オンライン辞書のハイパーテキスト化手法

内藤広志[†*] 山下真司[†*]
松山洋一[†*] 柵木孝一[†*]

CD-ROM などの普及により，パソコンやワークステーション上で手軽に電子化辞書が利用できるようになってきた．しかし多くの電子化辞書は冊子体の辞書と同様に見出し語による検索が主な機能である．そこで，電子化辞書をハイパーテキスト技術を用いて再構築すれば，ハイパーテキストのフィルタリング，ブラウジング，ナビゲーションなどの様々な検索・表示機能を享受できるようになる．しかし，既存の文書をハイパーテキスト化するために必要な作業は大きな負担であり問題になっている．本論文では，電子化辞書のハイパーテキスト化作業の形式化・自動化手法について述べる．本手法では，まず抽象辞書，およびハイパーテキストデータモデルを用いて原データの構造分析を行う．抽象辞書とは様々な辞書が共通に持っている構造を抽出したもので，各辞書の構造分析の基準となる．ハイパーテキストデータモデルはハイパーテキストのネットワーク構造を記述するために設計したデータモデルである．そしてその分析に基づいてデータの整形，ハイパーテキスト化，DB への格納の各ステップからなる原データの構造変換を行う．

Transforming Text into Hypertext for an On-line Dictionary

Hiroshi Naito,[†*] Shinji Yamashita,[†*] Youichi Matsuyama[†*]
and Koichi Masegi[†*]

With spread of CD-ROM, electronic dictionaries are available on personal computer and workstation, but only serve searching function by word as well paper dictionary does. If they are transformed into hypertext, they improve on viewing and retrieving information. It, however, is difficult to transform linear structure of them into hypertext structure. This paper describes formal and semiautomated method to transform electronic dictionary into hypertext. In this method, at first, structure of original data is analyzed based on Abstract Dictionary (common structure of dictionaries) and structure of target hypertext is designed using Hypertext Data Model. Next, original data is transformed into hypertext based on the analysis. This process is made from three steps, arrangement of data structure, restructuring into non-linear and storing into DB. We've developed Hypertext Dictionary Hydra (Hypertext Dictionary Reading Accessories) with this method.

1. はじめに

CD-ROM などの大容量記憶メディアの普及により，パソコンやワークステーション上で手軽に電子化辞書が利用できるようになってきた．しかし，電子化辞書の多くは，冊子体（書籍形式のこと）の辞書と同様に見出し語による検索しかできない．一方，情報検索やマルチメディアなどの分野で，ノードとリンクからなる単純なモデルによって情報を表現するハイパーテキストが注目を集めている．これは，ハイパーテキストのネットワーク構造を用いることで強力な検索機能が実現できるためである[2]．この技術を用いて電子化辞書を再構築すれば，ハイパーテキストの様々な検索・表示方法により辞書の機能を高めることができる．例えば，各見出しの語義中に現れる反対語や参照語を，見出し語間の関係を表すリンクとして表現し，マウスでリンクを表すボタンを選択することで，対応する見出し語を容易に検索できる．また，フィルタリング機能を用いて，一般ユーザにとって重要性の低い情報の表示を止めることができる．

今日，辞書の多くは電子化され，計算機可読辞書と呼ばれる計算機で処理可能な形式になっている．これをハイパーテキストに変換できれば，ハイパーテキスト辞書の構築は容易になる．冊子体形式の構造を持つデータをハイパーテキスト構造に変換することをハイパーテキスト化と呼ぶ．ハイパーテキスト化とは，原データをノードに分割し，リンクを明示化することである．

† キヤノン(株)情報システム研究所知能工学研究部
Information Systems Research Center, CANON INC.
* 現在 キヤノン(株)システムエンジニアリングセンター
Presently with System Engineering Center, CANON INC.

ハイパーテキスト化手法には次の二つのアプローチが考えられる.

(a) リンクの自動抽出を主眼に置いたアプローチ
(b) ハイパーテキスト構造の設計を重視したアプローチ

(a)は,原データの構造は変えずに,データ中からリンクを自動的に抽出するアプローチである.その方法には,「第二章」や「下図」などの参照を示すフレーズや専門用語をパターンマッチングで抽出する方法[14),16)]や,自然言語処理を用いてリンクを抽出する方法[12)]がある.辞書の場合,反対語や参照語などのリンク情報は"↔"などの特殊記号で示されているので,それを見付けることで,リンクは容易に抽出できる.しかし,単にリンクを抽出し,データ間のリンクによる検索を可能にするだけでは,ハイパーテキストの問題点である,認知的過負荷の増大やハイパーテキスト空間における迷子問題に陥りやすい.そのため,(b)のハイパーテキスト構造の設計を重視したアプローチが重要である.また,過去の資産の有効利用を考えると,辞書だけに限らず,ハイパーテキスト化する技術は重要であり,ハイパーテキスト化プロセスの工学的な手法(ハイパーテキスト工学)を確立することが望まれる[5),6)].

また,機械可読辞書は冊子体の辞書をそのまま計算機のデータとして入力したものであるため,冊子体の辞書と同じ問題を持っている.辞書は冊子体と言うリニアな構造を用いて,人間が理解しやすく,少ない空間に効率良く記述することを目的としている.そのため,各辞書項目が明確に分離されていなかったり,記述に多くの曖昧性があるなどハイパーテキスト化する上で多くの問題がある[10),15)].

そこで我々は,データベース分野で用いられているスキーマ設計法[13)]や意味データモデリング[7)],ソフトウェア工学で用いられているシステム分析手法[1)]をハイパーテキスト分野に適応して,辞書の記述の曖昧性の解決やハイパーテキストデータベースの設計・構築まで含めた辞書のハイパーテキスト化技術の手法を検討した.

