ツリー型不定形文書からの部分文書の検索手法の検討

依田平[¶]　大月一弘[‡,†]　清光英成[‡]　森下淳也[‡]

¶ 神戸大学大学院総合人間科学研究科　〒657-8501　神戸市灘区鶴甲 1-2-1
‡ 神戸大学国際文化学部　〒657-8501　神戸市灘区鶴甲 1-2-1
† 神戸大学付属図書館研究開発室　〒657-8501　神戸市灘区六甲台町 1-1

E-mail: ¶ yoda@moo.cla.kobe-u.ac.jp, ‡ {ohtsuki,kiyomitu,jm}@kobe-u.ac.jp

あらまし　デジタルアーカイブに格納されたコンテンツに対して, 利用者所望の部分コンテンツを検索する方法を提案する. コンテンツはツリー構造によって関連付けられているが, コンテンツの内容は規定されなくても良いものとする.

本研究においては, AND 検索におけるキーワード間の関係の違いに注目する. ここでは, キーワード間が修飾関係にある場合と, 並立関係にある場合の 2 種類の AND 検索を考える. 利用者はこの意味の違いを「と」と「の」を用いた簡単な表現を用いることで, コンテンツの論理構造を意識せずに問合せができる. 検索システムでは, 利用者の問い合わせを, ツリー構造上に現れるキーワード間の相対的な位置関係に対応させることにより, 部分コンテンツを抽出する.

キーワード　構造化文書,半構造データ,XML,XMLDB,検索,AND 検索,デジタルアーカイブ,デジタルライブラリ

A Retrieval Method for Corresponding Potions from Semi-Structured Data

Taira Yoda[¶]　Kazuhiro Ohtsuki[‡]　Hidenari Kiyomitsu[‡]　and　Jun-ya Morishita[‡]

[¶]Graduate School of Cultural Studies and Human Science, Kobe University　Tsurukabuto 1-2-1, Nada-ku, Kobe City, 657-8501 Japan
[‡]Department of Cross-Cultural Studies, Kobe University　Tsurukabuto 1-2-1, Nada-ku, Kobe City, 657-8501 Japan

E-mail: ¶ yoda@moo.cla.kobe-u.ac.jp, ‡ {ohtsuki,kiyomitu,jm}@kobe-u.ac.jp

Abstract　Digital Archive is a collection of materials that contains books, serials, extracts from newspapers, magazines, journals, leaflets and so on. Recently, it begins to collect non-paper media that are continuous media data such that audio, video and their compositions. Our major objective is to retrieve sub-materials from the archives and provide a view of archives to our users. We divide a material to some sub-materials in its logical structure, and give metadata to each sub-resource. Here, we have to resolve a problem that corresponding metadata are scattered around sub-resources in a material. In this paper, we propose augmented AND operations for providing an effective method to retrieve sub-materials from the archives.

Keyword　Structured Document, Semi-structured Data, Xml, XmlDB, Retrieval, AND query, Digital Archive, Digital Library

1.はじめに

近年，計算機技術や通信技術の長足の進歩により，多様なメディアで構成されたコンテンツをデジタル化して統一的に扱うデジタルアーカイブへの期待が高まっている．デジタルアーカイブの特徴として，格納データの粒度が荒いこと，複合メディアデータも一つの情報単位として扱っていることなどがある．前者は，リーフレットのような極めて軽量な資料からシリーズものの全集まで，資料の規模がまちまちであるということである．後者は，扱いの異なる複数のメディアで一つのコンテンツを構成しているということである. 例えば，神戸大学電子図書館システムで公開されているデジタルアーカイブでは，一枚ものの写真やチラシ，パンフレットといった軽量なものから，テキストと写真や図，表といった扱いの異なる複数のメディアで構成された書物などといった様々な規模の資料が扱われている．

資料全体を情報単位とすると利用者所望の情報を提供するためには, 利用者の要求を満たす資料を特定するだけでなく, 該当する部分あるいは資料中の領域を特定する工夫と，複合メディアコンテンツを統一的に扱う機能が必要となる[1],[2],[3]．このとき，デジタルアーカイブを利用する末端利用者に資料の論理構造についての知識を前提とした部分抽出のための問合せを記述させることは適切ではない．そこで検索システムには，利用者が資料の構造を意識せずに問合せができ，その問合せの内容と資料の構造を考慮することで利用者の問合せ要求に応える部分資料検索機能が必要となる．

本研究では，論理構造による包含関係に基づいて資料を分割し，分割された部分資料間の包含関係を階層的な木構造グラフによって関連付けて，資料をデジタルアーカイブに格納する．デジタルアーカイブでは多様な資料を取り扱っているため，各資料を表現するツリーの深さが不定であるような構造となる．このような不定形半構造データに対し，このような要求を満たす検索を実現するために, 以下のことを行った.

(1)資料の特定だけでなく，資料の該当部分を検索

単一のキーワードによる検索結果は資料の部分，複数のキーワードによる検索結果は各キーワードによる検索結果の間の関連を評価することで得られるようにした.

(2)AND 検索を拡張

あるキーワードが別のキーワードを修飾するような意味でアーカイブを検索する場合，それぞれのキーワードを含む部分がツリーグラフ上の同一パス上に存在すると考えられる．同様に，二つのキーワードが並立関係にある場合は，それぞれのキーワードを含む部分が資料の別の場所，あるいは近傍に存在すると考えられる．このことから，各キーワードを含む部分の資料中の相対的な位置関係を二つに分類した．それぞれの分類は，単一のキーワード検索を行った結果集合間の関係を表現し，対応する 2 種類の拡張 AND 検索を用意した.

