

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

TextLink: A Hypertext-Based Interface for Object-Oriented Databases

田中 克己
Katsumi TANAKA

西川 記史
Norifumi NISHIKAWA

神戸大学工学部
Faculty of Engineering, Kobe University

あらまし：本稿では、オブジェクト指向データベースのユーザーインタフェースとして、ハイパーテキストの概念を取り入れて我々が開発し、現在稼動中である TextLink-II システムについて述べる。TextLink-II の開発目的は、(a) オブジェクト指向データベースのユーザーインタフェースとしてのハイパーテキスト機能の利用方式の検討、(b) ハイパーテキストのリンク情報の更新とデータベースの更新との間の独立性を高める機能の開発である。このために、TextLink-II システムの中で導入された特徴的な機能は、(1) 実行可能リンク、(2) リンク情報の波及、及び (3) ハイパーテキスト型インタフェース上でのデータベーススキーマ・オブジェクトの更新機能の3つである。さらに我々は現在、実行可能リンクという概念をさらに一般化して、オブジェクト指向データベース上に仮想的にハイパーテキスト機能を実現するためのリンク定義言語 (Link Definition Language) LDL を開発中であり、これについても報告する。

Abstract: In this paper, we describe our TextLink-II system that was developed and is currently running as a hypertext-based user interface of our object-oriented database system. The purposes of developing the TextLink-II are: (a) to investigate how the hypertext concept can be utilized as a user interface of object-oriented DBMSs, and (b) to develop facilities to increase the independence between hypertext-link updates and database updates. For these purposes, the TextLink-II system introduces (1) *executable link*, (2) *propagation of linking information*, (3) *updates of DB schemata and DB objects through the hypertext-based user interface*. Furthermore, we will describe the *Link Definition Language (LDL)*, that we are now developing in order to realize virtually the hypertext facility over object-oriented databases.

1 まえがき

本論文では、オブジェクト指向データベース [1][2][3] のユーザインタフェースとして、ハイパーテキストの概念を取り入れて我々が開発した TextLink-II システムについて述べる。このシステムは、また逆に、属性値情報を有するテキストオブジェクト群からなるハイパーテキストシステムに対してデータベース機能を追加したシステムという位置づけも可能である。TextLink-II システムの開発の目的は、(a) オブジェクト指向データベースのユーザインタフェースとしてのハイパーテキスト機能の利用方式の検討、(b) ハイパーテキストのリンク情報の更新とデータベースの更新との間の独立性を高める機能の開発である。このために、TextLink-II システムの中で導入された特徴的な機能は次の3つである。

1. 実行可能リンク (executable link)
2. リンクの波及機能

3. ハイパーテキスト型インタフェース上でのデータベーススキーマ・オブジェクトの更新機能

さらに我々は現在、実行可能リンクという概念をさらに一般化して、オブジェクト指向データベースの上に仮想的にハイパーテキスト機能を実現するためのリンク定義言語 (Link Definition Language) LDL を開発中であり、これについても報告する。

2 Hypertext とオブジェクト指向データベース

Hypertext は、コンピュータ・プログラムや文書などのテキストオブジェクトどうしを種々の参照関係を表わすリンクによって非線形な形に関連づけたものであり、計算機化された Hypertext の利点として以下のものが考えられる。

- リンクに基づいたテキストオブジェクト間の navi-

gation や browsing が可能。

- 参照情報（リンク）は、テキストオブジェクト中に保持されるので、テキストの構成順序が変化しても、リンクは影響を受けない。

しかし、Hypertext 自身、従来より次のような欠点も指摘されている。

- 非線形テキストでは、位置や方向の感覚を、失いやすい。
- 複数のウィンドウを同時に開き、平行して仕事をするとユーザの認識限界を越えることがある。

これらの欠点を克服することなどを目的として、通常、リンクの類別化や多様なブラウザが提供されている。さらに、次のようなデータベース的な機能を持ったものも多数開発されている [4]。