本論文では,まず2章で計算機可読辞書のハイパーテキスト化に関する課題について述べる.そして,3章でハイパーテキスト化手法の概要を述べ,続いて,4章で本手法中で用いられるデータモデルについてまとめる.そして,5章と6章で我々のハイパーテキスト化手順の詳細について述べる.

2. 辞書のハイパーテキスト化

辞書には多種の情報が記述されている.例えば,新明解国語辞書[11)]を例にとると,各見出しの説明として次のような情報(辞書項目)が書かれている[15)].

1. 見出し, 2. 正書法(漢字表記), 3. 子見出し, 4. 品詞・活用, 5. 重要度, 6. アクセント, 7. 歴史的かなづかい, 8. 原語, 9. 漢語の造語成分, 10. 語義番号, 11. 語釈義文, 12. 用例, 13. 補足的説明, 14. 派生語, 15. 反対語, 16. 参照語.

冊子体の辞書では,これらの多種の項目のタイプを識別し,その始まりや終わりを示すために,項目の出現順序,語義番号,種々の括弧や矢印などの特殊な記号(以後,これらを総称して特殊記号と呼ぶ.),略語が用いられている.例えば,新明解国語辞書には総数62個の特殊記号がある.

ハイパーテキスト化の観点から見ると,各見出しの記述法には次のような問題点ある.

(a) 見出しに直接関連しない情報まで含んでいる.
(b) 項目の出現順序は一定ではない.
(c) 項目が省略される場合が多い.
(d) 特殊記号の用法に曖昧性がある.

(a)の例として子見出しがある.これは何をノードの単位にするかの決定に関連する.子見出しを独立したノードにすべきか,見出しのノードに含めるかの判断が必要である.(b)は,単に順序が代わるだけでなく,出現した項目が関係する項目が変化する.その例として**図1**に"‹›"で示される「反対語」の記述をあげる.図で,見出し「じょうき」の場合は,「上記」の反対語が「下記」であるが,見出し「しょうがく」の場合では,語義1の反対語が「高額」で,語義2の反対語は「多額」と,語義ごとに反対語がある.(c)は項目が必須情報であるかを判断しなければならない.(d)により辞書項目のタイプの抽出が難しくなる.例えば,**図2**に示すように文脈によって変化する.「分野」の"〔 〕"は語釈の補足的説明を,「文房

じょうき①②ジャウ− 【上記】 −する 前に書きしるしてある・こと(文句)。↔下記

じょうとう⓪ジャウ− 【上等】 −な−に ↔下等 ㊀等級が上である・様子(もの)。㊁すぐれていい・様子(もの)。

しょうがく⓪セウ− ㊀【小額】 ↔高額 小さい単位の金額。「−紙幣⑤」 ㊁【少額】 ↔多額 少ない金額。

図1 「反対語」を示す特殊記号"↔"の出現位置の揺れ[11)]
Fig. 1 Various positions where antonym descriptor "↔" occurs.[11)]

具」の“〔 〕”は熟語の構成を,「分封」の“〔 〕”は説明的略号を表す．さらに辞書には“〔”に対応する“〕”がないなど多くの入力エラーがある．

このような問題のため，辞書をハイパーテキスト化する際，辞書にはどのような項目が記述されており，そのデータ構造がどうなっているかを十分に分析する必要がある．そして，ハイパーテキスト構造は，構造分析の結果に基づいて設計されなければならない．したがって，辞書のハイパーテキスト化手法の第一の要件として，辞書の構造分析とハイパーテキスト構造の設計（モデル化）の支援が挙げられる．

また，辞書の構造は国語辞書，英和辞書，和英辞書などの辞書の種類によって異なり，同種の辞書でも出版社によってその構造は異なる．そのため，個々の辞書に対して適切なハイパーテキスト構造を個別に設計することは開発効率が悪い．ハイパーテキスト辞書の構築の効率化の観点から，辞書のハイパーテキスト化手法の第二の要件として，様々な辞書のモデル化の支援が挙げられる．

前述したように辞書の記述には曖昧性やエラーがあるため，ハイパーテキスト化の完全自動化は困難である．しかし，辞書の場合，データサイズが数 MB から数百 MB と極めて大きいので，辞書のハイパーテキスト化手法の第三の要件として，計算機を用いて処理を出来る限り自動化することが挙げられる．

以上をまとめると，辞書のハイパーテキスト化手法において重視すべき点として

（a） 構造分析とモデル化のサポート，
（b） 様々な辞書のモデル化の支援，
（c） ハイパーテキスト化処理の自動化

が挙げられる．

3. 辞書ハイパーテキスト構築手法の概要

本手法では**図 3** のようにハイパーテキスト化に必要な作業をステップに分け，各ステップの作業内容を明確にした．ハイパーテキスト化の作業は，ハイパーテキストデータの構造を設計する**データ定義部作成フロー**と，そこで設計されたスキーマ定義に基づいて実際のデータを変換する**データ変換フロー**の二つに大別される．本図で，網掛けになっている部分が自動化されている処理である．

また，前章で挙げた問題点を解決するために，次の三点をハイパーテキスト化の作業フローで採用した．

（a） ハイパーテキストデータモデル
（b） 抽象辞書
（c） SGML によるデータ記述と構造変換処理

（a）は，ハイパーテキスト構造の図式表現で，データ定義部作成フローのモデル化ステップで使用する．Coad らのダイアグラム[1]を辞書の分析に必要なリンクの分類に基づいて拡張した．詳細は 4 章で説明する．

