクチャでは, 唯一の文書クラスが定められていると考えることができる.

異種混合データベースの管理では, 文書アーキテクチャ/文書クラスの差異をいかに吸収するかが肝要である. Xebec は, これらの差異を吸収するため, 特定の文書アーキテクチャに依存しない, 独自のデータモデルを有している. データベース中の文書は, 文書の雛型である文書スキーマにしたがって生成される.

文書の格納時には, 対応づけと呼ぶ処理によって, 格納される文書の論理構造 (内容) を再構成する [12]. 対応づけで得られた論理構造は, もとの文書データと対でデータベースに格納される. このように論理構造を二重化したことにより, 異種混合データベースの管理を実現している.

対応づけにより, 異なる文書アーキテクチャ/文書クラスの文書群を同一のスキーマのインスタンスとして扱うことができる.

検索条件を指定するためには, データベース中の表現についてのみ考慮すればよく, もとの文書アーキテクチャ/文書クラスが何であったかを気にする必要はない.[1]

[1]検索条件には, もとの文書データの文書アーキテクチャ/文書クラスに関する条件を指定することもできるが, これらの条件の指定は他の条件の指定とは完全に独立である.

検索結果は, ユーザが指定した文書アーキテクチャ/文書クラスの文書としてデータベースから取り出すことができる. 指定された文書アーキテクチャ/文書クラスがもとの文書データのものと異なる場合には, 対応づけが行なわれる.

管理対象は, Akane[7], CALS[5], RTF[4] の文書である. CALS は, SGML[6] に基づく文書アーキテクチャである. CALS の枠組みで, 文書クラスを定義できる. 一方, Akane と RTF では, 文書クラスを新たに定義することはできない.

対応づけ処理はシステムに組み込みではなく, 対応づけ規則を評価することで行なっている. 対応づけ規則は格納用と取り出し用を別に定義する. 格納用の対応づけ規則は, 入力文書が属する文書クラスの情報と, 出力文書が属すべき文書スキーマの情報, および文書がもっている情報をもとに記述する. 取り出し用の対応づけ規則は, 入力文書が属する文書スキーマ情報と, 出力文書が属すべき文書クラスの情報, および文書がもっている情報をもとに記述する.

Xebec は, 文書に付与された属性に加え, 論理構造の要素のタイプ, 要素がもつ内容, 要素に付与された属性, 要素間の関係を複合的に利用した, 高度な検索機能を提供する.[10]

Xebec にはテクスチュアルなデータ定義言語やデータ操作言語はなく, グラフィカルユーザインタフェースによって文書スキーマの定義や検索式の指定を行なう.[8]

サーバ/クライアントアーキテクチャの採用により, 複数のプラットフォームで Xebec のクライアントを作成することができる.

サーバは, Xebec のアプリケーションインタフェースを提供するカーネルと, 永続オブジェクトの管理を行なうストレージシステムとからなる. ストレージシステムには, 汎用のオジェクト指向データベース管理システム ObjectStore を用いている.

# 3 データモデル

本節では, Xebec データモデルの文書スキーマと文書インスタンスについて, 検索機能の理解に必要な部分を中心に説明する.

## 3.1 文書スキーマ

文書スキーマは, スキーマ名, 管理属性, 意味記述, タイプ定義の 4 つ組で定義される.

スキーマ名は一意的であって, これによってデータベース中の文書スキーマを指定できる.

タイプ定義は, この文書スキーマから生成し得る文書インスタンスの構造を規定するもので, タイプ名, 意味記述, 属性定義, 内容定義, 構造定義からなる.

構造定義は, タイプ定義と構造生成子による有向グラフで表現される. 構造生成子には以下の 5 つがある.

**AGG**
: 下位の要素が, 任意の順序でちょうど 1 度出現する.

**SEQ**
: 下位の要素が, 指定された順序でちょうど 1 度出現する.

**OPT**
: 下位の要素が高々1 度出現してよい.

**CHO**
: 下位の要素のうち, 1 つだけ出現する.

**REP**
: 下位の要素が, 1 度以上出現する.

内容定義では, そのタイプ定義にしたがって生成されるノードがもち得る内容体系が定義される. 内容体系には, テキスト, 幾何学図形, イメージ, テーブル, 数式がある. リーフとして出現するノードは必ず内容をもつが, リーフでないノードは内容をもたなくてもよい. リーフでないノードがもち得る内容はテキストだけである.