- テキストオブジェクトやリンクへの属性値情報の付与や、述語ベースの質問言語機能 (Neptune, NoteCards など)
- テキストオブジェクトやリンクの選択（フィルタリング）機能 (Intermedia の active web, NLS/Augment のビューフィルターなど)
- 分散型・マルチユーザアクセス機能 (Neptune や Intermedia)
- テキストオブジェクトのバージョン機能 (Neptune など)
- テキストオブジェクトへの手続きの付与 (KMS, Neptune, NoteCards など)

この中でもブラウン大学の Intermedia[5][6] は主にリンク情報の格納・管理のために関係型 DBMS である Ingres を用いており、Hypertext と汎用 DBMS の融合という観点から興味深い。しかし、このような形での Hypertext と汎用 DBMS の結合では依然次のような欠点が存在する。

- テキストオブジェクト間リンクが直接的・固定的であり、テキストオブジェクトの内容が頻繁に変化する場合、リンクの更新が煩雑・複雑になる
- 関係データベースやオブジェクト指向データベースが提供している強力な集合型データ操作言語を利用できるものはない。

我々の TextLink-II システムでは、この 1 つ目の問題に対しては、リンク先のオブジェクト群が固定的にリンクされるのではなく、データベース質問という形で表現される、実行可能リンク (executable link)(文献 [7] では「動的リンク」) を実現して対処している。また、2 つ目の問題に対しては、基本的な集合型操作言語を Smalltalk-80 で実現している。

一方、オブジェクト指向データベースにおいては、複合オブジェクト (complex object) のように、基本的なデー

タの処理単位であるオブジェクト自身の規模が大きくなり、単にオブジェクト集合を検索する機能だけでは不十分であり、検索された個々のオブジェクト内を Hypertext 的に navigation できる機能があれば有用である。しかも、複合オブジェクトを実際にユーザインタフェース上でいかなる形で表示するかといった新しい問題もある。Maier らは 1 つの複合オブジェクトの表示に入れ子構造を持ったウィンドウを用いるという方式を提案している。また、最近 Altair の Plateau らは複合オブジェクトの表示方式としてオブジェクトのアイコンと入れ子構造型ウィンドウや複数ウィンドウのグラフ表現を組み合わせたユーザインタフェース LOOKS Hyper-Object System をオブジェクト指向 DBMS O_2 上で開発している。ここではこれらの方式とは異なり、基本的に 1 つのオブジェクトは 1 つのテキストで表示するという立場を取る。

3 TextLink-II システムの概要

本節では、我々が開発した TextLink-II システムの概要について述べる。

3.1 概念設計

TextLink-II システムは、ユーザがデータベースへのアクセス・構築等を簡単に行えるよう開発されたシステムであり、データベースの構造および操作言語を知らなくともデータベースの構築・検索が可能である。実際には、

1. ユーザは 1 枚のテキストを通じてデータベースおよびオブジェクトの更新、検索を行う。この更新は属性値の更新の他、属性追加や新規クラス生成などのスキーマ更新が行える。検索における結果もまたウィンドウを通じて表示される。
2. 各操作は大部分がマウスによって行われる。このためメニューに統一性をもたせることが可能である。
3. ユーザが直接データベースに働きかけることが可能である。このため既存の検索の他にユーザ独自のオペレーションをもつことが可能。
4. データベースの構築・更新、オブジェクトの作成が大部分マウスを用いた対話的・視覚的な方法で行える。

次に本システムの主要な機能について、その概要を説明する。

3.1.1 固定リンクと実行可能リンク

固定リンクとは、表示されているテキスト中の任意のテキストセグメントから特定のデータベースオブジェクトを直接参照するリンクのことである。このリンクは、ユーザが自由に設定、削除が可能であり、削除されない限りそのまま保持される。このリンクは、ポインタと同じものと考えてよい。リンクを開く場合は、リンク先のデータベースオブジェクトが新しいウィンドウ内にテ