（b）は，**図 4** に示すように，様々な辞書が共通に持っている構造を抽出したものである．この抽象辞書を詳細化して各辞書の構造を定義できる．新しい辞書をハイパーテキスト化する際，辞書固有の情報だけを抽象辞書に加えれば良いので，モデル化が容易になる．

（c）は，ハイパーテキスト化におけるデータの記述言語として，文書記述の国際規格である **SGML**[8] を

＊ぶんや【分野】〔分けて〕受け持つ範囲．

＊ぶんぼうぐ【文房具】〔「文房」は，書斎の意〕ものを書くために必要な道具．

ぶんぽう【分封】〔古〕大名が自分の領地を割いて，臣下に分けること．

図 2 “〔 〕”の用法の揺れ[11]
Fig. 2 Various meanings of supplement descriptor “〔 〕”.[11]

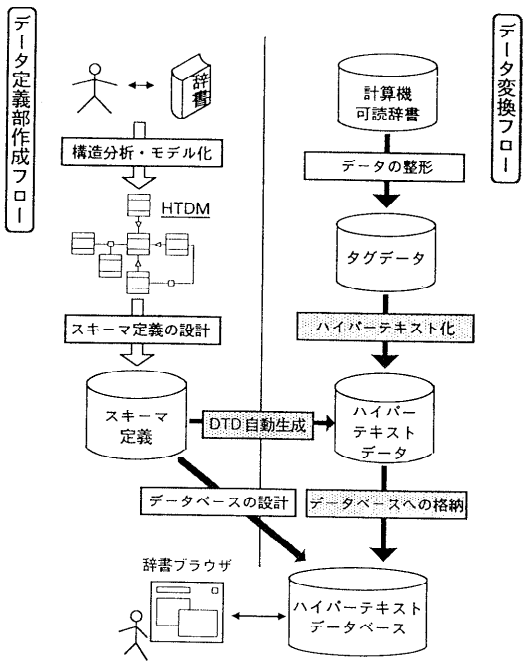


図 3 ハイパーテキスト辞書構築の作業フロー図
Fig. 3 Flowchart of generating Hypertext dictionary.

用いたことである．**SGML** について簡単に説明する．**SGML** を用いて記述された文書は

- 文書内で用いられるシンタクスと文字セットを定義する **SGML** 宣言
- 文書の構造を定義する文書型宣言 (Document Type Definition，以後 DTD と省略)
- 個々の文書を表す文書インスタンス

から構成される．文書インスタンスは DTD に定義される構造に従って記述されるので，**SGML** を使用して複雑なデータを形式的に表現することが可能である．また，**SGML** は文書交換を目的としており，文書形式を変換するための多くのツールが開発されている．我々が使用した MARK-IT* は，Application Interface Language (以降 **AIL** と省略) と言う簡易言語を持っており，これを用いてハイパーテキスト化に必要なプログラムを開発した．

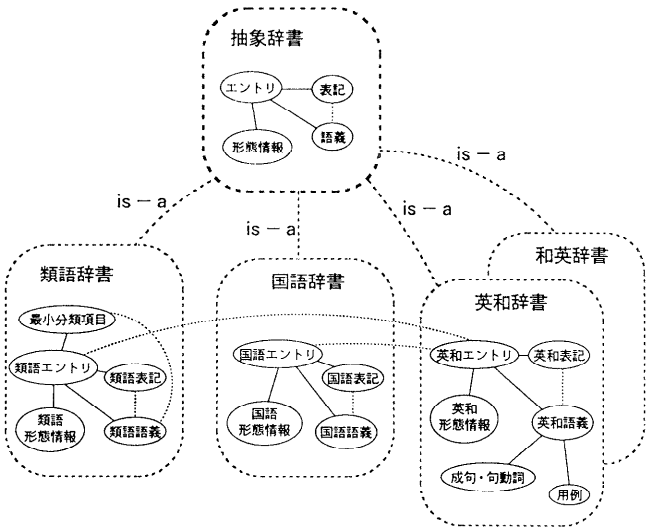


図 4 抽象辞書と各辞書との関係
Fig. 4 Relationship between abstract dictionary and concrete dictionaries.

4. ハイパーテキストデータモデル

本章ではデータ定義部作成フローにおいて用いるデータモデルについて述べる．

4.1 リンクの分類

ハイパーテキストは，文書の断片であるノードと，その間の関係を表すリンクにより文書を表現したものである．各ノード間に様々な関係が存在しているため，文書の構造を表現するには複数種類のリンクが必要である．

これまでにもいくつかの文献[3),4),9)]でリンクの分類が取り上げられているが，それらの多くはリンクの構造や意味だけでなく，ユーザとの対話法や実装法までを含む分類を行っている．例えば，文献 9) では九種類のリンクを挙げているが，その中にはリンク元のノードよりも詳細な情報を提示する Zoom links，リンク元のノードよりも高い視点からの情報を提示する Pan links，プログラムを起動する Execute links などがある．

そこで我々は，ハイパーテキスト構造の設計という観点から，必要なリンクの分類を行った．まず，リンクを大きく次の二つのタイプに分類する．ただし，リンクは個々のノード間の関係を表現するものとし，ノードのクラス間の継承関係を表す IS-A はリンクの分類より外した．

(a) **Organizational**：論理構造を表現するリンク．

(b) **Referential**：ノード間の意味的関係を表現するリンク．例えば，「右手」という単語と「左手」という単語の間にある“反対語”という関係を表す．

両者の違いは，(a)は，ノード間に従属関係を与えるが，(b)は従属関係を与えないことである．つまり，前者は，親ノードが削除されると，子ノードも削除されるが，後者は，子ノードとのリンク関係がなくなるだけで，子ノード自体は削除されない．

