

様々なオフィス内の情報を統一的に扱うデータベースのモデル：ブリッジモデル†

小沢英昭^{††} 安西祐一郎^{††} 相磯秀夫^{††}

オフィスオートメーション機器の普及と共に、オフィスでは情報の電子化が進んでいる。しかし紙に書かれた文書も多く存在し、電子的に作成された情報が有効に利用されている状態とはいえない。オフィス内の情報としては、文書のように構造化して扱うことが難しい情報や、在庫管理のように従来から構造化した情報として利用されてきた情報が混在して存在する。この2つの種類の情報を統一的な環境のもとで利用できなければ、情報全体を有効に利用することはできない。本論文ではオフィスにおける文書の作成、管理を支援するため、ハイパテキストとリレーショナルデータベースを融合するブリッジモデルを提案している。このモデルは、ハイパテキストへの集合の概念と関係演算の導入を基盤とし、両者のデータベースをすべて同一の関係演算を用いて処理できる。また集合論で用いられる Euler 図を応用したユーザインタフェースにより、両者のデータベース全体をハイパテキストにおけるリンクの関係とみなすことが可能である。さらに、ブリッジモデルに基づいたオフィスシステムを試作し、モデルの有効性の検討を行った。

1. はじめに

オフィスへの計算機の普及により、オフィス内の文書や各種の情報をデータベース化する動きが盛んである。既に名刺や商品の在庫等のデータは、リレーショナルデータベースなどにより、データベース化され広く利用されてきた。文書データベースとして文書のデータもデータベース化されつつあるが、名刺などのデータに比べて画一的な概念スキーマを定義して、構造化することは難しく、従来のデータベースでは取り扱い難かった。そこで近年文書の情報をリンクとノードを利用して緩く構造化し、情報を簡単に検索できるユーザインタフェースを持たせたハイパテキストが注目を集めている^{9),11)}。

ハイパテキストでは、データを格納するノードと、ノード間の関連づけを行うリンクを分けており、ノード中のデータを構造化して扱う概念を持たない。よって文書などの構造化し難い情報をノードに格納し、関連性については別の概念であるリンクを用いることで、ハイパテキストは、文書のデータベースを構築するのに適しているとされている。

オフィスには、文書などの構造化しにくいデータと名刺などの構造化しやすいデータの両者が混在しているため、両者を統合して扱うデータベースが必要である。そこで本論文では、オフィスにおける文書の作

成、管理を目的として、リレーショナルデータベース(以下 RDB) とハイパテキストを共存させつつ、両者を統一的に扱うデータベースモデルとして、ブリッジモデルを提案する。

このモデルではハイパテキストに集合の概念と関係演算を導入し、RDB とハイパテキストを統合化する。

さらに本論文では、統合化されたデータベースの検索のために、集合論で用いられている Euler 図を利用して、ハイパテキストの持つアイコンを手繰る検索と同等の機能を持つユーザインタフェースを実現し、データベースモデル中に統合化した結果について述べる。

2. ブリッジモデル

オフィスにおける構造化しやすい情報と構造化しにくい情報を、統一的に扱うために必要な機能として、大きく次の3つの機能が必要だと考えられる。

● 2つの種類の情報を別のデータベースとする

本質的に性質の異なる情報を、単一のデータベースとして扱うことには無理がある。そこで、それぞれの性質にあったデータベースに格納し、一段高いレベルで統一化を図るべきである。

● 構造化しにくい情報も一部を構造化する

文書にも、例えば手紙であれば宛名、メモであれば日付や人の名前等、部分的に構造化しやすい情報も含まれており、これらはユーザの必要に応じて、構造化して利用すべきである。人名や日付などの構造化された情報は、RDB 等のデータと関連づけて利用できる。

† Bridge Model: An Integrated Database Model for Office Information Systems by HIDEAKI OZAWA, YUICHIRO ANZAI and HIDEO AISO (Department of Electrical Engineering, Faculty of Science and Technology, Keio University).

†† 慶應義塾大学理工学部電気工学科

●共通のインタフェースを提供する

従来のデータベースで用いられてきた問い合わせ言語は、データベース検索を専門としないユーザには難しい。ハイパテキストと同一のインタフェースを提供して、簡単に検索できるようにすべきである。

この3つの機能を持つデータベースをモデル化すると、図1のようにデータ構造ごとに格納するデータベースが異なり、個々のデータベースを越えて、データ間の関係づけの機能とユーザインタフェースを持つモデルとして表現できる。このモデルは複数のデータベースを橋渡しして、統一的な環境をユーザに提供する。本論文では、このモデルをブリッジモデルと呼ぶ。