文書インスタンスの解釈が曖昧にならないよう[2], また有意な単位で文書内容の一部を抽出できるようにするために, 構造定義には一定の制約を設けている. Xebec では, この制約を満たさない文書スキーマは定義できない.

属性の型は, 文字列型, 整数型, 日時型, 人型,

[2] 文書スキーマを構文規則とみたとき, 文書インスタンスの導出木が一意的になるようにする.

列挙型がある．人型は，名前，地位，所属からなる．列挙型の場合は，項目を併せて定義する．

図1に，文書スキーマの表示例を示す．矩形で示されているのがタイプ，楕円で示されているのが構造生成子である．この文書スキーマ Article は，章立てをもつ文書構造を定義している．この定義では，章 (Section) が再帰的な定義になっており，文書インスタンスは任意の深さの章をもつことができる．

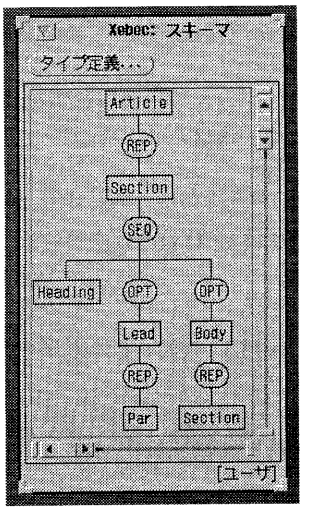


図 1: スキーマの表示例

### 3.2 文書インスタンス

文書インスタンスは，管理属性と論理構造からなる．

管理属性は，著者や作成日時などの書誌的な情報を表現するものである．

論理構造は，有効順序木で表現される．論理構造のノードにはタイプがあり，文書スキーマのタイプ定義に応じて属性や内容をもつことができる．前述のように，リーフノードは必ず内容をもつが，リーフでないノードは内容をもつとは限らない．

テキスト内容は，テキスト本体，体裁情報，下位構造のアンカーとからなる．テキスト中に下位構造が出現するのは，そのテキスト内容をもつノードのタイプ定義が構造定義をもつ場合である．

テキスト以外の内容はもとの文書データで表現される．このため，テキスト以外の内容は内容検索の対象外となる．

図2に，文書インスタンスの論理構造の表示例を示す．この文書インスタンスは，図1の文書スキーマ，Article にしたがって作成されたものである．

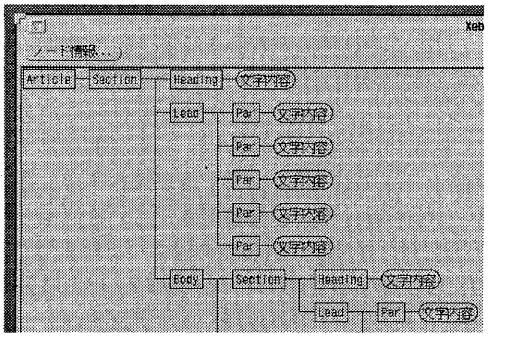


図 2: 文書インスタンスの論理構造の表示例

## 4 文書の検索

### 4.1 機能設定の方針

以下の方針により，検索機能を設定した．

1. 論理構造に基づく検索機能を提供する．
2. 汎用的な検索機能を提供する．
3. コンベンショナルな文書管理システムの検索機能を含む．

ここで，方針 2について説明する．

システムによっては，「ある条件を満すノードの 3 番目の弟」，「ある条件を満すノードの 3 番目の子の 2 番目の子」といった，操作対象の文書に極度に依存した条件指定を許している．しかし，これらの条件指定をするには，操作に先立って操作対象の文書の性質を詳細に知っていなければ記述のしようがない．

このような操作対象の文書に極度に依存した条件は，汎用の文書データベース管理システムでは記述する必要がないと考えた．

もしそのような処理を望むなら, 汎用的に記述できる範囲で文書または文書部品 (論理構造の部分木) を検索し, 検索結果を詳細に解析した上で, 紫 [11] のような文書処理システムで処理すればよい.

## 4.2 検索機能

Xebec V1 における文書の検索には, 管理属性, スキーマ, 論理構造に関する条件を指定することができる. 論理構造に関する条件を指定するときには, 必ず文書スキーマを指定しなければならない.