さらに Organizational リンクを次のように分類する．

(c) **Aggregational**：部分-全体関係を表現し，複合ノードを定義するリンク．例えば，「辞典の各エントリは，見出しと語義から成る」という関係を表す．

(d) **Grouping**：複数のノードの集まりであるノードを定義するリンク．例えば，「類語辞典の上位カテゴリは複数の下位カテゴリからなる」という関係を表す．

これらのリンクを用いることで，ハイパーテキスト構造が明確になり，また，データベースのスキーマの設計も容易になる．

* SEMA GROUP BELGIUM S.A. 社が商品化した SGML パーサ

4.2 データダイアグラム

当初我々は，ERダイアグラム[13),17)]を用いて辞書のモデル化を行っていた．しかし，ERダイアグラムには次の欠点がある．

(a) 複合ノードの明示的な表現ができない
(b) IS-A 関係が表現しにくい
(c) リンクの種類を明確に表現できない

これに対して，Coad らのダイアグラムは，Assembly Structure, Classification Structure によって上記の(a)と(b)を解決している．そこでこれをハイパーテキストの立場から整理し，4.1 節で述べたリンクの分類に基づいて拡張した．次に，拡張機能を中心に，ハイパーテキストデータモデルの図式表現（データダイアグラム）について述べる．

4.2.1 ノード

オブジェクト指向の考えにより，ノードの構造をクラスとして定義する．ノードクラスは次の二つの要素によって構成される．

(a) **class name**: ノードクラスを識別するための名前
(b) **attribute**: クラスに属するノードが持つ基本的なデータ

ノードクラスの表記法は，Coad らのオブジェクトの表記法を用いた（**図5**(a)）．ただし，ハイパーテキスト構造の設計には必要性がないためメソッドの記述は省いた．Coad らの記述法の拡張として，図5(b)のように抽象クラスを図5(c)のように各 attribute の多値性を表現した．抽象クラスは下位クラスを具体化するために使用されるもので，そのクラスのインスタンスは生成できない．ノードクラス間の継承関係を表す IS-A 関係は，Coad らの Classification Structure の記法を用いて**図6**のようにして記述する．

4.2.2 リンク

リンクは次の三つの要素によって構成される．

(a) **link name**: リンククラスを識別するための名前
(b) **type**: リンクのタイプ

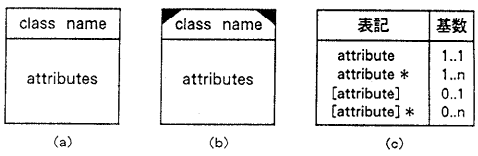


図 5 ノードクラスの表記法
Fig. 5 Description for node class.

(c) **foot**: リンクの多値性を表す写像基数

type は 4.1 節で述べたリンクの分類に対応する．type がリンクの構造に基づいた一般的なものであるのに対して，link name はアプリケーション固有の役割を反映したものをつける．リンクの意味を type と link name で表現することで，アプリケーションに共通の部分と特有の部分を分離することができる．

リンクの表記法を**図7**(a)に，type および foot の表記法を図7(b)に示す．特にハイパーテキストの特徴である，データの任意の部分とのリンク（埋め込みリンクと呼ぶ）を表すため，図7(c)のように埋め込みリンクが現れる attribute を記述する．

また，図7(d)のように attribute を付加されたリンクを記述する．リンクが attribute を持つ例には，外来語の原語を示す「原語」リンクがある．それは「言語名」と「綴り」の attribute を持ち，他の辞書を

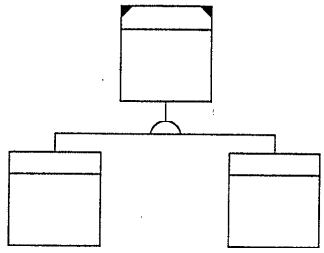


図 6 IS-A 関係の表記法
Fig. 6 Description for IS-A relationship.

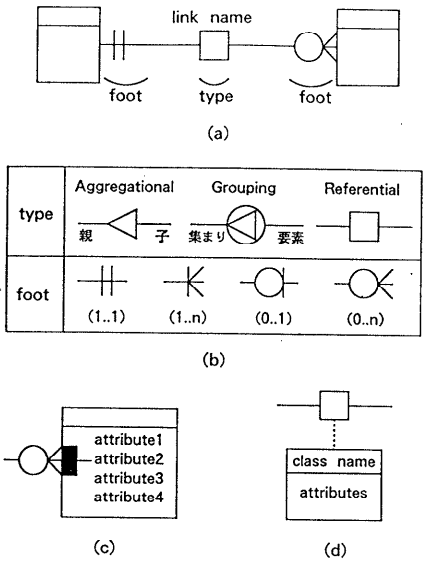


図 7 リンクの表記法
Fig. 7 Description for link.

検索するために使用される．

5. データ定義部作成フロー

本章では，データ定義部作成フローについて述べる．

5.1 構造分析・モデル化

ハイパーテキスト化手法では原データに含まれているハイパーテキスト構造を抽出するために原データの構造分析が必要である．構造分析は BNF を用いて行い，辞書の構文規則を明確にし，そこに含まれる項目を抽出する．辞書項目は構文的に判断できるものをさらに意味的な観点から詳細に分類する．この過程で辞書の構文的な曖昧さや意味的な曖昧さも明らかになる．**図 8** に国語辞書の構文規則を BNF で記述した例を挙げる．ただし，わかりやすさのため一部を省略してある．