このブリッジモデルには、(1) RDB のようにデータの値を用いて、データ間の関連性を表現する機能と、(2)ハイパテキストのようにデータとは別に、関連性を表現する機構を、合わせ持つことが必要となる。ブリッジモデルでは、図1の RDB とハイパテキストのように、データの値を用いて検索条件を生成し、データ間を関係づける時には、両者を橋渡しする特別なノードを介して結ぶことにより、2つの関連性の表現の機構をユーザに、単一のリンクによる機構として提供する。この特別なノードをブリッジノードと呼ぶ。ブリッジノードは、ブリッジモデル上で検索条件を用いて、RDB 中のデータやハイパテキスト中のノードなどを検索するための、検索式を作成するノードである。

よってブリッジモデルには、2種類のリンクが必要である。1つはブリッジノードによって生成されるリンクであり、これを Dynamic Connection と呼ぶ。もう1つはハイパテキスト中のノード間を結ぶように、ノード上に明示的に示されたリンクであり、これ

を Static Connection と呼ぶ。この2種類のリンクを、以下のように定義する。

●Dynamic Connection

1. ハイパテキスト、および RDB 内のデータへの関係を、ブリッジノードを介して定義するリンク。
2. ブリッジノードは必要に応じて作成されるので、Dynamic Connection は、ノード上にリンクアイコンとして必ず表示される必要はない。
3. ブリッジノードがなくなると、ユーザからは見えない。

●Static Connection

1. Static Connection は、リンクが存在することをアイコンとして、ノード上に明示的に表現する。
2. ハイパテキストのノード間を関連づけているリンクと、Dynamic Connection をアイコン化したものの2種類がある。
3. Dynamic Connection をアイコン化して Static Connection に変換するオペレーションを、Freezing と呼ぶ。
4. ハイパテキスト内の任意のノードを関連づけているリンクは、リンクアイコンにノードへのポインタを保持しているが、Freezing によるリンクアイコンは、関連づけられるノードへの検索式を保持している。

このブリッジモデルを実現するには、以下の3点について検討することが必要である。まず Dynamic Connection を実現するために、1つめとして RDB のデータと関係づけるために、ハイパテキスト中のデータを部分的に構造化する手法であり、2つめとして部分的に構造化されたハイパテキストのデータと RDB のデータの間関係演算の定義である。そして3つめはブリッジノードにより、ユーザに対して1つのデータベースのように見せる、ユーザインタフェースである。この3つの要素について以下の各節で述べる。

3. ノードの構造化

3.1 RDB とハイパテキストのデータ

RDB では、各レコードはいくつかのフィールドから構成されていて、1つのデータ実体を表現している。ハイパテキストでは、ノードが一まとまりになったデータの実体を表現しており、レ

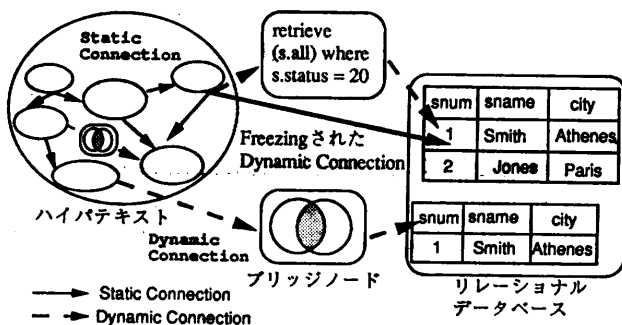


図1 ブリッジモデルによるデータベース
Fig. 1 A Bridge model: integrated database model.

コードとノードは同じレベルのデータとみなせる。
 このノードと RDB のデータ間で関係づけを行うには、ノードを部分的に構造化することが必要である。ここで例えば図 2 のノードを考えると、'KIDS' というリンクアイコンはノードの内容の一部でもあるし、また他のノードに対する関連性も表現している。RDB におけるフィールドが、レコードの値とレコード間の関係を表現していることを考えれば、この 'KIDS' という文字列は、RDB のフィールドと同じ働きをしている。

われわれはリンクアイコンを RDB のフィールドとの間で関係づけるために、リンクアイコンの概念を拡張し、これをリンクキーと呼び、次のように定義する。

リンクキー：ハイパテキストにおいて、任意のデータ間の関係を表現するためのキーワードである。ユーザインタフェースとしてはリンクアイコンであり、ユーザの必要に応じて作成される。リンクキーは名前と値を持ち、Static Connection を表現するリンクキーと、Dynamic Connection に用いられるリンクキーの 2 種類がある。

例えば図 2 には、'KIDS' と '国語辞典' の 2 つのリンクキーが存在する。

この 2 つのリンクキーは、性質が異なる。'KIDS' は Static Connection に対応するリンクキーで、HyperCard¹⁾ などのリンクアイコンと同じく、値として 'KIDS' というノードのハイパテキスト中でユニークな識別子を保持し、リンクキーの名として、'KIDS' と名づけられている。これに対して '国語辞典' は Dynamic Connection に用いられるリンクキーで、名として '辞書名' を持ち、値として '国語辞典' という文字列を持つ。