キスト形式で表示される。

しかし、固定リンクには種々の問題点が存在する。このリンクはデータベース内に設定されているが、データベースは本来少しづつ更新されていくものである。データベースが更新されれば、そのリンク先のオブジェクトも変化するが、システムが巨大になりリンクの数が膨大なものになってくると、リンクをテキストセグメント・データベースオブジェクト間に固定していたのではその更新が非常に煩雑なものになってしまう。一方、リンク先のオブジェクトはなにかある特徴を持っており、ユーザはそれを手がかりとしてリンクを設定しているわけであり、オブジェクトがその特徴を満たせば自動的にリンクを設定してくれることを望むはずである。そこで我々は、データベースの更新に対してこの特徴をもとに自動的にリンク先を更新する実行可能リンクの概念を導入した。

実行可能リンクとは、固定リンクと異なりリンク先がオブジェクトではなくデータベースに対する検索質問になっている。ユーザは、ここにデータベースに対する検索質問を記述することによってリンクを設定するわけである。そしてここに記述された検索質問はリンク参照時に初めて実行され、データベース内から質問の条件を満たすオブジェクトすべてを集め、リンク先オブジェクトとして表示するわけである。すなわち、リンクは利用するたびごとに設定されることになり、データベースの更新による影響を受けなくなる。これは、実行速度の点で固定リンクに劣るが、ユーザがデータベースの更新ごとにリンクのメンテナンスを行う必要がなくなるので、勝っていると思われる（このシステムは、データベースの更新・リンクの設定等もユーザが行うことを前提としている）。このリンクを Fig3.1 に示す。

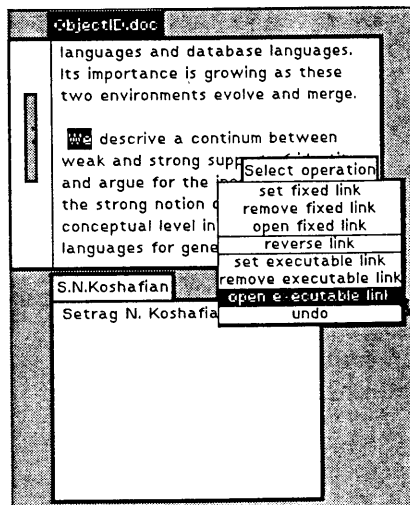


Fig3.1 実行可能リンク

3.1.2 リンクの波及

ユーザが文献等を読んでいてリンクを設定する場合、通常はリンクを一つ一つ手で設定しなければならないが、

もしユーザがある範囲に属するオブジェクト（文献）のあるテキストセグメントからはすべて同一のオブジェクトに対してリンクを設定されるということを事前に知っているのであれば、それらのオブジェクト一つ一つにリンクの設定を行うのは、ユーザの負担を軽減する上であまり望ましい行為であるとはいえない。

例えば、クラス OODBDocuments とそのサブクラスに属するすべての文献の object identity という言葉からリンクを設定したいとする。このとき、このクラスに属するすべてのオブジェクト一つ一つにリンクを設定することは、非常につらい作業となる。そこである一つのオブジェクトからリンクを設定する際、その範囲を指定することで（この場合ではクラス OODBDocuments とそのサブクラスに属するすべてのオブジェクト）その範囲内の全オブジェクトにリンクを設定する機能である。このようなリンクの設定方法をリンクの波及という。この概念図を Fig3.2 に示す。

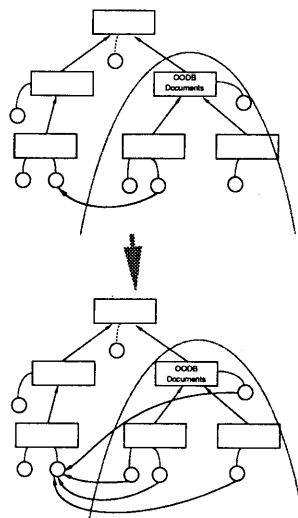


Fig3.2 リンクの波及

3.1.3 データベース更新機能

ユーザがデータベースのインタフェースとしてシステムを使用する場合、データベースおよびオブジェクトの作成・更新機能はそのシステムに必須の機能である。

本論文では、データベースとしてオブジェクト指向データベースを対象としているが、ここでのデータベーススキーマの更新とはオブジェクトの他のクラスへの移動及びクラス階層の変更・更新のことを指す（この機能は固定リンク設定時にリンク先オブジェクトが存在しない場合（データベースの作成途中でまだそのオブジェクトが作成されていない場合等）、リンク先のオブジェクトをリンク設定時に作成するためにあとからつけ加えられたものである）。