管理属性に関する条件を問合せツール [8] で表示した例を, 図 3に示す.

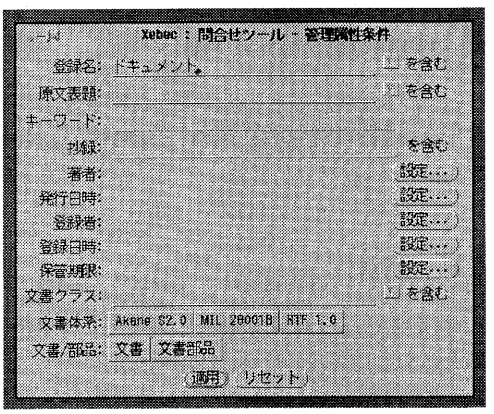


図 3: 問合せツールで, 管理属性に関する条件を表示した例

このユーザインタフェースでは, 属性ごとに検索条件を指定できる. フィールドが空欄になっていれば, その属性の条件は指定されなかったものとして扱われる. 複数のフィールドに条件が指定されれば, それらは連言的であると解釈される. すなわち, 検索結果はすべての条件を満す文書である.

論理構造に基づく検索条件には, タイプ・属性・テキスト内容といったノードに関する条件と, 親子関係や祖孫関係といったノード間の関係とを指定できる. 前者を局所条件, 後者を接続条件と呼び, これらを組合せた条件を文脈条件と呼ぶ.[3]

文脈条件による検索式の例を図 4に示す. この検索式は, 章見出しに「文書」という文字列を含む章で, 図見出しに「DBMS」という文字列を含む図をもつものを検索する.

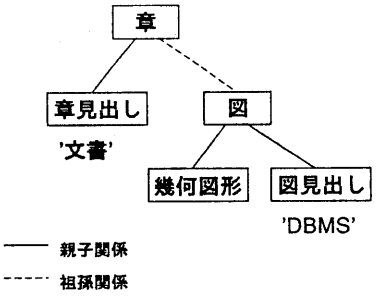


図 4: 文脈条件による検索式の例

文脈条件は, 文書スキーマを特定の処理に偏向させることなく, 高度な検索機能を提供するという特長がある.[9]

ユーザインタフェース上は, 接続条件と局所条件は別々のウィンドウで指定する. 図 5に, 問合せエディタで論理構造に関する条件を表示した例を示す.

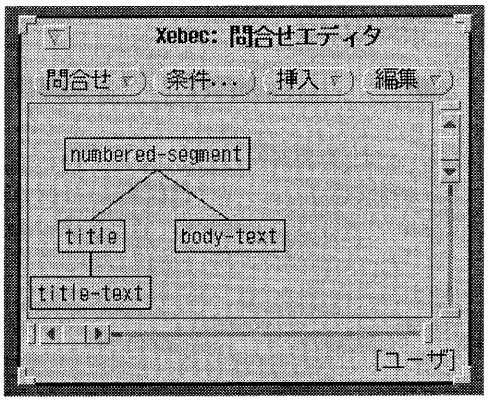


図 5: 論理構造に関する条件を表示した例

図 6に, ノード条件指定ウィンドウでノードに関する条件を表示した例を示す. 図 6では, こ

[3] ノードに関する情報とノード間の接続関係を組合せた情報を文脈情報と呼ぶ.

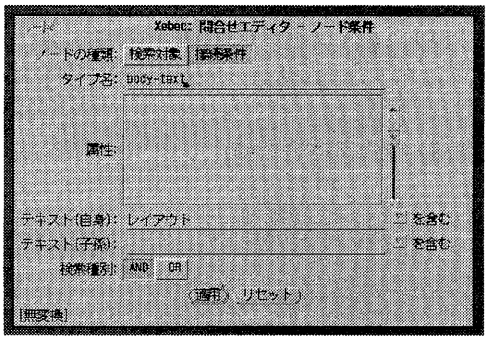


図 6: ノードに関する条件を表示した例

のノード自身がもつテキスト内容に関する条件が指定されている.

管理属性に関する条件と同様, 局所条件と接続条件は連言的である. すなわち, すべての条件を満たす対象だけが検索結果となる.