文書のデータを操作する時に必要となる属性は、個々のユーザによって異なるため、データベースの作成時に決定することはできない。そこでブリッジモデル

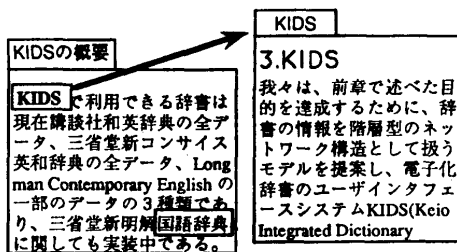


図 2 ハイパテキストのリンクアイコンの役割
 Fig. 2 Facilities of a link icon in the hypertext.

ルでは、リンクキーの作成は、ユーザによって任意に定義できることとした。

ハイパテキストのノードは、ノード中の情報全体で 1 つの意味を表しており、切り離すことはできない。そこでノードを RDB のレコードのように表現すると、ノードの情報全体を表す「Contents」と呼ぶフィールドと、いくつかのリンクキーのフィールドからなるものとする(図 3)。

3.2 ハイパテキストにおける集合の定義

ハイパテキストと RDB を関係演算によって関連づけるには、ハイパテキストに集合の概念を導入する必要がある。NLS⁶⁾ や Neptune⁴⁾ 等のように、リンクに意味を持たせることで、ノードをグルーピングできるシステムは多いが、集合の概念は定義されていない。

3.1 節で定義したリンクキーの値は、データの特徴の 1 つの側面を示している。例えば 8 月に作成した文書に対して、8 月というリンクキーの値を与えれば、一連のノードは 8 月の文書という 1 つのグループを形成する。そこで本論文では、ハイパテキストにおける集合を次のように定義する。

集合：ハイパテキストのある集合とは、いくつかの特定の名と値のリンクキーを持つノードの集まり。

ハイパテキストにおける集合に集合演算を行うには、UNION COMPATIBLE となる条件を定義しなければならない。ノード上のリンクキーは、ユーザの必要に応じて作成されるために、ノードごとにリンクキーの個数、名前は異なる可能性がある。そこでハイパドキュメントのノードに対しては、「Contents」のフィールドのみを対象として、UNION COMPATIBLE 可否かを定義する。

「Contents」は、ノード中のリンクキーの名か値が、含まれているので、「Contents」のフィールドの値のみで、主キーとなる。そしてハイパドキュメントのノードは、必ず「Contents」のフィールドを持つので、互いに UNION COMPATIBLE となる。

Contents	KIDS	辞書名
KIDS で利用できる辞書は現在講談社和英辞典の全データ、三省堂新コンサイス英和辞典の全データ、Longman Contemporary English の一部のデータの 3 種類であり、三省堂新明解国語辞典に関するも実装中である。	N2375	国語辞典

図 3 RDB のレコードとして表したノード
 Fig. 3 A Node is represented by the tuple expression.

4. ハイパテキストでの関係演算

4.1 関係演算の定義

4.1.1 伝統的な集合演算

3.1節で定義したリンクキーは、RDBのフィールドと同じなので UNION, INTERSECTION, MINUS は RDB の集合演算をそのまま継承できる。しかし TIMES については、3.2節でハイパテキストのノードの内容は、各ノードのフィールドとしてただ1つの「Contents」にあるとしたため、ノード間の直積により「Contents」が2つ存在することはできない。そこでブリッジモデルでは、文書と文書の間、および文書と RDB の間の、2種類の TIMES の定義が必要となる(図4)。

1. 文書と文書の場合

文書と文書の TIMES を行う場合、2つの Contents を合わせて1つの Contents を作成する。ノード中のリンクキーはそのまま、継承される。

2. 文書と RDB の場合

文書と RDB の直積は、Contents が無い文書と結合したのと同じであり、図4のように文書中の表のレコードが加わったハイパテキストのノードが生成される。

4.1.2 RDB の特別な関係演算

ハイパテキスト中のリンクキーは RDB のフィールドに相当するので、JOIN 等の関係演算をそのまま継承できる。ただし文書の直積を行った場合、2つの

ノードの「Contents」は結合されて1つになるために、「Contents」の DIVIDE を定義することができない。われわれは「Contents」の内容を関係演算を用いて操作するために、3.1節で提案したリンクキーに2つの機能を与えた。次節ではこのリンクキーの機能について述べる。

4.2 リンクキーの機能

4.1.1項において文書のデータに対する TIMES を定義したが、文書のデータの場合、単に2つの文書がつながるだけでは不十分である。例えば見積書などにつける挨拶の手紙は、日付や宛名、時候の挨拶などを変更して利用したいのであり、単にデータベースから検索した宛名が文書に連結されるだけでは無意味である。検索された宛名は、宛名を書く場所に挿入されねばならない。このためには「Contents」を、関係演算によって操作する以下の2つの型を定義する。