この分野には多くの方法・問題点が存在するが、ここでは TextLink-II システムで採用・実現した機能について述べる。

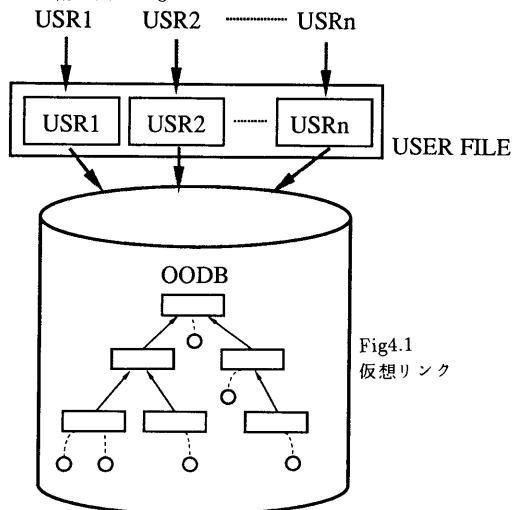
以上のシステム構成をもち、中心機能はリンク管理システムおよびデータベース更新システムである。

4 リンク定義言語 (Link Definition Language: LDL)

4.1 リンク定義言語

これまで述べてきたシステムは、あくまで1つのデータベースを一人のユーザが使用するという前提のもとで開発されてきたものである。したがってオブジェクト間に設定されているリンク情報はすべてデータベースオブジェクトが保持するものであった。しかし、TextLinkシステムにおいては一つのデータベースを複数のユーザが使用するような状況、あるいは一人のユーザが異なる視点からリンクを設定しようとする状況が生じると考えられる。このような場合、リンクの設定状況が複数通り存在することになる。したがってこれまでのようにデータベース内にリンク情報を保持していたのでは、これらの設定の違いを表わすことが不可能である。

そこで本研究ではデータベース内にリンク情報をもたせるのではなく、データベースに対する仮想リンクをこの視点ごとに設定する方法を採用することでこの要求に答えることにした。すなわち、各々のリンク情報がある特別な言語 (Link Definition Language; LDL) で記述、データベースとは独立にこれを保持し、データベースを利用する際にはこの記述を通してデータベースを見るようにすることにより、ユーザから見ればあたかもデータベースに直接リンクが設定してあるかのように見せかける (実際にはデータベース利用時においてもリンク情報はデータベース内には設定されない) 方法を採用した。この方法を用いれば、一つのデータベースを、各ユーザごと、あるいはそれぞれの視点ごとに異なったリンク情報をもつデータベースとして見る事が可能になる。この概念図を Fig4.1 に示す。



ここで用いられるリンクは仮想的であり、その設定状況を記述するのに LDL を用いる。一般的にリンクはあるオブジェクトから別のオブジェクトに対して設定するものと見ることができるが、このときのリンク元のオブジェクト及びリンク先のオブジェクトをそれぞれソースオブジェクト、ターゲットオブジェクトと呼ぶことにする。このように仮想的にリンクを設定する場合、ソースオブジェクトとターゲットオブジェクトとをそれぞれ記述しなければならない。したがって LDL にはソースオブジェクトの記述部とターゲットオブジェクトの記述部を用意する必要がある。これらをソース記述、ターゲット記述と呼ぶ。