(3)拡張 AND 検索を定義

拡張 AND 検索の意味を定義した. 用意した二つの拡張 AND 検索はキーワードが 2 個の場合だけではなく，n個の場合についても特別な演算を用意することなく利用することが出来る．また，2 種類の拡張 AND 検索を複合した AND 検索においては, 複合ルールを設けることにより, 利用者の検索意図に合致した検索結果を得られるようにした.

グラフの形状を利用して部分資料に分割された資料を検索する方法としては, 田島[4]らの web 情報に対する検索方法がある．田島らの方法は，グラフがネットワーク構造となっているため，2 つの資料間の距離をもとに検索を行っているが, 本方式は，もともとの資料の大きさがわかっておりグラフが木構造であることを利用した検索を行う特徴をもつ.

また，資料の論理構造を意識せずに部分資料を抽出する方法としては，絹谷らの情報検索技術を用いて XML 文書から XML 部分文書を抽出する検索方法がある[6],[7]．絹谷らの手法は, XML 文書の構造を解析することにより検索単位となりうる部分文書を決定し，その単位に検索を行うことによって部分文書を抽出するが，本方式では，結果となる資料の単位を予め決めずに検索を行い，入力条件に応じて結果となる資料の単位が変わる点に特徴を持つ.

本論文の構成は以下の通りである．第 2 章ではデジタルアーカイブの特徴とデータ構造について述べ，第 3 章ではデジタルアーカイブ検索について考察する. 第 4 章では拡張 AND 検索についての説明を行い，第 5 章では拡張 AND 検索の複合演算について考察する．第 6 章では絞り込み検索についての説明を行う．第 7 章では評価実験について述べ，第 8 章ではデジタルアーカイブに対する本方式の有効性を述べる.

2.アーカイブの特徴とデータ構造

2.1.デジタルアーカイブの特徴

本研究で対象とするデジタルアーカイブは，種別やメディアを問わず様々な種類の資料が網羅的に収容されているものとする．このようなデジタルアーカイブの特徴として,

・種々のコンテンツを寄せ集めたような巨大な資料がある.
・章立てといった論理構造が定型の書物ほどはっきりとしていない資料がある.

といった点が挙げられる.

2.2.デジタルアーカイブ検索に対する要求

このようなデジタルデジタルアーカイブに対する検索には, 利用者が資料の構造を意識せずに最適な部分資料を抽出できる機能が重要である.

検索対象となる資料が巨大な場合，検索結果として一つの資料全体でなく，入力条件に関連の高い部分を抽出して提示したほうが利用者にとって利便である．これは，どのような構造の資料が格納されているのか分からない網羅的なデジタルアーカイブでは特に有効である．ただし，デジタルアーカイブの利用者に，資料の論理構造についての知識を前提とした部分抽出のための問合せを記述させることは適切でない点を考慮しなければならない．そこで，デジタルアーカイブの検索システムには, この節の冒頭で述べた機能が重要となる.

2.3.アーカイブの構成

アーカイブから部分資料を抽出するために，資料を分割しアーカイブに格納する．資料の分割は論理構造による包含関係にのみ基づいて行われ，分割された部分間の関係を階層的なツリー構造グラフに関連付ける．分割は原則的に章節構造に基づいて行うが，写真や図といった個別の資料単位となりうる部分も章節構造と同等に取り扱う．資料の分割の例を図１に示す．分割された図１の部分資料間の関係をツリー構造グラフに関連付けたものを図２に示す．各ノードには子ノードに分割されなかった部分の情報のみが記述される. 例えば, 図１の①，②の部分に記載されている情報は，図２におけるルートノードに記述される．すなわち，リーフノード以外のノードにもそれぞれの部分資料（ルートノードに関しては資料全体）に関する部分情報が記述されている．この分割方式をデータモデル化したものを次節で示す.

2.4.データモデル

アーカイブ中の資料をS_i, アーカイブ中のすべての資料を,

$S = \{S_1, S_2, ..., S_n\}$　（ただし，n ：資料の数）

とする. 資料S_i ($S_i \in S$)は論理構造に基づきm_i個の部分資料に分割される.

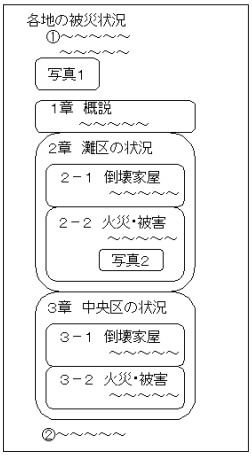

図1　資料の細分化

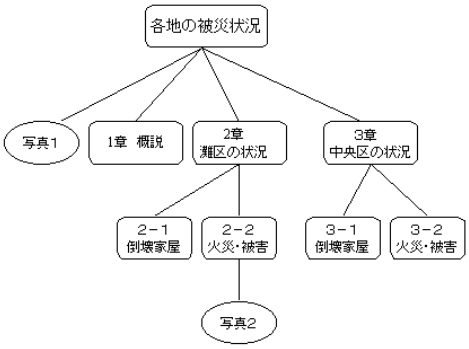

図2　ツリーへの関連付け

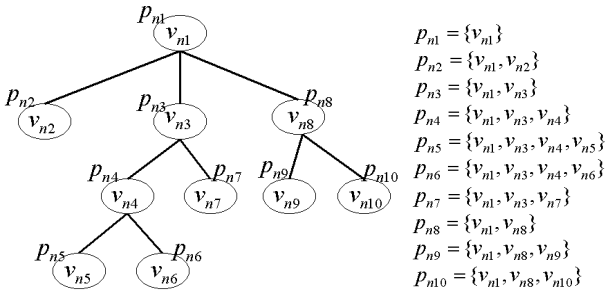

図3　ノードとパス

資料と部分資料の関係をツリーグラフで表現するために次の定義を行う．資料S_iを表すツリーをT_iとし，デジタルアーカイブを表すツリー集合をTとする．一般的には，ツリーT_iはそれに含まれるノードの集合V_iとエッジの集合E_iによって表現される．

$$T = \{T_1, T_2, ..., T_n\}$$
$$T_i = (V_i, E_i)$$
$$V_i = \{v_{i1}, v_{i2}, ..., v_{im_i}\}$$
$$E_i = \{e_{i1}, e_{i2}, ..., e_{i(m_i-1)}\}$$
(ただし，$1 \le i \le n$)