1. 任意データ型

任意データ型のリンクキーは、他の任意の型のデータと結合できる。これにより、文章中の穴に単語や単文などを埋める処理が、実現できる。例えば図5では、名簿から検索される5人に対して、宛名だけを変えた文書を送ろうとしている。

2. 定数データ型

定数データ型リンクキーは、リンクキーの値として任意の定数値を持ち、RDBのフィールドや他のリンクキーと JOIN することができる。

図6に示すように、特定の人だけに手紙を出すような

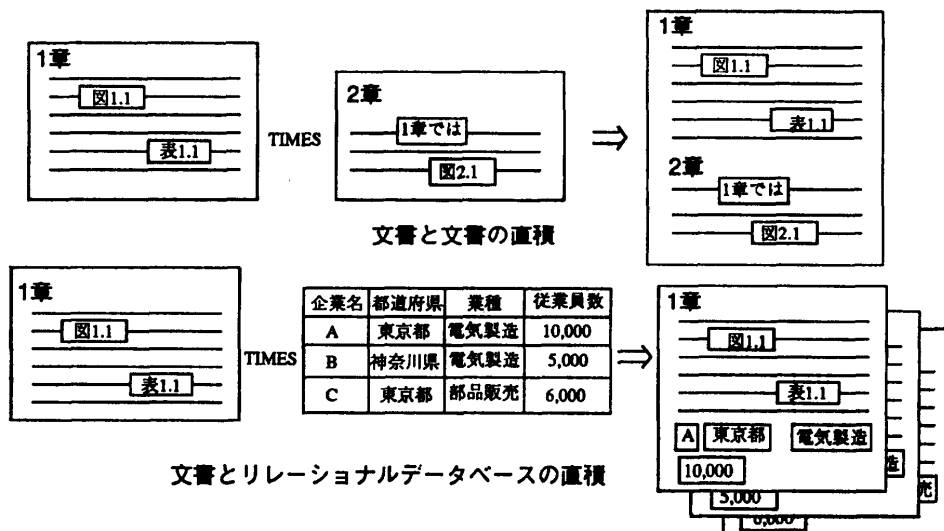


図4 文書のデータの直積

Fig. 4 TIMES operation of the documents in the Bridge model.

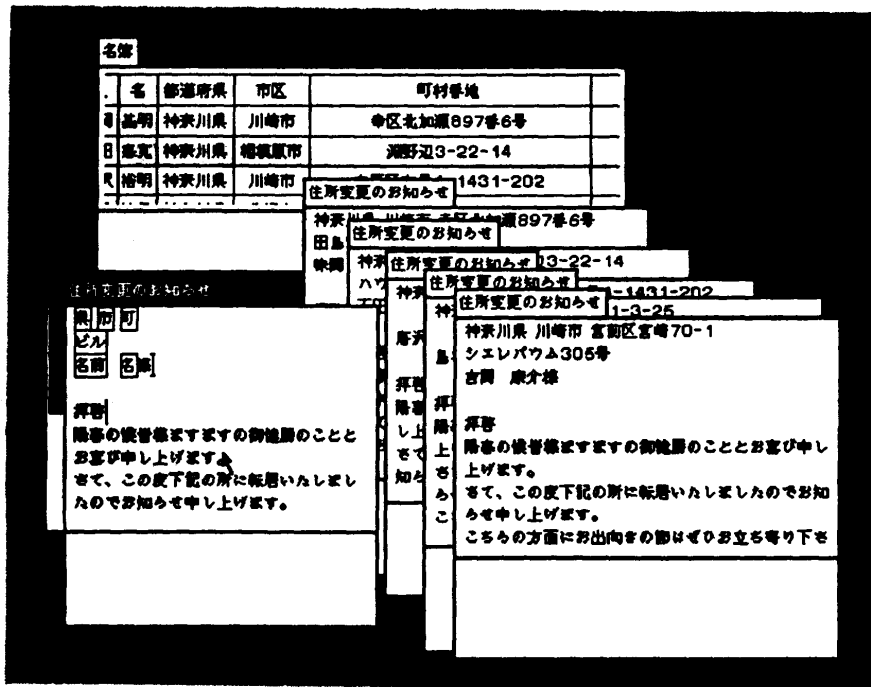


図 5 任意データ型リンクキーによる文書と住所 DB の直積
Fig. 5 TIMES between the hypertext and the RDB with null data link-keys.