次に BNF で記述された辞書の構造をベースに 4 章で述べたダイアグラムを用いてモデル化作業を行い，ハイパーテキスト構造を決定する．その際に，3 章で述べた抽象辞書をノード分割のテンプレートとして使用する．辞書項目のうち，他の辞書項目との関係を表すものは参照的なリンクと考えられる．例えば，“類義語” や “対義語” などをリンクとしてダイアグラムで表現する．

4 章で述べたダイアグラムを用いて新明解国語辞典をモデル化したダイアグラムを**図 9** に示す．ただし，構文的に抽出可能なものだけをリンクとした．

5.2 スキーマ定義の設計

構造分析・モデル化の結果を具体化し，ノードとリンクの形式的な定義を与えるステップが，スキーマ定義の設計である．この時，データダイアグラムで表された情報に実装上の情報が付加される．例えば，リンクを静的なリンクまたは動的なリンクとして実装するかを決定する．動的なリンクは，リンク先が前もって決まっていず，リンクを辿る時に始めて検索により求まるものである．動的なリンクは，リンク先が頻繁に変化するものや曖昧なものを表現するのに適している．スキーマ定義を設計する時，処理効率や記憶効率，使用する DBMS の制約などを総合的に判断して決定する．

スキーマ定義は，**図 10** のように，**SGML** のタグ表現を用いて表現する．本図で，“〈” から “〉” の部分がタグである．“〈” の後に続くトークンはタグ名で，タ

```
エントリ    := [マーク]見出し部[補足的説明]語義[数え方](子見出し部 語義)*
見出し部    := 見出し[ACC][歴史的かなづかい][正書法][品詞・活用][派生形]
語義        := (大語義並び|小語義並び)
大語義並び  := (大語義番号 ACCなど 小語義並び)
ACCなど     := [ACC][品詞活用][派生形]補足的説明*
補足的説明  := 〔 (語源など|位相など|注記など) 〕
小語義並び  := (小語義|(小語義番号 小語義)+)
小語義      := [ACCなど][反対]語義解説 補足的説明* 用例など* [反対][参照]
反対        := ↔ 反対語( ・ 反対語)*
参照        := ⇨ (参照語( ・ 参照語)*| 付表「 付表名 」)
```

図 8 BNF による国語辞書の構文記述の例
Fig. 8 BNF description of Japanese dictionary syntax.

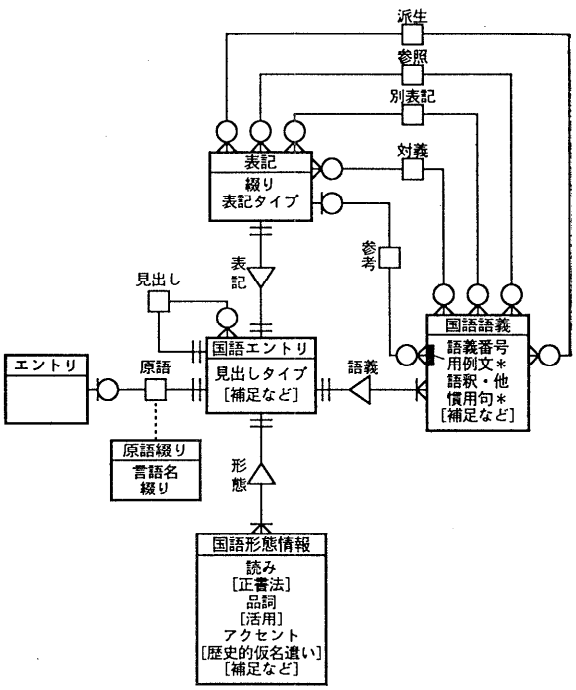


図 9 国語辞典のデータモデル
Fig. 9 Hypertext data model of Japanese dictionary.

```
<defschema> スキーマ名
   <isa>          上位スキーマ名
   <attribute>    アトリビュート名，データタイプ，
                  デフォルト値，最小基数，最大基数
      ......
   <link>         リンク名，リンクタイプ，
                  ターゲットスキーマ，最小基数，最大基
数
      ......
</defschema>
```

図 10 スキーマ定義のシンタクス
Fig. 10 Syntax of schema definition.

グのタイプを示す．“<”の後に“/”のないタグを開始タグと呼び，“/”があるものを終了タグと呼ぶ．開始タグと終了タグに挟まれた部分はタグによりマークされた項目（要素と呼ぶ）の内容である．

スキーマ定義では，ノードは「defschema」タグで，ノードの各 attributc は，「attributc」タグで記述する．attributeの多値性情報は最小基数と最大基数で記述し，それに加えて，attributeのタイプとデフォルト値（省略可能）を定義する．タイプは使用する DBMS のデータタイプ表記を用いる．ただし，**図 11**（b）の attribute「用例文」のタイプ定義「etext（同義 ref）」は，埋め込みリンクとして「同義 ref」リンクがテキスト中に現れることを示す．

リンクは「defschema」で記述されたノードの「link」タグで記述する．リンクの foot は，最小基数と最大基数で記述し，リンクタイプは，Oのようにタイプ名の先頭文字で表す．ターゲットスキーマはリンクの先となるノードのスキーマ名を表す．ただし，図11（b）の「参照 ref」のターゲットスキーマの定義「表記（綴り）」は，リンク先が「表記スキーマ」の「綴り」との比較によって求まることを意味する．図 11 は図9の「国語エントリ」と「国語語義」のスキーマ定義の

```
<defschema>   国語エントリ
    <isa>          エントリ
    <attribute>   見出しタイプ,string,,1,1
    <attribute>   補足など,string,,0,1
    <link>         子見出し,R,国語エントリ,0,*
    <link>         形態,0,国語形態情報,1,*
    <link>         語義,0,国語語義,1,*
    <link>         表記法,0,表記,1,*
    <link>         原語,原語綴り,エントリ,0,*
</defschema>
```

(a) 国語エントリスキーマ
(a) Japanese entry schema

```
<defschema>   国語語義
    <isa>          語義
    <attribute>   語義番号,int,1,1,1
    <attribute>   語釈・他,text,,0,1
    <attribute>   用例文,etext(同義ref),,1,*
    <attribute>   注記,text,,0,1
    <link>         参照ref,R,表記(綴り),0,*
    <link>         対義ref,R,表記(綴り),0,*
    <link>         同義ref,R,国語エントリ,0,*
</defschema>
```

(b) 国語語義スキーマ
(b) Japanese meaning schema

図 11 スキーマ定義の例
Fig. 11 Example of schema definition.