ツリーT_iにおいて，ルートノードからノードv_{ij}へのパスをp_{ij}とする．ここでは，パスp_{ij}をルートノードからv_{ij}までのパス（経路）上に存在するノードの組で表現する．

$$p_{ij} = (v_{i1}, ..., v_{ij}) \quad (\text{ただし，} 1 \le j \le m_i)$$

図3にツリーT_nにおけるノードとパスを示す．各ノードは，パスによって代表され，あるパスを求めることができたなら，そのパスで代表されるノードと，そのノードのツリー上での位置が特定できることが分かる．つまり，ツリーT_iに含まれるそれぞれのノードを代表するパスの集合をP_i（パスp_{ij}が元となる）とすると，ツリーT_iはこのパス集合P_iによって次のように表現できる．

$$T_i = P_i$$
$$P_i = \{p_{i1}, p_{i2}, ..., p_{ij}\}$$

また，各ツリーにおける各ルートノードはそれぞれの資料を代表するものとして扱う．部分資料はサブツリーとし，それぞれのサブツリーを代表するものは，それぞれのサブツリーのルートノードとし，ノードv_{ij}をルートノードとするサブツリーを$ST(v_{ij})$と表現する．サブツリー$ST(v_{ij})$に含まれるノードのパスは，必ずそのルートノードのパスを含む．このことから，サブツリー$ST(v_{ij})$はパスp_{ij}を含むパス集合で記述できる．

$$ST(v_{ij}) = \{p_{ik} \mid p_{ik} \supseteq p_{ij}\}$$

p_{ik}：サブツリー$ST(v_{ij})$に含まれるノードへのパス．
（ただし，ここでのkは任意の自然数）

図3の例で，ノードv_{n3}をルートノードとするサブツリーを考える．このサブツリーを構成するノードは，$v_{n3}, v_{n4}, v_{n5}, v_{n6}, v_{n7}$の5つのノードであるが，これらのノードへのパスはそれぞれ必ずノードv_{n3}へのパスを含んでいることが分かる．つまり，サブツリーはそのルートノードへのパスが分かれば構成できることが分かる．

このようにサブツリーをそのルートノードへのパスで代表させ，パスはノードの組として表すことによって，検索演算を定式化する．なお，検索演算においては，パスを単にノードの集合として扱う場合がある．

3.デジタルアーカイブ検索

3.1.元資料に対するAND検索

ツリーに対するAND検索には，個々のノードに対するAND検索と，複数のノードをまとめたものに対するAND検索の2種類がある．個々のノードに対するAND検索によって部分資料が選出できるものの，個々のノードを完全に独立したものとして取り扱った場合，資料の分割のために，元々は同一の資料に含まれていたキーワードが共起しなくなるという問題が生じる．このためにAND検索に関しては何らかの形で複数のノードをまとめたものに対して行う必要が生じる．

3.2.AND検索におけるキーワード間の意味の違い

(1)キーワード間の関係

AND検索におけるキーワード間の関係には，本来意味の違いがある[5]．ここでは，これを大きく2つに分けて，一方は2語が修飾関係になる場合，他方は2語が並立の関係にある場合とする．例えば，**「灘」**AND**「被害」**，**「灘」**AND**「芦屋」**という2つの検索を考える．このとき，前者の検索意図は**「灘の被害」**と考えられ，これは**「灘」**が**「被害」**を修飾していると捉えることができる．後者の検索意図は**「灘と芦屋」**と考えられ，これは**「灘」**と**「芦屋」**が並立の関係で使われていると捉えることができる．

(2)ツリー上でのキーワードの分布

包含関係によって分割された資料においては，あるキーワ

ードが別のキーワードを修飾するような場合, 一方のキーワードを含む部分が他のキーワードを含む部分に含まれると考えられる．これは, 部分資料をそのルートノードで表した場合, そのルートノードが同一パス上に存在するということである．例えば,**「灘の被害」**に関する資料は,「灘で分類された中の被害という部分」か「被害という項目の中の灘という部分」にあると考えられる．これに対し，二つのキーワードが並立関係にある場合は, それぞれのキーワードを含む部分が資料の別の場所, あるいは近傍に存在すると考えられる．例えば,**「灘」**と**「芦屋」**に関する資料は,**「灘」**を含んだ部分と**「芦屋」**を含んだ部分が同一パス上よりも，資料中の別の場所，あるいは近傍にあると考えられる．