場合には、送りたい人の名前を定数データ型のリンクキーとし、住所などの入る部分を任意データ型のリンクキーとすることで、新しい手紙を作成できる。リンクキーの2つの機能は、関係演算を用いてノード中のデータを加工、変更する能力を実現した。

これにより 4.1.2 項では定義できなかった、「Contents」の DIVIDE も実現できる。例えば図 7 で 1 章だけを切り出すには、2 章を任意データ型リンクキーとして、「Contents」が空のノードと結合することで達成できる。

リンクキーの機能により、「Contents」のデータを含めハイパードキュメントのノード、ならびに RDB のデータは、すべて関係演算を用いて処理することができた。この結果 Dynamic Connection を、ブリッジモデル全体で定義できる。

5. ブリッジノードのインタフェース

ブリッジモデルをユーザインタフェース面で統一化するためには、ブリッジノード上で Dynamic Connection を作成する特別なユーザインタフェースを、提供することが必要である。そこで検索式の例として、QUEL で書かれた検索式を考える。

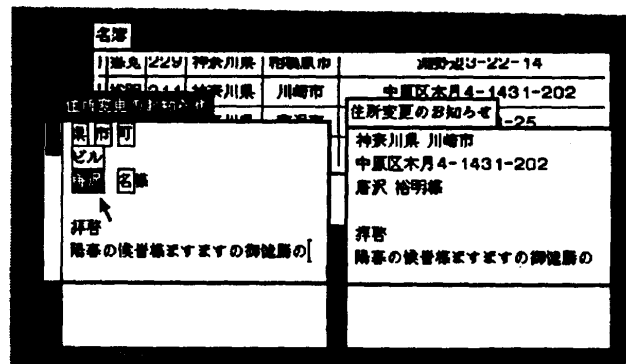


図 6 定数データ型リンクキーによる文書と住所 DB の JOIN
Fig. 6 Natural join between the hypertext and the RDB with constant data link-key.

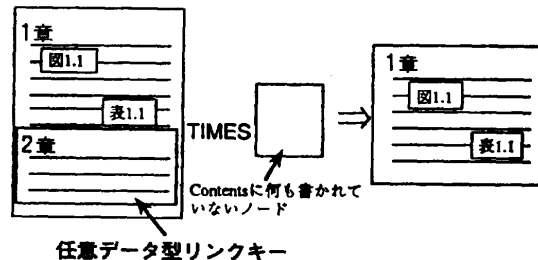


図 7 Contents のデータの分離
Fig. 7 DIVIDE of the documents in the Bridge model.

range of n is 名刺
 retrieve (n , all) where n . 都道府県="東京都"
 and n . 会社名="日本"

(1)

データベースのアクセス法の戦略には、いくつかの方法があるが、ここでは次のように分解する。

1. フィールド名と値との関係づけ
2. フィールド名と値のペアの論理演算
3. 結果のテーブルの PROJECTION

このうち2の処理は論理演算の組合せであり、とりうる可能性の幅が非常に大きい。例えば3つの条件があったとすると、すべての組合せは約 30 個にもなる。

われわれは、論理演算によって得られるデータベースの一部分をユーザが直観的に判断できるように、集合論で用いられる Euler 図による表現を導入する。Euler 図では式(1)の AND 条件は、図 8 のように2つの円の重なっている部分として、直接的に表現できる。2次元の Euler 図1つでは、最大3個までの要素の関係しか表現できないが、複数枚の Euler 図を組み合わせることで、任意の論理演算の関係を表現することが可能である。

Euler 図上での論理式の表現は、検索式中に含まれる AND, OR などの論理演算子が、Euler 図の各領域に対応する、いくつかのリンクアイコンに置き換わったものとみなせる。よってユーザは、Euler 図上での領域の選択を変えることによって、論理演算の条件を変えながら、データベースをブラウジングできる。

Euler 図は、多数のリンクアイコンを集約しているのと同じであり、ブリッジノードは、ノード上のすべての情報がリンクアイコンからなるノードといえる。

6. 他の研究との比較

文書をデータベース化する研究は盛んである。タイトルや著者名などの構造化しやすい部分の情報を利用して、文書を検索したり、関係演算の定義を行い、文書のデータベース化を行う手法^{9),10),11)}が、提案されている。

しかし例えば受けとった文書をファイルにして、定型的文章を再利用するように、ユーザの要求に応じて、文書の内容の任意の部分を利用できるように、データベース化することが必要である。本論文で提案したリンクキーによる構造化の手法は、ユーザの要求に応じた文書の構造化の機能を提供することができ、ユーザの要求に応じたデータ間の関連づけを行える。

さらに本論文で提案したブリッジモデルは、ハイパテキストによって構築された文書データベース内で文書の検索を支援するだけでなく、複数のデータベースに格納されている情報を、仮想的に単一のデータベースとして提供することができた。

データベースのユーザインタフェースとして、ハイパテキストのインタフェースを取り入れた Text-Link¹²⁾などのシステムもあるが、これらのシステムではリンクの作成にはハイパテキストと別の概念を用いている。例えば TextLink では、問い合わせ言語を