4.2 ソース・ターゲット記述

LDL は、ソース記述部およびターゲット記述部から構成される。両者は表現的にはほぼ同一である。ここではそれぞれの記述例について述べる。

オブジェクト指向データベースの文献のうち、'Tanaka'さんが書いた文献で、1986 年以前に書かれたものの本文の 'hypertext' という語句から、そのタイトルに 'hypertext' という語句が含まれている文献に対してリンクを設定する場合を例にとり、ソース及びターゲット記述を説明する。

4.2.1 ソース記述

ここではオブジェクト指向データベースの文献はクラス OODB とそのサブクラスのオブジェクトであるとすると、記述は以下ようになる。

```
Select: hypertext
From: OODB*
Where: (authors name = 'Tanaka'
        and: publishedYear < 1986)
```

ソース部の記述は大きく 3 つの部分に分けられる。

クラスフィールド (From:)

属性フィールド (Where:)

キーワードフィールド (Select:)

(a) クラスフィールド

ここには、OODB のクラス階層によってオブジェクトをチェックするための、クラス及びクラス階層の演算を記述する。上の例における OODB* はクラス OODB とその全サブクラスを表わす。

(b) 属性フィールド

ここには、属性値の満たすべき条件を記述する。上の例では、属性 authors の人物オブジェクトの属性 name が 'Tanaka' でありかつ属性 publishedYear の値が 1986 より小さなものが選ばれる。

(a),(b) によって選択されるオブジェクトは、その条件

部がともに成立（真である）するオブジェクトである。

(c) キーワードフィールド

ここには実際にリンクが張られるキーワードのテキスト式（後述）を記述する。上の例では全テキストの `hypertext` というキーワードからリンクが張られていることを表わしている。

実際にはこのソース記述の (a),(b) によって選択されたオブジェクトのテキスト内にある、このフィールドで記述された語句すべてから仮想リンクが設定されるものと見ることができる。

4.2.2 ターゲット記述

ターゲット記述は以下のようになる。

```
Select: <abstract chapt1>
From: Database*
Where: (title contents: 'hypertext')
```

ターゲット記述は3つの部分からなる。このうち From 及び Where フィールドは上記 (a),(b) と同じである。1 番目のフィールドはリンク先オブジェクトの文献のある部分の名前を記述する。From の Database は、データベースにおける最上位のクラスのクラス名であり、属性フィールド Where における条件式はタイトルに 'hypertext' という語句が含まれるという条件を表わすものである。Select フィールドに記述されている abstract および chapt1 は、クラスフィールドおよび属性フィールドの条件を満たすオブジェクトの abstract および第1章 (chapt1) を表わしている。この部分がリンク先のテキストとなる。

4.3 条件記述

次に実際にフィールドに条件式を記述する場合における記述様式を説明する。

(a) クラスフィールド

ここには、クラスを用いた演算を記述する。このフィールドに記述された演算の結果作られるクラス集合に属する全オブジェクトが次の属性フィールドにおける演算の対象となる。

C クラス C のみを要素とする集合を表わす

C* クラス C とその全サブクラスを含む集合を表わす

クラスを用いたオブジェクトの表示は上記2種類を用いて定義する。これらはすべてクラスを要素とする集合である。次に、このフィールドで許される演算を示す。

& アンドを表わす。すなわち、2つの集合 U および V を考えたとき、U と V の共通の要素をもつ新しい集合を返す。

UNION: オアを表わす。これは U と V の両方の要素をもつ集合を返す。ただし、要素の重複は許さない。

- 差を表わす。U-V は、U から V の要素を除いた集合を返す。V の要素が U に含まれない場合、この要素は無視される。

$U = \{A, B, C\}, V = \{A, D, E\}$ とすると、各演算の結果は、以下ようになる。

```
U & V = {A};
U UNION: V = {A, B, C, D, E};
U - V = {B, C}
```

また、これらの演算は続けて記述することも可能である。この場合 () をもちいて優先順位を付けることができる。() を用いなければ左から順次計算される。記述例を示す。

● クラス C2 と C3 の全サブクラス

C2 UNION: C3*

● C3 とその全サブクラスを除くクラス集合

Database* - C3*

(Database はユーザの作成したデータベースの最上位クラスの名前)