検索対象を指定しないときには, 検索結果は検索式のルートで指定されたノードである. 検索対象を指定すれば, ルート以外のノードを結果として得ることができる. 図 5で "body-text" を取り出したければ, 図 6のスイッチ "ノードの種類" を "検索対象" にする.

# 5 評価方式

本節では, 検索式の評価方式について概説する.

Xebec は, 管理属性の索引, ノードのタイプおよび属性の索引, ノードの祖孫関係 (先祖と子孫の関係) の索引をもっている. 祖孫関係の索引は, マルチインデックス [2] を応用したものである. それ以外の索引には, ObjectStore の索引を利用している.

テキスト内容については索引をもたず, 篠原・有川法 [13] にもとづくアルゴリズムでフルテキスト検索を行なう. 篠原・有川法は, Aho-Corasick 法 [1] を多バイト文字列に適用するアルゴリズムである. これらのアルゴリズムは, 複数のキーワード検索を同時に行うことができ, 照合のコストがキーワードの数によらず一定であるという特長をもつ.

## 5.1 検索式の検証

指定された検索式の条件を満たす文書またはノードが存在し得るかどうか検証する. 検証には, 以下の検査が含まれる.

1. 統語論的に正しいか
2. 意味論的に誤っていないか

意味論的な検証では, 検索式を文書スキーマの定義と照合し, 以下の検査を行なう.

1. 指定された文書スキーマが存在するか
2. 指定されたタイプが存在するか
3. 指定された, 要素間の親子関係/祖孫関係を満たすものが存在し得るか
4. 指定された属性が存在するか
5. 指定された内容体系は合っているか
6. 属性の値域が合っているか
7. 属性値の範囲指定は妥当か

これらの検査により検索条件を満す文書が存在し得ないと判定された場合, その検索式は評価されない.

## 5.2 検索

指定された検索式の条件を満たす文書または文書部品を検索する. 文書の識別子の集合が指定された場合, それらの中で, 指定された条件を満すものを検索する.

検索条件には, 評価コストの安価なものと高価なものがある. 安価なものは索引機能を利用できる条件で, 高価なものは索引が利用できない条件, すなわちフルテキスト検索を行なう必要のある条件である.

検索式の評価戦略を定めるにあたり, 以下のことを考慮した

- 属性値の評価には, 索引が利用できる.
- テキスト内容検索には索引を利用せず, フルテキスト検索を行なう.
- 接続条件の評価には索引が利用できる.
- 検索式は連言的である.
- 接続条件は, ノードの存在に関する条件である.

以上を踏まえれば, 以下のように処理できることが分かる.

1. いずれかの条件を満たさなくなった文書またはノードは, 評価結果が偽となる. 評価が偽になった時点で, 候補から除外する.
2. 接続条件を満すノードが少なくとも 1 つもつことが分かった文書またはノードについては, その接続条件を満たす他のノードは探さない.

検索式の評価の概要を以下に示す.

1. クライアントから渡された検索式を内部表現に変換する.
2. 検索式を検証する.
3. 変換して得られた検索式を, 評価コストの安い順に並べ換える.
4. 管理属性に関する条件のうち, 評価コストの安いものを評価し, 条件を満たす可能性のある文書集合を求める.
5. 指定された文書スキーマの文書インスタンスを求める。
6. 候補となっている文書を読み出す権利があるかどうか検査する.
7. 管理属性に関する条件のうち, 評価コストの高いものを評価する.
8. 論理構造に関する条件を表現する木をたどりながら, テキスト以外の局所条件, 接続条件を評価する.
9. 検索対象の文書部品をたどりながら, テキスト内容に関する条件を評価する.

# 6 検索実行例

本節では, 検索の実行例を示す.

**検索条件**

文書に関する条件は, 「図 7に示す文書スキーマの文書インスタンスのうち, 登録名に『ドキュメント』を含む文書」である.(図3参照.) 論理構造に関する条件は, 「見出し段落に『ドキュメント』を含み, 直下の段落に『レイアウト』を含む節」である.(図 5参照.)