用いて記述する。ブリッジモデルでは、Dynamic Connection の作成でもハイパテキストのリンクアイコンと同じようなインタフェースを提供し、ユーザインタフェース性の向上を図っている。

結局ブリッジモデルは、ハイパテキスト中のノードのデータを、ユーザの指示により構造化し、RDB 中のデータと関係演算を用いて、同一の検索方法で検索できることと、検索条件をブリッジノードにより図的に表現し、Freezing により検索条件をアイコン化できることにより、ハイパテキスト、RDB に

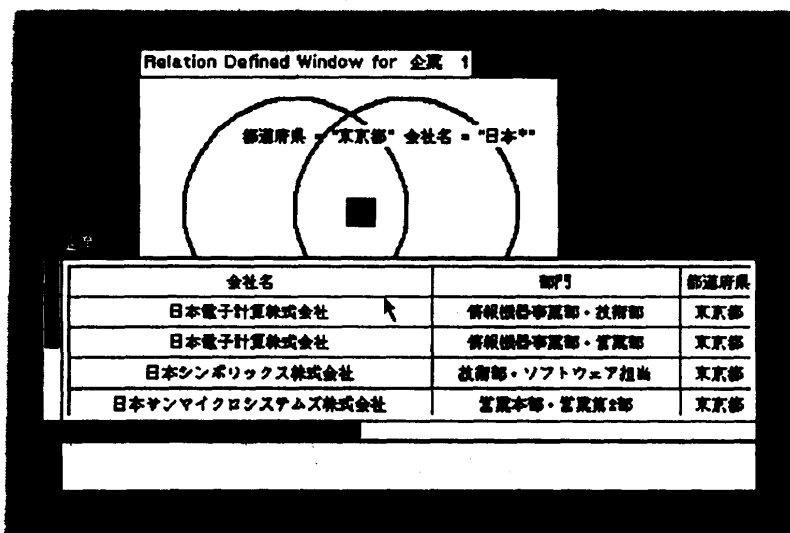


図 8 Euler 図によるデータベースの検索
 Fig. 8 Database retrieval with Euler graph.

よるデータベース全体を、統一的なデータモデルとして表現することができた。

7. オフィスシステムへの応用

7.1 オフィスにおける情報の利用法

オフィスには、メモのように構造化しにくい情報とカレンダーや住所録のように構造化しやすい情報が混在しているが、われわれはそのようなことを意識していない。われわれは無意識に構造化しやすい情報は構造化して、構造化しにくい情報は構造化せずに利用している。

ブリッジモデルは、RDB とハイパテキストを単一のデータベースとして扱えるので、構造化しやすい住所録やカレンダーなどの情報を RDB に、メモや電子メールなどの情報をハイパテキストで管理しつつ、容易に両者を組み合わせたシステムを実現できる。

7.2 ブリッジモデルによる実現

われわれは RDB の部分に Ingres[®]を用い、ハイパテキストのデータベースは Smalltalk-80 を用いて、Smalltalk-80 上にブリッジモデルに基づいたオフィスシステムのプロトタイプを実装した^{12),14)}。

このシステムは、ハイパテキストを中心として、カレンダーや住所録などのデータベース、辞書のデータベース、電子メールといった各種のシステムを統一的なユーザインタフェースのもとで提供している (図9)。

例えば図9では、予定がカレンダー上にリンクアイコンとして表現され、電子メールとの間で関連づけられている。各種のデータベースの情報は図のように、ブリッジノードを介してハイパテキストのノードや電子メールのノード等と関連性を持つことができる。また図ではカレンダーに貼り付けられた「電子協」というメモから、電子協というリンクキーと、部門の情報からブリッジノードを利用して、データの検索を行っている。何回もこのデータを参照するのであれば、ブリッジノードによって作成された Dynamic Connection は、2章で述べた Freezing により、リンクアイコンとして各ノード上に貼り付けることもできる。図では、中央下にある「サンマイクロ」というリンクアイコンが、Freezing された Dynamic Connection である。