例である．

スキーマ定義は，5.4 節のデータベースのテーブル設計と，5.3 節のハイパーテキストデータの DTD 作成で用いられる．また，アプリケーションの開発の際に，データベースアクセス用のオブジェクトのクラス仕様を定義するために用いられる．

5.3 DTD の自動生成

スキーマ定義からハイパーテキストフォーマットデータの構造を定義する DTD を作成するためのステップが DTD の作成の自動生成である．このステップにおける処理の流れを**図 12** に示す．例えば，国語辞書の DTD を作成するためには，まずその上位クラスを含む抽象辞書と国語辞書のスキーマ定義を各々 DTD の記述形式に変換する．これら二つの DTD は独立では完全な DTD ではなく，これらを結合することによって完全な国語辞書 DTD が完成する．これらの処理も **AIL** を用いて行っているので，複雑な DTD を手作業で記述する必要はない．

5.4 データベースの設計

本ステップではスキーマ定義に基づいてデータベースの設計を行う．我々は RDB を用いているので，本ステップではテーブルの設計を行う．ノード，リンク，属性に対応して次のようなテーブルを作成する．テーブルを設計する際，設計ルールがあり，これに従うことでスキーマ定義より一意にテーブルを設計することができる．

- 各ノードクラスに対応して
  （a） ノード格納用テーブル
- 各リンククラスに対応して
  （b） リンク格納用テーブル
- 属性が多値属性の場合のみ
  （c） 多値属性格納用テーブル

（a）はノード番号フィールドを持ち，これによりノードの識別が行われる．ノード番号は各データベースごとに一意性を保たれる．（b）ではこのノード番号

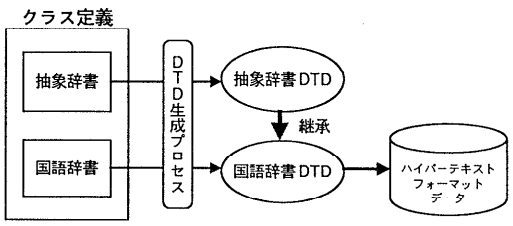


図 12 DTD 生成法
Fig. 12 Method of generating DTD.

を用いてノード間の関連付けを行う．(b)は，リンクタイプに対応して次の規則を用いて設計する．

- Organizational リンク用のテーブル作成規則（複数の親を取り得る場合のみ）
- Referential リンク用のテーブル作成規則
- Grouping リンク用のテーブル作成規則
- 外部のデータベースに対する Referential リンク用のテーブル作成規則

また，RDB に格納されたデータの問い合わせプログラムの開発を容易にするため，DB クラスライブラリが用意されている．DB クラスライブラリは SQL インタフェースを隠蔽し，オブジェクトを単位とする DB 問い合わせやリンクに基づいた検索を実現する．

6. データ変換フロー

本章ではデータ変換フローにおける各ステップについて述べる．

6.1 データの整形

2章で述べたように計算機可読辞書は多くの曖昧性や入力エラーを含んでおり，そのままでは計算機でハイパーテキスト化できない．そこで，**SGML** を用いた曖昧性のない表現（タグデータ）へと変換する必要がある．我々は，UNIX ツールのパターン検索・処理言語 awk やストリームエディタ scd を用いて原データのタグ付けを行った．しかし，原データには曖昧性や入力エラーなどがあり，正常に処理できない箇所があり，その部分はエディタで編集した．**図 13** は新明解国語辞書のデータの例であり，**図 14** は図 13 にタグ付けを行った例である．タグデータは，タグデータ自身の DTD に合致するか **SGML** パーサを用いて検証できるので，原データの持っていた記述の曖昧性は排除される．

本処理は，データ量も大きく，人手で処理しなければならない部分も多く，また，ハイパーテキスト化処理は試行錯誤を伴い，スキーマ定義が変更されることもあるので，コンバート処理をその後の処理から独立させている．しかし，今後 CD-ROM 等の普及に伴い，**SGML** などの文書記述言語によるデータ表現が一般的になると，このステップは不要になるであろう．

6.2 ハイパーテキスト化

本ステップでは，タグデータのタグ付けされた項目を，5.3 節で生成された DTD で定義された構造へ変換して，ハイパーテキストフォーマットデータを生成する．**図 15** は図 14 をハイパーテキスト化したハイパーテキストフォーマットデータである．図 15 のデータの構造は図 11 の定義に従っている．

ハイパーテキストフォーマットデータは，ノードを表す項目の内容に，属性またはリンクを表す項目が現

ひく◎【引く】㊀〔その物の一端をつかんで〕自分の手元へ近づける。「綱を—〔=引っ張る〕・弓を—〔=弦を引いて、射る〕・老人の手を—〔=誘導して前進させる〕・客を—〔=勧誘する〕・人の心を—〔=自分の方に向けさせる〕・かぜを—〔=かぜにかかる〕・息を—〔=吸い込む」↔押す ㊁ 多くの物の中から必要な物を（原形のまま）取り出す。「大根を—〔=引っ張って、土の中から抜く〕・例を—〔=引用する〕・辞書を——……
……