このことから, キーワードを含む部分資料の資料中での相対的な位置関係を利用すれば，AND 検索における意味の違いを区別した検索が行えるものとなる．

(3)キーワード間の関係の違いに対応した検索

以上のことから，本システムでは AND 検索を拡張し，利用者は AND 検索時にキーワード間の関係を,

・「と」：２語が並立関係

・「の」：２語が修飾関係

と入力する．システム側ではそれらに対応した拡張 AND 演算子を用意し，関係の違いを区別した AND 演算を行う．

4.検索演算

本アーカイブの構造を考慮し，部分資料を抽出する検索, 及びキーワード間の関係の違いに対応した AND 検索が行えるように検索演算を定義する．

4.1.単純検索

単一のキーワードによる検索を単純検索と呼ぶ．この検索で求める部分資料は，キーワードを含むノードをルートノードとするサブツリーとする．これにより，利用者は単にキーワードを指定するだけで，そのキーワードに関連の高い部分資料を抽出することが可能となる．サブツリーは，そのルートノードのパスが分かれば構成できるために，データモデルにおいて単純検索のデータ操作を次のように定義する．

定義１　単純検索

単純演算は，利用者の入力したキーワードを含むノードへのパスを求める演算である．

キーワード k を含むノードへのパスを p_{ij}^{k} とすると，単純検索はこのパス p_{ij}^{k} を求める演算である. また, キーワード k による単純検索の結果集合を Q^{k} とすると, Q^{k} の元には p_{ij}^{k} が入り，Q^{k} は次のように表せる．

$$Q^{k}=\{p_{ij}^{k} \mid k \in key(v_{ij})\}$$

$key(v_{ij})$：ノード v_{ij} に含まれるキーワード集合

この Q^{k} が単純検索で求められたサブツリー集合である．

図４に単純検索の例を示す．図中の k はキーワード，○はノードを表し，太線で囲まれた部分は k を含むノードをルートノードとするサブツリーを表す．単純検索によって，キーワード k を含むノードへのパスが求められ，単純検索の結果となる部分資料（太線で囲まれたサブツリー）を得る．

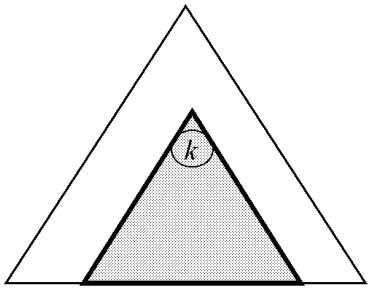

図４　単純検索の例

4.2.単純 AND

単一のノードに対する AND 検索を単純 AND と呼ぶ. この検索で求める部分資料も単純検索と同様，キーワードを含むノードをルートノードとするサブツリーとする．データモデルにおける単純 AND のデータ操作を次のように定義する．

定義２　単純 AND

単純 AND は，利用者の入力したキーワード集合を含むノードへのパスを求める演算である．

利用者によって入力されるキーワード集合を,

$$K=\{k_1,k_2,...,k_n\}$$

(ただし，k_n：キーワード，n：自然数)

とする．キーワード集合 K を含むノードへのパスを p_{ij}^{K} とすると，単純検索はパス p_{ij}^{K} を求める演算である．キーワード集合 K による単純 AND の検索結果集合を Q^{K} とすると，Q^{K} の元には p_{ij}^{K} が入り，Q^{K} は次のように表せる．

$$Q^{K}=\{p_{ij}^{K} \mid K \subseteq key(v_{ij})\}$$

この Q^{K} が単純 AND で求められたサブツリー集合である．また，Q^{K} は個々のキーワードによる単純検索の検索結果集合を利用し，次のようにも表せる．

$$Q^{K}=Q^{k_1} \cap Q^{k_2} \cap ... \cap Q^{k_n}$$

($Q^{k_1},Q^{k_2},...,Q^{k_n}$ ：キーワード $k_1,k_2,...,k_n$ それぞれによる単純検索の検索結果集合)

4.3.親戚 AND

「と」の関係の AND 検索を行うために親戚 AND を準備する．これは，二つのキーワードがアーカイブ構造上で並列関係にある部分資料を検索する．親戚 AND で求める部分資料は，二つのキーワードが含まれる最小のサブツリーとする．データモデルにおける親戚 AND のデータ操作を次のように定義する．

定義３　親戚 AND

親戚 AND は，一方のキーワードを含むノードへのパスと，他方のキーワードを含むノードへのパスとの共通のパスを求める演算である．

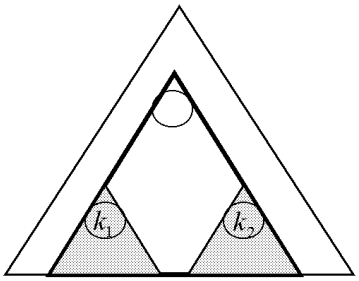

図５　親戚 AND の例

キーワードk_1とk_2による単純検索によってツリーT_iから得られたパスの集合をそれぞれ$Q_i^{k_1}$，$Q_i^{k_2}$とする．以下では，簡単のため，それぞれのパスからツリーを特定するための添え字iは省略して表記する．例えば，パスp_{ij}はp_jと表記する．$(p_m^{k_1}, p_n^{k_2}) \in Q_i^{k_1} \times Q_i^{k_2}$のとき，親戚 AND はこれら二つのパスの間に次が成立するような関係である．

$$p_m^{k_1} \cap p_n^{k_2} \neq \phi$$

この関係をパス$p_m^{k_1}$と$p_n^{k_2}$は**親戚関係**であると呼ぶ．

キーワードk_1，k_2による親戚 AND 演算の解となるサブツリーのルートノードへのパスX_{mn}は以下で表現できる．

$$X_{mn} = p_m^{k_1} \cap p_n^{k_2}$$

キーワードk_1, k_2による親戚 AND を「$k_1 + k_2$」と表し，ツリーT_iに対する親戚 AND 検索の結果集合を$Q_i^{k_1+k_2}$とすると，$Q_i^{k_1+k_2}$は次のように定義できる．