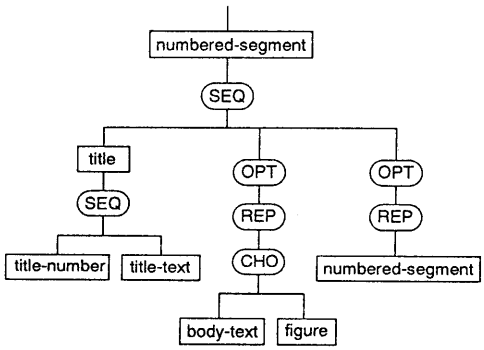


図 7: 検索対象の文書スキーマ (検索に関わる部分を抜粋)

**データベース**

格納されている文書の数は 325. もとの文書データは, Akane S2.1 を使って書かれたものである.

**平均検索時間**

6 度検索し, 2 度め以降の時間を平均したところ, クライアントから検索式を受け取って結果を送信し終えるまでの時間は 18.4 秒であった.

**環境**

ObjectStore のサーバと Xebec サーバは同一ホスト (SPARCstation 10) で, クライアントはサーバとは別ホスト (SPARCstation 5) で実行した. オペレーティングシステムはいずれも Solaris 2.3 である.

現在のシステムは最初のプロトタイプであり, 性能解析や検索アルゴリズムのチューンは行なっていないため, 性能は決して十分なものではない.

# 7 おわりに

本稿では, Xebec V1.0 の検索の機能および評価方式の概略を述べた.

現在, Xebec V1.0 を機能・性能の両面から評価中である. この評価を踏まえ, Xebec V1.0 を改良していく予定である.

## 8 謝辞

富士ゼロックス(株)システム・コミュニケーション研究所の奥村洋所員と内田剛所員は, 著者とともに検索機能について検討した. 安松一樹所員, 安藤俊明所員, 門馬敦仁所員は, 著者ともに研究を進めており, 日頃から有益な議論をしていただいている. 上林憲行主幹研究員, 小部正人副主任研究員には, 研究遂行のため格段の配慮をいただいている. この場をお借りして, 感謝の意を表する.

## 参考文献

[1] Alfred V. Aho and Margaret J. Corasick. Efficient string matching: An aid to bibliographic search. *Communications of the ACM*, Vol. 18, No. 6, pp. 333–340, June 1975.

[2] Chris Clifton and Hector Garcia-Molina. Indexing in a hypertext database. In *Proceedings of the Sixteenth International Conference on Very Large Data Bases*, pp. 36–49, 1990.

[3] International Standardization Organisation. *Information Processing – Text and Office Systems – Office Document Architecture (ODA) and Interchange Format. ISO 8613*, 1989.

[4] Microsoft. *Rich Text Format. Microsoft Word Technical Reference. Chapter 10.*

[5] US Department of Defence. Military standard: Automated interchange of technical information (MIL-STD-1840), 1987.

[6] International Standardization Organisation. Information processing systems – text and office systems – standard generalized markup language (SGML). ISO 8879, 1986.

[7] 屋内恭輔, 保科孝之, 田口安男, 小林晴法, 西田賢一, 黒澤宏. 構造化ドキュメントエディタ Akane. 富士ゼロックス テクニカルレポート, pp. 98–105, 1993.

[8] 沼田賢一, 奥村洋, 千葉和也. 文書データベース管理システム Xebec のユーザインターフェースについて. データベースシステム・データ工学合同研究会. 情報処理学会, 電子情報通信学会, July 1995.

[9] 中津山恒, 楠本浩二, 村田真. 構造化文書の文脈情報に基づく文書操作システム. In *92-TCG-5-7.* 情報処理学会, January 1993.

[10] 中津山恒, 京嶋仁樹, 奥村洋, 安松一樹, 安藤俊明, 内田剛, 千葉和也, 沼田賢一, 上林憲行. 文書データベース管理システム Xebec の概要. In *95-DBS-101.* 情報処理学会, January 1995.

[11] 楠本浩二, 黒澤宏, 鈴木克明. 構造化文書を対象とした操作コマンド 紫. In *95-DBS-101.* 情報処理学会, January 1995.

[12] 京嶋仁樹, 安松一樹. 文書データベース管理システム Xebec の論理構造変換方式. OFS94-51～55. 電子情報通信学会, March 1995. OFS94-53.

[13] 篠原武, 有川節夫. 日本語テキスト用の Aho-Corasick 型パターン照合アルゴリズム. 情処自然言語処理研究会 NL 52-4, 1985.