われわれはブリッジモデルにより、構造化しやすい

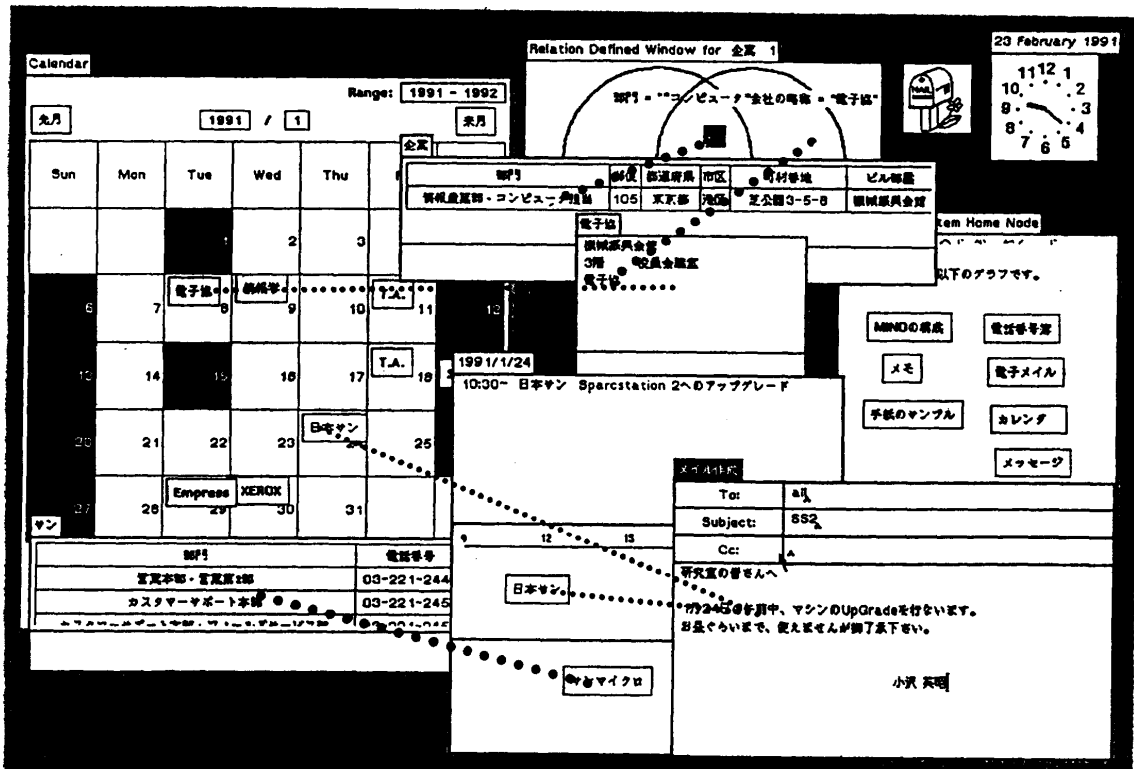


図9 ブリッジモデルによるオフィスシステム
Fig. 9 Office system which is based on the Bridge model.

情報は RDB として格納し、構造化しにくい情報はハイパテキストとして格納し、さらに統一的なユーザインタフェースを提供した。この結果われわれが日常利用している、構造化された情報と構造化されていない情報が混在する環境を実現することができた。

7.3 ま と め

本論文で作成したプロトタイプに最も近いシステムとしては、OBE¹⁵⁾ があげられる。OBE の場合、作成される文書は RDB 中のデータの単なるビューであり、せっかく作成された文書をデータベース化して再利用することはできない。

作成される文書もデータの 1 種であることを考えれば、データベース化して、RDB 中のデータと統一的に利用されるべきである。本システムでは、作成された文書はハイパテキスト中のノードとなり、ブリッジモデル上のデータベースの情報として利用できる。

8. お わ り に

計算機上での文書の作成システムは、近い将来、より幅広い文書や情報の作成管理システムになると考えられる。本論文のシステムはこのような状況の下で、文書の作成と利用を一元的に行える。

本の索引や、図書館の目録などに代表されるように、大規模な情報を効率的にハンドリングするためには、情報の構造化の問題を避けて通ることができない。オフィスにおける文書を取り扱うには、構造化が緩いデータベースと厳密に構造化したデータベースの両者を融合することが必要である。

本論文では、文書を単なる文字列としてではなく、ユーザの意思によって構造化されるデータだと考え、ハイパテキストと RDB を、仮想的に単一のデータベースとしてみなすブリッジモデルを提案した。

ブリッジモデルではデータベースを、Euler 図を用いて作成した Dynamic Connection と、ハイパテキストのリンクと同じように、情報間の関係を直接的に表現する Static Connection の 2 種類のリンクで関係づける。この結果ユーザインタフェース上、仮想的に単一のデータベースが実現できた。

さらに本論文では、オフィス内で情報処理にブリッジモデルが有効なことを示した。われわれは日常無意識に、構造化されたデータと構造化されていないデータを、混合して利用している。われわれが提案したブリッジモデルでは、カレンダーの例のように、構造化しやすい部分と構造化し難い部分を独立に扱いながら、

統一的な検索手法とユーザインタフェースにより、互いに関連づけて利用することが可能となった。

現在の実装では、われわれが紙の文書をファイルするように、ファイルの仕方の指示に相当するリンクキーの設定を、すべてユーザに任せている。この結果リンクキーの設定の仕方により、検索しにくいノードが生じる可能性がある。そこで今後の課題としては、情報の再利用を優先したいユーザに対して、よりシステムティックにハイパテキストの検索をさせることを、考慮すべきである。これには文書を完全に構造化されたデータベースとする⁷⁾や、ルールベースを用いた検索法を試みている⁵⁾などの手法についても、取り入れていくべきだと考えている。

