

オブジェクト指向データベースにおける不確実性管理

小林 拳 銭 晴† 田中 克己

神戸大学工学部 計測工学科
† 神戸大学大学院自然科学研究科

本稿では、オブジェクト指向データベース (OODB) システムにおいてファジイデータベース処理能力を実現する方法を提案し、その実現のための基本構成要素を示す。この方法により、(1) オブジェクト自身の曖昧さ、(2) インスタンスオブジェクト間の関係の曖昧さ、(3) instance-of 関係の曖昧さ、および、(4) is-a 関係の曖昧さを取り扱うことが可能になる。この方法の注目すべき特徴は、OODB システムの提供する拡張性を利用した点である。つまり、データベーススキーマを変更せずに曖昧さを付加することが可能である。また、実現例として作成した映画情報データベースを用いて、本システムの機能について簡単に述べる。

Uncertainty Management in Object-Oriented Database Systems

Susumu Kobayashi, Qian Qing† and Katsumi Tanaka

Faculty of Engineering, Kobe University
† Graduate School of Science and Technology, Kobe University

In this paper, we propose an approach and basic constructs to realize a fuzzy database processing capability on an Object-Oriented Database (OODB) system. The proposed approach can manage several types of uncertainties: (1) uncertain objects, (2) uncertain relationships between instance objects, (3) uncertain *instance-of* relationships, and (4) uncertain *is-a* (*subclass/superclass*) relationships. A notable feature is the usage of extensibility offered by OODB systems. That is, it is possible to add these several uncertainties without restructuring a database schema. We will describe our prototype fuzzy OODB system and briefly show its functionalities by using an example movie database.

1 まえがき

近年、データベースにおいて不確実な情報の取り扱いの重要性が目立っており [1], 不確実なデータを格納および検索する方法が研究されている。実際、VLKB, CAD システム, OIS アプリケーション, グループウェアのためのデータベースなどは、人間の知識を含んでおり、それらの知識には明確に定義することができない、また定義する必要のない曖昧なデータが数多く存在する。

従来のデータベースの枠組みでこのような曖昧なデータを取り扱うには、複雑な応用プログラムやスキーマの変更が必要になる。また、データベースに対して不確実な問い合わせを行えることも望ましいが、従来の関係データベース (RDB) における問い合わせ言語 (SQL など) では、そのような機能は提供されていない。これらの理由から、不確実な情報の処理および検索の機能を有するデータベースシステムが報告されているが [2] [3] [4] [5] [6], それらは理論的基盤の確立している RDB を拡張したものである。しかし、ファジイ RDB システムには以下の欠点が挙げられる。

意味論的な不明確さ

ファジイ RDB システムにおいて、ファジイリレーションは有限個のドメインの直積によるファジイ部分集合として定義される。各ドメインはファジイ集合であってもよい。ファジイリレーションの個々のタプルは、そのタプルに対するメンバシップ値をもち、その値は属性 μ の値として表わされる。

例えば、ファジイリレーション Employee(Name, Job, Salary, μ) において次のタプル t を考える。

$$t = (\text{John}, \text{engineer}, 60000, 0.67)$$

ファジイリレーションの数学的な定義によれば、値 0.67 はタプル t がファジイリレーション Employee に属する可能性の度合を意味している。ところがこの解釈は誤解されやすく、ユーザは以下のような解釈をするかもしれない。

1. John が Employee である度合が 0.67 である。
2. John の職業が engineer である度合が 0.67 である。
3. John の給料が 60000 である度合が 0.67 である。

複合オブジェクト管理能力の弱さ

先の例において、John が Employee である度合が 0.9 で、John の職業が engineer である度合が 0.8、そして John の給料が 60000 である度合が 0.6 であるとする。ファジイドメインを用いると、この情報は次のように表わされる。

$$((\text{John}, 0.9), \text{engineer}, 0.8) \text{ and} \\ ((\text{John}, 0.9), 60000, 0.6)$$

従来の RDB システムにおいては、このような複合オブジェクトの取り扱いが十分でなく、NF²への拡張が必要になる。

拡張可能性の低さ

スキーマを作成した後、あるドメインをファジイドメインに拡張する場合、根本的なスキーマの変更が必要になる。例えば属性 Salary のドメインを、high, low のようなファジイ述語や、ファジイ集合を含むファジイドメインに変更することを考える。このとき、まずメンバシップ値を示す属性を追加することが必要になるし、また、元のスキーマも再構成しなければならないかもしれない。