$$Q_i^{k_i+k_2} = \bigcup_m \bigcup_n (X_{mn})$$

従って，キーワードk_1，k_2による資料全体に対する親戚 AND 検索の結果集合$Q^{k_1+k_2}$は次のように定義できる．

$$Q^{k_1+k_2} = \bigcup_i Q_i^{k_1+k_2}$$

また親戚 AND では交換則が成り立ち，

$$k_1 + k_2 \equiv k_2 + k_1$$
$$Q^{k_1+k_2} = Q^{k_2+k_1}$$

キーワードの指定順序によらず同一の結果を得る．

図５に親戚 AND の例を示す．図中のk_1，k_2はキーワード，○はノードを表し，色の付いた部分が各キーワードによる検索で得られたサブツリーを表す．二つのキーワードを含む最小のサブツリーは太線部であり，これが親戚 AND によって求められた部分資料を表す．

この検索は，基本的には元資料に対する AND 検索で，Web 検索における AND 検索と同じである．異なる点は，Web 検索における AND 検索ではキーワードの出現位置にかかわらず検索結果がページ単位だったのに対し，親戚 AND ではキーワードを含んだノード間の距離が近ければ検索結果が元資料から抜き出された部分資料単位になる点である．さらに，この検索における特徴的な点は，キーワードが含まれるノードの相対的な位置関係に基づいてのみ結果の選定が行われる点である．このような評価法を取ることで，入力キーワードに応じて最小の部分資料を柔軟に選出することが可能となる．

4.4. n項の親戚 AND

n個のキーワード$k_1, k_2, ..., k_n$に対するパス$p^{k_1}, p^{k_2}, ..., p^{k_n}$が，

$$p^{k_1} \cap p^{k_2} \cap ... \cap p^{k_n} = \phi$$

を満たすとき，パス$p^{k_1}, p^{k_2}, ..., p^{k_n}$は**親戚関係**であると呼ぶ．$n$個のパスが親戚関係であるサブツリー，即ち，$n$個のキーワードをすべて含む部分資料を検索する演算を$f(k_1, k_2, ..., k_n)$とすると，

$$f(k_1, k_2, ..., k_n) = p^{k_1} \cap p^{k_2} \cap ... \cap p^{k_n}$$

と表すことができる．

n個のキーワードに対する親戚 AND 演算の合成演算，

$$k_1 \ \textbf{\textit{親戚 AND}} \ k_2 \ \textbf{\textit{親戚 AND}} \ k_3 \ \ ... \ \textbf{\textit{親戚 AND}} \ k_n$$

の結果をQとする．それぞれのキーワードによる単純検索で得，この演算における最初の二つのパスによる親戚 AND 演算の結果のパスを$p^{k_{12}}$とすれば，

$$p^{k_{12}} = p^{k_1} \cap p^{k_2}$$

である．このパス$p^{k_{12}}$と３つ目のパスp^{k_3}とで親戚 AND 演算を行った結果を$p^{k_{123}}$とすれば，

$$p^{k_{123}} = p^{k_{12}} \cap p^{k_3}$$
$$= p^{k_1} \cap p^{k_2} \cap p^{k_3}$$

となる．同様に，演算を繰り返すことによって，

$$p^{k_{12...n}} = p^{k_1} \cap p^{k_2} \cap ... \cap p^{k_n}$$

となる．これはn個のキーワードをすべて含む部分資料を検索する演算を$f(k_1, k_2, ..., k_n)$と同値である．このことから，複数個のキーワードをすべて含む部分資料を検索する演算fは，親戚 AND 演算の合成演算で表すことができる．

また，親戚 AND 演算の合成演算においても交換則が成り立ち，演算の処理順序，即ち，キーワードの指定順序によらず同一の検索結果を得る．

4.5. 直列 AND

「の」の関係の AND 検索を行うために直列 AND を準備する．この検索は，二つのキーワードを含むノードが同一パス上に並ぶ場合を検索する AND 検索である．

アーカイブ構造上で同一パス上に並ぶということは，一方のキーワードを含むサブツリーが，他方のキーワードを含むサブツリーを包含していることを意味する．従って，サブツリーのルートノードをパスで代表させる本モデルでは，直列 AND 演算の解は，包含されるサブツリーのルートノードを代表するパスとする．

包含される側を解とすることには，次のような利点がある．デジタルアーカイブで扱う資料の特徴として，コンテンツを寄せ集めたような巨大な資料があることと，論理構造が書物ほどはっきりとしていないといった点があった．このような資料においては，包含する側のサブツリーを結果として利用者に提示した場合，利用者はその結果からほしい部分を特定するのに時間がかかるというデメリットを生じる．このため，直列 AND 演算の解は包含される側のサブツリーとすること

が好ましいことが分かる．データモデルにおける直列 AND のデータ操作を次のように定義する．

定義４　直列 AND

直列 AND は，一方のキーワードを含むノードへのパスが，他方のキーワードを含むノードへのパスに完全に含まれる場合，その含んでいる方のパスを求める演算である．

キーワード k_1 と k_2 による単純検索によってツリー T_i から得られたパスの集合をそれぞれ $Q_i^{k_1}$，$Q_i^{k_2}$ とする．$(p_m^{k_1}, p_n^{k_2}) \in Q_i^{k_1} \times Q_i^{k_2}$ のとき，直列 AND はこれら二つのパスの間に次が成立するような関係である．

$$p_m^{k_1} \subseteq p_n^{k_2} \text{ 又は } p_m^{k_1} \supseteq p_n^{k_2}$$

この関係をパス $p_m^{k_1}$ と $p_n^{k_2}$ は**直列関係**であると呼ぶ．

キーワード k_1，k_2 による直列 AND 演算の解となるサブツリーのルートノードへのパス Y_{mn} は以下で表現できる．

$$Y_{mn} = \begin{cases} p_m^{k_1} \ (p_m^{k_1} \supseteq p_n^{k_2}) \\ \\ p_n^{k_2} \ (p_m^{k_1} \subseteq p_n^{k_2}) \end{cases}$$