謝辞 本研究を進めるにあたり、実装などで協力をいただいた松本茂君と東樹康子さんならびに、電気工学科安西研究室の皆さんに感謝します。

参 考 文 献

- 1) Apple Computer Inc.: HyperCard Reference Manual, Apple Computer Inc. (1988).
- 2) Bertino, E., Rabitti, F. and Gibbs, S.: Document Query Processing Strategies: Cost Evaluation and Heuristics, *Conference on Office Information Systems*, pp. 169-181 (1988).
- 3) Conklin, J.: Hypertext: An Introduction and Survey, *IEEE Comput.*, Vol. 20, No. 9, pp. 17-41 (1987).
- 4) Delsile, N. and Schwartz, M.: Neptune: a Hypertext System for CAD Applications, *ACM SIGMOD '86*, pp. 132-143 (1986).
- 5) 遠藤 勉, 宇津宮孝一, 藤田米春, 二村祥一, 吉田和幸: マルチメディア文書の知的処理システムに関する研究, *九州産業技術*, No. 12, pp. 19-38 (Dec. 1989).
- 6) Engelbert, D.C. and English, W.K.: A Research Center for Augmenting Human Intellect, *AFIPS*, Vol. 33, No. 1, pp. 395-410 (1968).
- 7) Güting, R.H., Zicari, R. and Choy, D.M.: An Algebra for Structured Office DOCUMENTS, *ACM TOIS*, Vol. 7, No. 2, pp. 123-157 (1989).
- 8) Held, G.D., Stonebraker, M.R. and Wong, E.: INGRES—A Relational Data Base System, *NCC 1975*, pp. 409-416 (1975).
- 9) 京嶋仁樹, Gopal, N., 上林憲行: 文書プログラミングについての一考察, *信学技報*, OS 88-53, pp. 27-35 (1989).
- 10) Lai, K.Y., Malone, T.W. and Yu, K.C.: Object Lens: A "Spreadsheet" for Cooperative Work, *ACM Trans. Office Inf. Syst.*, Vol. 6,

No. 4, pp. 332-353 (1988).

- 11) Nelson, T. H. : Replacing the Printed Word : A Complete Literary System, *IFIP Proc.*, pp. 1013-1023 (1980).
- 12) 小沢英昭, 松本 茂, 安西祐一郎, 相磯秀夫: ハイパーテキストの拡張によるオフィスシステムへの応用, 1991年情報処理学会情報学シンポジウム講演論文集, pp. 137-146 (1991).
- 13) 田中克己, 西川記史: オブジェクト指向データベースのハイパーテキスト型インタフェース TextLink について, 情報処理学会研究報告, 89-DBS-72, pp. 65-72 (1989).
- 14) 東樹康子, 松本 茂, 小沢英昭, 安西祐一郎: ハイパーテキストを応用したメールシステム, 第40回情報処理学会全国大会論文集, pp. 594-595 (1990).
- 15) Zloof, M. M. : Office-by-Example : A Business Language That Unifies Data and Word Processing and Electronic Mail, *IBM Syst. J.*, Vol. 21, No. 3, pp. 272-304 (1982).

(平成3年4月3日受付)

(平成4年2月14日採録)



小沢 英昭 (正会員)

昭和61年慶應義塾大学工学部電気工学科卒業。平成3年同大学院博士課程単位取得退学。現在日本電信電話株式会社ヒューマンインタフェース研究所勤務。マンマシンインタフェース, データベース, 文書処理などに興味を持っている。ACM 会員。



安西祐一郎 (正会員)

昭和49年慶應義塾大学大学院博士課程修了。工学博士。昭和56~57年カーネギーメロン大学客員助教授。慶應義塾大学講師, 北海道大学文学部行動科学科助教授を経て, 昭和63年慶應義塾大学理工学部電気工学科教授。計算機科学, 人工知能, 認識の情報処理過程の研究に従事。著書「知識と表象」(産業図書)ほか。電子情報通信学会, 日本ソフトウェア科学会, ACM 各会員。



相磯 秀夫 (正会員)

昭和7年生。昭和30年度慶應義塾大学工学部電気工学科卒業。昭和32年同大学院修士課程修了。大阪大学工学部助手を経て, 通商産業省工業技術院電気試験所(電子技術総合研究所)に勤務。トランジスタ計算機の研究開発に従事。その間昭和35年から1年半米国イリノイ大学計算機研究所に留学。昭和46年慶應義塾大学工学部電気工学科教授, 平成2年慶應義塾大学環境情報学部学部長。工学博士。著書「計算機アーキテクチャ」(岩波講座)ほか。電子情報通信学会, IEEE, ACM 各会員。