スキーマ定義の曖昧さの表現能力の弱さ

従来のファジイ RDB では、データや問い合わせを不明確にすることが可能になったが、スキーマ自体を不明確にすることができなかった。これは従来の OODB についても同様である。しかし、アプリケーションによっては、データベーススキーマ (RDB では表の骨格, OODB ではクラス階層) を明確に定義することが困難な場合がある。

OODBMSは、RDBと比べて、モデリング、複合オブジェクト管理、拡張可能性において優れた能力を提供している[9]。そこで本稿では、上記のファジイRDBの欠点を補うためにOODBモデルに基づいてファジイDB機能を実現する方法を提案する。また、その実現のためのいくつかの基本構成要素を示す。

2 OODBにおける不確実性

前節で述べたように、不確実情報を管理する場合、意味論的表現が不明確になったり、スキーマ定義の曖昧さのように従来の枠組みでは表現することもできないことがある。そこで本節では、データベースシステム(特にOODBシステム)において生じ得る不確実性にはどのようなものがあるかを考察し、その管理方法について述べる。

一般にデータベースで取り扱われる情報は、その情報の内容であるデータオブジェクトと、データオブジェクト間の関連によって記述される。よって、情報の曖昧さには、(1)データオブジェクトに起因する曖昧さと、(2)関連に起因する曖昧さがあると考えられる。さらに、関連には、(i)インスタンスオブジェクト間の関連を示す属性-属性値関連、(ii)インスタンスオブジェクトとクラスオブジェクト間の関連を示すinstance-of関連、(iii)クラスオブジェクト間の関連を示すis-a関連、そして、(iv)Hypertextなどにみられる参照関係を示すリンクなどがある。これらのすべてについて、互いに独立に曖昧さが存在するであろう。

以下では、上に示した分類に従って、それぞれの曖昧さを管理する方法について述べる。

2.1 オブジェクト自身の曖昧さ

情報の不確実性の最も一般的な原因として、オブジェクト自身を明確に表現することが困難なことが挙げられる。例えば、リレーションEmployeeにおいて次のような情報を表現することを考える。

1. Michael's Job is *researcher or engineer*.
2. Linda's Salary is *high*.

ここで、属性Jobに対する値は、ドメインJobScriptに属する1つのオブジェクトとし、属性Salaryに対する値は、整数型の1つのオブジェクトとする。

前者の例では、オブジェクトMichaelの属性Jobに対する値がドメインJobScriptに属する1つのオブジェクトでなければならないが、明確に決定することができず、その候補を示している。このような情報はOR型情報(disjunctive information)と呼ばれる。

後者の例では、オブジェクトLindaの属性Salaryに対する値が、整数ではなくファジイ述語highになっている。

これらのオブジェクトの曖昧さは、ファジイ集合やメンバシップ関数を用いて表現することができる[6][7]。前者の例では、次のようなファジイオブジェクトres_or_engを定義する。

$$\text{res_or_eng} = \{ 0.9/\text{researcher}, 0.8/\text{engineer} \} \text{ (ファジイ集合)}$$

ここで、0.9および0.8はファジイ集合のグレード値を表わし、researcherおよびengineerはドメインJobScriptに属するオブジェクトである。また、ファジイオブジェクトres_or_engは、通常の集合値(set-value)ではないことに注意しなければならない。

このようなOR型情報の表現と解釈については、文献[8]のOR-Object、文献[5]のp-domainなどで論じられている。

また、ファジイ述語(ファジイオブジェクト)highは、整数値とグレードを対応づけるメンバシップ関数 μ_{high} で定義することができる。

$$\mu_{high}(80000) = 1, \mu_{high}(70000) = 0.9, \dots$$

上記のファジイオブジェクトは通常の明確なオブジェクトと同様に取り扱われることが望ましい。

オブジェクトの曖昧さを実現する方法としては、ファジイ述語やファジイ集合の定義を特定の表またはクラスで管理し、その値が割り当てられる属性のドメインをファジイドメインに拡張すること

が考えられる。しかし、この方法ではスキーマの変更が必要であるし、オブジェクトの曖昧な振舞いを実現するためには複雑な手続きが必要になる。

OODB モデルでは、オブジェクトはデータと手続き(メソッド)を一体化したものであり、オブジェクトの振舞いはメソッドの定義に従うことになる。よって、ファジイオブジェクトの曖昧な振舞いを実現したいときは、その振舞いの様子をメソッドによって定義すればよい。また、属性に対するドメイン(クラス)の特殊化したドメイン(サブクラス)を定義し、ファジイオブジェクトをそのクラスのインスタンスにすることにより、属性に対するドメインをファジイドメインに拡張しなくてもその属性にファジイオブジェクトを割り当てることができる。スキーマの変更が不要になる。