キーワード k_1，k_2 による直列 AND を「$k_1 \# k_2$」と表し，ツリー T_i に対する直列 AND 検索の結果集合を $Q_i^{k_1 \# k_2}$ とすると，$Q_i^{k_1 \# k_2}$ は次のように定義できる．

$$Q_i^{k_i \# k_2} = \bigcup_m \bigcup_n Y_{mn}$$

従って，キーワード k_1，k_2 による資料全体に対する直列 AND 検索の結果集合 $Q^{k_1 \# k_2}$ は次のように定義できる．

$$Q^{k_1 \# k_2} = \bigcup_i Q_i^{k_1 \# k_2}$$

また直列 AND では交換則が成り立ち，

$$k_1 \# k_2 \equiv k_2 \# k_1$$
$$Q^{k_1 \# k_2} = Q^{k_2 \# k_1}$$

キーワードの指定順序によらず同一の結果を得る．

図 6 に直列 AND の例を示す．図中の k_1，k_2 はキーワード，○はノードを表し，色の付いた部分が各キーワードによる検索で得られたサブツリーを表す．太線部で囲まれたサブツリーが，直列 AND によって求められた部分資料を表す．

4.6. n項の直列 AND

n個のキーワード間がすべて修飾関係にある部分資料を検索する演算を $g(k_1, k_2, ..., k_n)$ とする．

$$P = \{p^{k_1}, p^{k_2}, ..., p^{k_n}\}$$

とすると，パス集合 P に含まれるパスがすべて直列関係にあるならば，次の関係が成立する．

$$p_i \subseteq p_j \text{ 又は } p_i \supseteq p_j$$
$$(1 \le i \le n, 1 \le j \le n, i \ne j)$$
$$p_i, p_j \text{ は } P \text{ 中の任意の要素}$$

n個のキーワードに対する直列 AND 演算の合成演算，

k_1 ***直列 AND*** k_2 ***直列 AND*** k_3 ***...直列 AND*** k_n

の結果を Q とすると，親戚 AND の場合と同様に，Q と

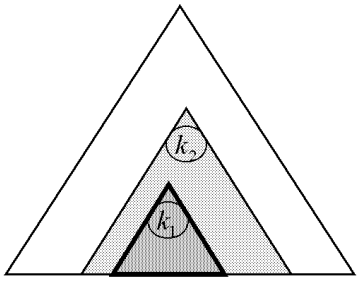

図 6　直列 AND の例

$g(k_1, k_2, ..., k_n)$ は同値となる．すなわち，g は直列 AND の合成演算で表すことができる．

また，直列 AND 演算の合成演算においても交換則が成り立ち，演算の処理順序，即ち，キーワードの指定順序によらず同一の検索結果を得る．

5.拡張 AND 検索の複合演算

第 4 章ではキーワード間の関係が１種類である拡張 AND 検索についての定義を行ったが，ここでは２種類のキーワード間の関係が入力された拡張 AND 検索について考察する．

２種類のキーワード間の関係が入力される AND 検索を，拡張 AND 検索の複合演算と呼ぶ．複合演算では，２種類の拡張 AND 検索の合成演算によって結果を求める．しかしながら，このとき，利用者が入力式に込めた意味と，演算ロジックにおける意味が異なっているため，利用者の検索意図を解釈せずに検索を行えば，検索結果は得られるものの，それは利用者の検索意図と合致しない部分資料となる場合がある．

5.1.「と」と「の」の入る検索

「灘と芦屋の被害」という,「と」と「の」を両方使った検索を考える．この検索における利用者の意図は，少なくとも**「芦屋」**と**「被害」**が修飾関係になっていると解釈できる．この入力の拡張 AND 検索は,

灘 ＋ 芦屋 ＃ 被害

（＋：親戚 AND，＃：直列 AND）

という AND 検索となる．この検索の結果を，**「灘」**による単純検索によって得られたパスと**「芦屋」**による単純検索によって得られたパスとで親戚 AND 演算を行い，その結果のパスと**「被害」**による単純検索によって得られたパスとで直列 AND 演算を行う合成演算によって得るとする．このとき検索結果として，**「芦屋」**と**「灘」**が修飾関係になっていない部分資料が結果となる場合がある．

図 7 のツリーからキーワード k_1，k_2，k_3 によるそれぞれの単純検索の結果として，パス $p_2^{k_1}$, $p_3^{k_2}$, $p_6^{k_3}$ が得られたとする．“ k_1 ＋ k_2 ＃ k_3 ”を親戚 AND，直列 AND の順で処理した場合，先の親戚 AND 演算によってパス p_1 が得られ，次の直列 AND 演算によってパス $p_6^{k_3}$ が得られる．しかしながら，このパス $p_6^{k_3}$ によって代表される部分資料 $ST(v_6)$ は，k_2 と k_3 が修飾関係で使われていない部分資料であることがこの図から分かる．