(b) 属性フィールド

このフィールドには、属性の条件を記述する。このフィールドにおける演算対象となるオブジェクトはクラスフィールドによって選択されたクラスの全オブジェクトであり、このフィールドにおける演算の結果選択されるオブジェクトは、このフィールドに記述された条件式の結果が真となるオブジェクトである。

また、ここに記述する属性はクラスフィールドで選択されたオブジェクトに含まれる属性のみではなく入れ子になっているオブジェクトの属性も含まれる。

ここで記述される条件は比較演算子 (=, ≠, >, <, ≥, ≤) を用いて表わされるものと及びその属性の '存在' である。比較演算子の左側には属性名を、右側には文字列・数値あるいは他の属性名またはオブジェクトを記述する。これらの条件はアンド (and:) あるいはオア (or:) を用いた接続が可能である。演算の優先順位等の指定はクラスフィールドの場合と同様である。

1 で述べた例における条件のひとつに、'Tanaka' さんが書いた文献というものがある。これを例にとり説明する。クラスフィールドによって選ばれた文献オブジェクトの属性 authors の値は人物オブジェクトであり、属性 name にこの人物オブジェクトの名前が書かれてあるとする。

この条件は属性 authors のオブジェクトの一つの属性 name が 'Tanaka' であるということを示している。

この場合、明かに上の条件はクラスフィールドによって選ばれたオブジェクト以外のオブジェクトが対象となっ

ている。このときの条件式は次のようになる。

```
authors name = 'Tanaka'
```

このように属性名を続けて記述することによって入れ子になっているオブジェクトの属性値に対する照合も行なうことができる。また、属性値が集合オブジェクトである場合には、その要素の n 番目の値が条件を満たす (at: n)、 n 個の要素の値が条件を満たす (of: n)、すべての要素が条件を満たす (all) を比較演算子の前に挿入することにより集合の要素を満たす条件を指定できる。上の例のように省略した場合には、要素の一つが条件を満たせば、その条件式が成立することを示す。

次に、属性の存在を条件とする場合を考える。属性 year を持つオブジェクトの検索は以下のように記述される。

```
year
```

また、属性値が、オブジェクトの場合にはメッセージを送ることによって（このメッセージは、Smalltalk-80 の文法に準じる）属性値の性質に対しても条件を記述することが可能である。上記の例において著者が 4 人以上である条件は、以下のように記述できる。

```
authors size >= 4
```

このようにメッセージを値に対して送ることも可能である（size は集合の要素数を返す Smalltalk-80 のメッセージである）。

さらに、リンク使用時に属性の値をユーザが指定することにより対話的にオブジェクトを決定することも可能である。このような場合には次のように記述する。

```
keyword = %x
```

これは属性 keyword の値をリンク使用時にユーザが入力することで初めて一つの条件式となる。

また、条件の対象となるオブジェクトに属性が存在しない場合（入れ子になっているオブジェクトについても）、比較条件が適切でない（数値と文字列を比較しようとした、オブジェクトどうしの比較において $=$ 、 $\neq$ 以外の比較演算子を使用した場合等）は、その演算に対する結果はすべて偽と判断される。

(c) キーワードフィールド

ここには、リンクを設定したい部分（単語、パラグラフ等あるいはテキスト式）を記述する。実際のリンクは、1,2 によって選択されたオブジェクトの全テキストのうち、このフィールドに記述されたキーワードから設定されていると見ることができる。テキストにキーワードが含まれない場合にはそのオブジェクトからリンクは設定されない。

キーワード hypertext からリンクを設定する場合には、次のように記述する。

```
Select: hypertext
```

また、テキスト式とは文字列の正規形を含む型のキーワードである。文献内のより細かな範囲を指定するには次のように記述する。

```
Select: hypertext < chap1, chap3 >
```