図 13 国語辞書の原データ[11]
Fig. 13 Part of original data of Japanese dictionary[11]

```
<一般の単語>
    <見出し>ひく
    <ACC>0
    <正書法>引く
    <語義番号>1
    <補足的説明>その物の一端をつかんで
    <語義文>自分の手元へ近づける．
    <用例等>綱を～〔=引っ張る〕・弓を……
    <反対語>押す
    <語義番号>2
    <語義文>多くの物の中から必要な物を……
    <用例等>大根を～……
        ……
</一般の単語>
```

図 14 国語辞書のタグデータの例
Fig. 14 Part of tagged data of Japanese dictionary.

```
<国語エントリ id='' ひく-1''>
  <形態>
    <国語形態情報>
      <読み>ひく
      <正書法>引く
      <ACC>0
  <語義>
    <国語語義>
      <語義番号>1
      <補足的説明>その物の一端をつかんで
      <語釈・他>自分の手元へ近づける．
      <用例文>綱を～<同義 ref ref='' ひっぱる-1''>引っ張る</>・弓を……
      <対義 ref>押す
    <国語語義>
      <語義番号>2
      <語釈・他>多くの物の中から必要な物を……
      <用例文>大根を～……
      ……
</国語エントリ>
```

図 15 ハイパーテキストフォーマットデータの例
Fig. 15 Part of hypertext format data of Japanese dictionary.

れるという，項目の入れ子構造として記述される．例えば，図 15 で，ノードを表す「国語語義」項目の内容に，attribute を表す「語義番号」項目と，リンクを表す「対義 ref」項目が現れる．属性は，それを表す項目の内容に値を書くことで記述される．リンクの表現はタイプにより異なる．Aggregational リンクの場合は，その内容にリンク先のノードを表す項目が現れる．例えば，「形態」項目の内容に，「国語形態情報」ノードが現れる．Referential リンクの場合は，項目属性「ref」にリンク先ノードの ID を指定し，その内容にリンク先をプレビューするためのデータを書く．例えば，同図の「同義 ref」リンクの先は ID が“ひっぱる-1”のノードで，「引っ張る」はプレビュー用のデータである．動的な Referential リンクの場合は，検索に用いられる値が項目の内容に現れる．例えば，「対義 ref」リンクの先は，「押す」を綴りに持つノードである．

ハイパーテキスト化処理を行うプログラムをトランスレータと呼ぶ．トランスレータの役割は次の三つである．

(a) データ構造の変換
(b) タグ名の変換
(c) リンク先を識別するための ID の付与

(a)は，タグデータとハイパーテキストフォーマットデータの間でデータの構造が変わる場合に行われる．例えば，図 14 の「一般の単語」項目はフラットな構造であるが，図 15 では，それが「形態」と「語義」の項目に分割される．また，ハイパーテキストフォーマットデータに対応する項目がない場合は，新しい項目の作成を行う．例えば，図 15 では，「国語形態情報」項目が新たに生成されている．

(b)は，タグデータとハイパーテキストフォーマットデータの間で対応するデータのタグ名が違う場合に行われる．例えば，図 14 の「見出し」タグは，図 15 では，「読み」タグに変換されている．

(c)は，リンクの先となるノードを識別するために，ノードの識別子を付与する処理である．図 15 の例では，「国語エントリ」の ID として“ひく-1”が付与されている．

これらの処理は，**SGML** パーサ MARK-IT の機能の，特定のタグが現れた時，そのタグ名に対応する **AIL** で記述された手続きを起動する機能を用いて実現している．

6.3 データベースへの格納

本ステップでは，ハイパーテキストフォーマットデータを 5.4 節で設計したデータベースに格納する．使用する DBMS のタイプ（例えば，RDB や OODB）によって本ステップの処理は異なる．我々は RDB を使用しているので，**AIL** で実現したジェネレータにより，ハイパーテキストフォーマットデータから SQL 文を生成し，その SQL 文を実行することでデータを RDB に格納している．

ハイパーテキストフォーマットデータのノードは，ノード番号が付加され，各ノードが 1 レコードとしてノード格納用テーブルに格納される．ノードの属性はレコードのフィールドにセットされる．ただし，属性が多値属性の場合は，多値属性格納用テーブルに格納される．

リンクの格納法はリンクタイプにより異なる．

Aggregational リンクの場合は，ノード格納用テーブルの各レコードの parent フィールドに，親ノードのノード番号をセットすることで表現される．Grouping リンクの場合は，各グループを識別するためのグループ格納テーブルが作成され，グループ番号とそれに属するノードの番号が格納される．

静的な Referential リンクの場合は，ハイパーテキストフォーマットデータの ID がノード番号に変換され，リンク元とリンク先のノード番号のペアーがリンク格納用テーブルに格納される．動的な Referential リンクの場合は，リンク元のノード番号と，ハイパーテキストフォーマットデータのリンクの内容に書かれたリンク検索用の値が格納される．また，埋め込みリンクの場合は，**SGML** のタグ表現がそのままテキストデータとして各属性を格納するフィールドに格納される．

7. おわりに

本論文ではハイパーテキスト化技術の大規模データへの適用手法の研究として，計算機可読辞書のハイパーテキスト化手法について述べた．本論文で述べた手法の特徴は次のとおりである．

- ハイパーテキストデータモデルによるハイパーテキスト構造の表現
- 抽象辞書の具体化による個々の辞書のモデル化
- **SGML** によるデータ記述と構造変換処理