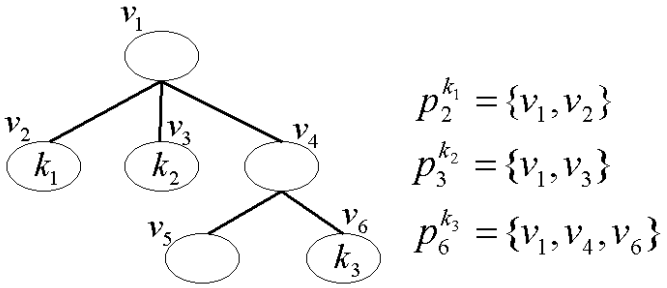

図 7　複合演算

このような利用者の意図に合致しない部分資料を選出することは，演算に優先順位を設けることによって防ぐことができる．つまり，先に直列 AND 演算を行い，次に親戚 AND 演算を行うという順序で合成演算を行えば，少なくともキーワード k_2 と k_3 が修飾関係で使われている部分資料が得ることができる．

5.2.一語が２語以上を修飾する検索

キーワード間の関係の「の」は，あるキーワードが他のキーワードを修飾することを意味するが，場合によっては１つのキーワードが２つ以上のキーワードを修飾することがある．このような場合，利用者は（）を利用し，入力キーワード間の修飾関係を明確に指定するものとする．例えば，**「灘の被害と復興」**では，

「灘の(被害と復興)」　　…①

と入力する．この時，**「灘」**というキーワードは**「復興」**と**「被害」**の両方を修飾している．すなわち，この入力における利用者の意図は，

「灘の被害」と「灘の復興」…②

であると解釈できる．つまり，利用者の入力では，①式と②式は等価であり，（）には分配則が成り立っている．

一方，①，②式の拡張 AND 検索は，

灘 ＃ (被害 ＋ 復興)　　…①’

灘 ＃ 被害 ＋ 灘 ＃ 復興　…②’

であるが，複合演算では（）に分配則が成立せず，この２式は等価ではない．

図 7 のツリーからキーワード k_1，k_2，k_3 によるそれぞれの単純検索の結果として，パス $p_2^{k_1}, p_3^{k_2}, p_6^{k_3}$ が得られたとする．“ k_1 ＃ （k_2 ＋ k_3）”を，（）内の親戚 AND 演算から先に行った場合パス p_1 が結果として得られ，次の直列 AND 演算によってパス $p_2^{k_1}$ が得られる．しかしながら，このパス $p_2^{k_1}$ によって代表される部分資料 $ST(v_2)$ は k_1 と k_2 及び k_3 が修飾関係で使われていない部分資料であることが分かる．一方，“ k_1 ＃ k_2 ＋ k_1 ＃ k_3 ”を，直列 AND 演算から先に行い，その結果同士で親戚 AND 演算を行った場合は，図 7 のようなサブツリーは選出されず，k_1 と k_2 並びに k_3 が修飾関係で使われている部分資料のみが選出される．つまり，（）が入る式は，利用者の意図と演算ロジックが一致しないということであり，また，複合演算において（）は演算の優先順位の（）として扱うことができない．このことから，複合演算では，（）が入力された場合は，（）の扱いに注意が必要である．

5.3.複合演算ルール

拡張 AND 検索の複合演算では，以下の演算ルールを設けることにより，利用者の検索意図に合致した部分資料のみを選出できるようにする．

(1)AND 演算の優先順位を直列 AND＞親戚 AND とする．

(2)（）がある場合は，個々の演算を行う前に，利用者の検索意図に基づいて（）を展開する．

即ち，（）を展開してから拡張 AND 演算へ対応させ，拡張 AND 演算の処理の優先順位に従って結果を求めるということである．このようなルールを設けることにより，検索システムは，利用者の複雑な入力に対する特別の演算を用意することなく，一定のルールに従って解を得ることができるという利点を得る．

6．絞り込み検索

絞り込み検索は，(1)先の検索で得られた検索結果集合と(2)絞り込みの際の入力条件によって得られた検索結果集合とで，集合演算を行うことによって行う．ここで，部分資料に対する検索には２種類の絞り込み検索がある．一方はより小さい部分に絞り込む検索で，他方は一般的な結果の資料数を減らす絞り込み検索である．

この２種類の絞り込み検索に対応するために，(1)と(2)の間の集合演算を，

直列 AND：より小さい部分に絞り込みたい場合

親戚 AND：一般的な資料数を減らす検索を行いたい場合

とする．

７.評価実験

拡張 AND 検索が検索意図に合致した部分資料を抽出することが出来ているのかを検証するために，プロトタイプのシステムを作成し，次の評価実験を行った．

1)利用者がキーワードとキーワード間の関係を指定することで意図する部分が抽出できているか？

データは，2000 年 11 月現在の神戸大学電子図書館のデータを用いた．データ数は，元資料数が 40429 件,ノード数が 97555 件，部分資料への分割が行われている資料数が 3092 件であった．

表１に**‘ 明石 ’and‘ 被害状況 ’**,**‘ 芦屋 ’and‘ 被害状況 ’**,**‘ 明石 ’and‘ 芦屋 ’**の拡張 AND 検索等の実験結果を示す．**「明石」**を含むノード数は 421 件,**「芦屋」**を含むノード数は 983 件,**「被害状況」**を含むノード数は 273 件であった．項目 5 は指定したキーワードを含んでいる資料の数，項目 6 は指定したキーワードを共に含んでいるノードの数で，両者を比較すればノードのみに対する検索では，本来該当されるべき資料をかなり得損ねていることがわかる．項目 7 は親戚 AND で得られた部分資料数を表しており，項目 5 との比較をすれ