上記の例では、文献の第 1 章（chap1）及び第 3 章（chap3）の hypertext という語句からリンクが設定されることを意味する。ターゲット側のこのフィールドには、文献の部分名を記述する（chap1,abstract 等）。省略した場合には文献全体を指す。

4.4 リンクの削除・修正

リンクの削除及び変更はテキスト参照中あるいはその他の状態において可能である。削除は通常ソース側において行なわれる。ユーザが削除したいリンクを指定することによってそのリンクの LDL 記述の登録を取り消すことにより行なわれる。

リンクの修正（設定の変更）は設定の場合とほぼ同じである。リンクを指定することにより、そのリンクを記述している LDL を修正することで行なわれる。

5 質問処理法

ここではソース側 LDL 記述及びターゲット側 LDL 記述の、リンク検索時のそれぞれの実行原理を示す。リンクを開く場合、通常はユーザが現在見ている（画面内でアクティブになっている）オブジェクトから開かれる。したがってソース側の記述はこのオブジェクトがソース記述に適合するかどうかを検査するものである。一方ターゲット側はデータベースに対する検索質問となる（実行可能リンクと同じ）。

リンクを開く時にはユーザはリンクの設定してある語句をマウスでクリックし、マウスメニューの 'open link' を選択する。以下の処理は、次のように行なわれる。

- ユーザの選択したキーワードをテキストフィールドに持つ LDL が存在するかどうかを調べる。
- 1 で集められた LDL のそれぞれのソース側について、現在のオブジェクトがその設定に適合するかどうかを調べる。
- 2 で適合した LDL について、それぞれのターゲット記述が記述しているオブジェクトをデータベース内から検索してくる。
- 3 で集められた全オブジェクトがリンク先のオブジェクトとなる。

6 まとめ

今回の研究において我々はオブジェクト指向データベースにおけるユーザインタフェースのプロトタイプと

して Hypertext の概念を取り入れた TextLink-II システムの設計・開発およびそれに対する種々の問題の解決法の一つとして LDL を用いたシステムの設計を行なった。

このインタフェースはユーザとの対話形式を用いた統一的な操作で各機能を実行できるものである。また、文献を対象としたデータベースであるため、この内容の理解を助けるために Hypertext の概念を拡張し、さらにデータベースの作成・更新等が行えるものとした。

また、これに加え LDL を用いて仮想的にリンクを設定することで各ユーザごとに異なったリンク設定を行えるとともに、リンクの更新、データベースの更新によるリンクの更新を容易なものとするのが可能である。

これからの課題としては、LDL の定義をマウスメニュー等を用いた簡単な操作で視覚的に行えるための補助ツールの開発、LDL の実行アルゴリズムの具体化および LDL を用いたシステムの開発等が挙げられる。

参考文献

- [1] Tanaka,K., *Object-Oriented Database System: Background and Concepts*, in Japanese, bit, Vol.20, No.6, pp.687-694, 1988.
- [2] Tanaka,K. and Yoshikawa,M., *Object-Oriented Database System :Trends on Research and Development*, in Japanese, bit, Vol.20, No.7, pp.777-787, 1988.
- [3] Bancilhon,F., *Object-Oriented Database Systems*, Proc. 7th ACM PODS, pp.152-162, March 1988.
- [4] Conklin,J., *Hypertext: An Introduction and Survey*, IEEE Computer Magazine, pp.17-41, Sept. 1987.
- [5] Meyrowitz,N., *Intermedia: The Architecture and Construction of an Object-oriented Hypermedia System and Applications Framework*, Proc. OOP-SLA'86, pp.186-201, Sept. 1986.
- [6] Smith,K.E. and Zdonik,S.B., *Intermedia: A Case Study of the Differences Between Relational and Object-Oriented Database Systems*, Proc. OOP-SLA'87, pp.452-465, Oct. 1987.
- [7] Tanaka,K. and Yoshikawa,M., *Towards an Integrated Environment for Editing, Viewing and Publishing Multimedia Database Objects*, Proc. International Symposium on Computer World'88, Kobe, Japan, pp.59-66, Oct. 1988.