本手法に基づいて辞書ハイパーテキスト Hydra (Hypertext Dictionary Reading Accessories) の試

作も行った[18]．今後，検討すべき点としては次が挙げられる．

- 他の辞書への適用実験に基づく手法の有効性の検証
- 辞書以外の一般文書への対応
- ハイパーテキストフォーマットデータの記述能力の向上

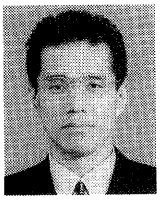
ハイパーテキストデータのフォーマットに関しては，その標準化案として ISO で **HyTime** (Hypermedia/Time-based Structuring Language) が検討されている．そこで，今回述べたハイパーテキストフォーマットを HyTime と融合させた記述形式の検討を行う予定である．

謝辞 本研究を進めるに当たりご指導いただいた情報システム研究所知能工学研究部の田村秀行部長，ならびに辞書データの分析にについてご協力いただいた情報システム研究所知能工学 13 研究室の皆さんに感謝いたします．

参 考 文 献

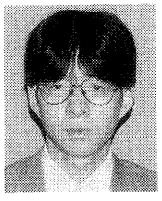
1) Coad, P. and Yourdon, E.: *Object-Oriented Analysis* (1st ed.), p. 232, Prentice-Hall (1990).
2) Conklin, J.: Hypertext: An Introduction and Survey, *IEEE Comput.*, Vol. 2, No. 9, pp. 17-14 (1987).
3) Conklin, J. and Begeman, M. L.: gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *ACM Transactions on Office Information Systems*, Vol. 6, No. 4, pp. 303-331 (1988).
4) DeRose, S. J.: Expanding the Notion of Links, *Hypertext '89 Proceedings*, pp. 249-257, ACM Press (1989).
5) Furuta, R., Plaisant, C. and Shneiderman, B.: *Automatically Transforming Regularly Structured Linear Documents into Hypertext*, Vol. 2, No. 4, pp. 211-229, Electronic Publishing (1989).
6) Glushko, R. J., Weaver, M. D., Coonan, T. A. and Lincoln, J. E.: Hypertext Engineering: Practical Methods for Creating a Compact Disc Encyclopedia, *The ACM Conference on Document Processing Systems*, pp. 11-20 (1988).
7) Hull, R. and King, R.: Semantic Database Modeling: Survey, Applications, and Research Issues, *ACM Comput. Surv.*, Vol. 19, No. 3, pp. 201-260 (1987).
8) ISO 8879, Standard Generalized Markup Language (SGML) (1986).
9) Parsaye, K., Chignell, M., Khoshafian, S. and Wong, H.: *Intelligent Databases*, p. 478, John Wiley & Sons (1989).
10) Raymond, D. R. and Tompa, F. W.: Hypertext and the Oxford English Dictionary, *Comm. ACM*, Vol. 31, No. 7, pp. 871-879 (1988).
11) 金田一京助，柴田 武，山田明雄：新明解国語辞典第二版，p. 1431，三省堂 (1974).
12) 黒橋禎夫，長尾 眞，佐藤理史，村上雅彦：専門用語辞典のハイパーテキストシステム，情報処理学会研究報告 [情報メディア]，91-IM-1-4 (1991).
13) 酒井博敬：情報資源管理の技法，p. 190，オーム社 (1987).
14) 島 健一：通信技術文書体系化システム—COSMOS，電子情報通信学会技術研究報告 [データ工学]，DE 89-29-35，pp. 49-56 (1989).
15) 鶴丸弘昭，内田 彰：国語辞典からの情報抽出と構造化について，長崎大学工学部研究報告，Vol. 15，No. 24，pp. 41-48 (1985).
16) 土井美和子，福井美佳，山口浩司，竹林洋一，岩井 勇：プレーンテキスト/ハイパーテキスト間の変換，情報処理学会研究報告 [情報学基礎]，89-FI-13-5 (1989).
17) マーチン，J.，マックルーア，C.：ソフトウェア構造化技法，p. 417，近代科学社 (1986).
18) 根本治朗，内藤広志，山下真司，松山洋一，柵木孝一：辞書ハイパーテキスト Hydra，情報処理学会研究報告 [データベース・システム]，91-DBS-86-4 (1991).

(平成4年 2 月 3 日受付)
(平成4年11月12日採録)



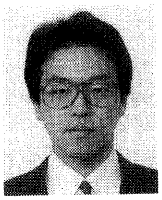
内藤 広志 (正会員)

1979 年東京大学教養学部基礎科学科卒業．同年安立電気(株)入社．1985 年キヤノン(株)入社．以来，知識ベースシステム，ハイパーメディアシステムなどの研究に従事．現在，システムエンジニアリングセンター製品システム第三開発部に勤務．ユーザインタフェース，オブジェクト指向設計に興味を持つ．ISO/IEC SG 18/WG 8 国内委員会委員．



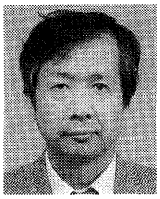
山下 真司

1988 年京都大学工学部精密工学科卒業．1990 年京都大学大学院工学研究科修士課程修了．同年キヤノン(株)に入社．現在はハイパーテキスト文書データベースの作成の研究に従事．



松山　洋一（正会員）

1962 年生．1985 年熊本大学工学部電気工学科卒業．1987 年同大学院工学研究科修士課程修了．同年キヤノン(株)に入社．グラフィカル・ユーザインタフェース，グループウェアに興味を持つ．



柵木　孝一（正会員）

1947 年生．1972 年北海道大学大学院工学研究科修士課程修了．同年，キヤノン(株)に入社．現在，同社システムエンジニアリングセンターにおいて，文書処理システムの開発に従事．人工知能学会会員．