表 1　拡張 AND 演算の検索結果の違い

1	単語A	明石	芦屋	明石
2	単語B	被害状況	被害状況	芦屋
3	単語A（N）	421	983	421
4	単語B（N）	273	273	983
5	A and B（M）	14	30	53
6	単純検索（N）	2	12	23
7	親戚AND（N）	23	56	89
8	直列AND（N）	3	14	24

（N）：ノード数　（M）：資料数

ば，検索結果がいくつかの部分資料に分かれて抽出されていることがわかる．

利用者の意図が**「明石の被害状況」**，であった場合に，表１の**‘ 明石 ’ and ‘ 被害状況 ’** で直列 AND の項目を見れば，検索結果は 3 件となっている．結果の資料を実際に確認したところ，同一ノードに記述されていたものが 2 件，直列に並んでいるノードのペアに記述されていたものが 1 件であった．これらはすべて**「明石の被害状況」**に関する資料，例えば「兵庫県南部地震による震災の記録」の**「被害状況」**に含まれる**「明石市**にある附属幼稚園の東塀が倒壊」，であった．また，親戚 AND の結果の内，直列 AND に該当しないものを確認したところ，これらは**「明石の被害状況」**とは関係のない資料であった．これらのことは**「芦屋の被害状況」**に関しても同様で，この結果，直列 AND は修飾関係の 2 語を調べる時に有効であることが確認できた．また，**‘ 明石 ’ and ‘ 芦屋 ’** で親戚 AND の項目を見れば，検索結果は 89 件であった．これらの資料を確認したところ，検索結果となった部分資料は，検索意図に合致するものであった．

このことから，拡張 AND 検索を上手く使い分けることによって，利用者の意図を反映させた結果を得ることが可能となることが分かった．

８.電子図書館用デジタルアーカイブに対する適合性

本方式では，資料の分割は論理構造による包含関係に基づいて行われるが，分割資料から構成されるツリー内の各ノードにつけられる属性間に関する特別な制約はない．例えば，一つの資料の中に，同じ属性が重複して存在しても構わない．仮に，本の中で他人の本の文章を引用していたとする．この場合，ルートノードには，<著者名>，<題名>などといった属性があり，その本の著者名や題名が付与されていると思われる．また，引用の部分をひとつのノードとし，その部分にも引用元の<著者名>や<題名>という属性をつけることができる．この場合利用者は，<著者>＝"A"　直列 AND <著者>="B" といった条件式を書くことで部分を特定することができる．

ノード間の属性に制約がないことは，個別に作成したツリーを統合する場合にも役立つ．例えば，個別に登録した本がシリーズものであった場合，ツリー構造をあらわす情報のみを書き換えるだけで，ひとつのツリーとして再構成することができる．この場合，各ノードに付与された属性やその値を変更する必要はない．

別な言い方をすれば，デジタルアーカイブ作成時において，作成者は，同一資料内の他の部分資料に付与される属性にあまり気をつかわずに各部分資料（ノード）に属性を付与することができ，大量の資料をアーカイブ化する作業が比較的簡単に行えるものと考える．複数人によって共同でアーカイブを作成する場合，データ作成者の主観の違いによって細分化のルールが異なったとしても，論理構造による包含関係に基づいてツリーを作成するという条件さえ満たしていれば，検索の精度は多少落ちるが，システムとしては機能するものと考えられる．

このように，本方式で想定するアーカイブにおいては，データの統一性や一貫性を保つためのデータスキーマの制約を設ける必要もなく，大量のデータを人海戦術でアーカイブする場合の，作業者の負担を軽減できるという特徴をもっており，比較的簡便にアーカイブを作成する上で有益な手段となると考えられる．

９．まとめ

本稿では，部分資料間の相対関係を有効に使うことによる，利用者の検索意図に適した部分資料の抽出方法の提案並びに定式化を行った．

今後は，さらなる評価実験を行い，提案方式の課題，問題点を導き出し，方式の改良を行うと共に，拡張 AND 検索の検索結果のスコアリング等に関しても研究する予定である．

参考文献

[1] 依田平，大月一弘，森下淳也，清光英成，“ デジタルアーカイブに対する効率的な検索の提案　―神戸大学電子図書館システムを例として― ”，情報処理学会シンポジウムシリーズ 18 号　人文科学とコンピュータシンポジウム論文集，pp.259-266，2001.

[2] 依田平，小椋正道，大月一弘，森下淳也，清光英成，“ 電子図書館用デジタルアーカイブの検索方法の検討 ”，情報処理学会研究報告 70 号，pp.469-476，2001.

[3] 渡邊 隆弘，“ 震災アーカイブにおけるメタデータの設計 ”，情報処理学会シンポジウムシリーズ 17 号　人文科学とコンピュータシンポジウム論文集，pp.89-96，2000.

[4] K. Tajima,K. Hatano,T. Matukura,R. Sano,
K. Tanaka:Discovery and Retrieval of Logical Information Units in Web,(invited) Proc. of WOWS, (in conj. with ACM DL'99), Berkeley, CA, pp 13-23, Aug. 1999.

[5] 山本昭,“ ブール検索における and の使用法と意味論　―共出現の諸ケースと検索者側での対応 ”，情報の科学と技術，50 巻，10 号，p.501，2000.

[6] 波多野賢治，渡邉正裕，吉川正俊，植村俊亮，“ 情報検索技術を用いた部分文書構造の自動抽出 ”，IPSJ TOD，Vol.42，No.SIG8(TOD10)，pp.36-46，2001.

[7] 絹谷弘子，波多野賢治，吉川正俊，植村俊亮，“ XML 文書の文書構造と内容を用いた部分文書の抽出手法 ”，IPSJ TOD，Vol.43，No.SIG2(TOD13)，pp.80-93，2002.