例えば、整数クラス Integer のサブクラスとして FuzzyInteger を作成し、ファジイオブジェクト *high* を、FuzzyInteger クラスのインスタンスとする。そして、FuzzyInteger クラスにおいて、例えば、オペレータ '=' の定義を、比較値がある範囲内であれば真を返し、さらに比較値をメンバシップ関数の引数にしたときの戻り値を、曖昧さとして表示する、というように再定義する。同様にして、Integer に定義されている他のメソッドを再定義すれば、FuzzyInteger クラスのインスタンスは通常の整数値と同様に取り扱うことができ、複雑な手続きや応用プログラムを書かなくても曖昧な振舞いを実現できる。

2.2 関連の曖昧さ

オブジェクト自身に曖昧さがなくても、オブジェクト間の関連に曖昧が生じることがある。関連は、先に述べたように (i) ~ (iv) に分類されるが、ここでは (i) ~ (iii) の関連に関する曖昧さについて述べ、(iv) については、5 節で詳しく述べることにする。

(i) 属性-属性値関連の曖昧さ

OODB では、インスタンス間の関係は属性として表現される。オブジェクト自身に曖昧さがなくても、それらの関係が曖昧になるかも知れない。例えば、

John might be perhaps a friend of Susan.

という情報において、*John* と *Susan* は通常の明確なオブジェクトで、オブジェクト *John* が *Susan* の属性 *friend* に割り当てられるときに曖昧が生じると考えられる。

(ii) instance-of 関連の曖昧さ

OODB では、すべてのインスタンスはある 1 つのクラスに属さなければならない。しかし、曖昧な情報を取り扱う場合、この制限は厳しすぎるが多く、複数のクラスに所属させて、クラスに属する可能性の度合を表現したいことがある。

(iii) is-a 関連の曖昧さ

OODB では、クラス階層は上位クラスで定義されたデータ構造やメソッドを共有または継承するために構成される。また、それと同時に、実世界の実体の概念的な集まりの分類階層に対応している。is-a 階層の曖昧さは、クラス階層を分類階層とみなした場合に生じる。

クラス階層を継承機能のために定義するためのものであると考える場合、is-a 関連の曖昧さは、あまり重要でないかもしれない。しかし、ここでは、クラス階層をオブジェクトの分類階層であるとみなしたときに、その階層において曖昧さが表現されるべきであるということに注目する。

3 ファジイ OODB の構成要素

2 節で述べた曖昧さはそれぞれ独立なので、それらを管理するには明確な区別をしなければならないが、RDB では、レコードベースのデータ構造のため、これらの曖昧さを同時に、明確な区別をしながら管理することが困難である。

OODB では、クラスの定義により各機能をモジュール化することが可能であり、複合オブジェクトによって複雑なデータ構造を管理することが容易である。

そこで本節では、ファジイ OODB 機能を実現するために基本構成要素となるクラスを作成することを提案する。各クラスは独立であるから、それぞれの曖昧さを明確に区別しながら管理することが可能であり、これらのクラスを追加するだけで容易にファジイ機能を付加することができる。以下に、本システムで実際に用いたクラスの概要を示す。

3.1 Fuzzy-Object クラス

このクラスはファジイオブジェクトを実現するためのクラスである。このクラスのインスタンスは、主に2つの属性 `objectName` と `contents` を持つ。(Fig. 1 参照)

`objectName` の値は、そのファジイオブジェクトを識別するための文字列で、`contents` の値は、次の形式をもつタプルのリストである。

(度合, 要素となるオブジェクトへのポインタ)

つまり、`contents` の値は1つのファジイ集合を表している。

この構造により、ファジイオブジェクトを1つのオブジェクトとして取り扱うことが可能になる。さらに、このクラスを属性値のクラスのサブクラスとして定義することで、その属性値のクラスを変更せずに、ファジイオブジェクトを割り当てることが可能になる。

3.2 Fuzzy-Instance-of クラス

このクラスは `instance-of` 関連の曖昧さを実現するためのクラスである。基本的なデータ構造は Fuzzy-Object クラスと似ているが (Fig. 2 参照)、属性 `contents` の中のタプルにクラス名が含まれており、そのクラスとの `instance-of` 関連を示すことができる。

例えば、Action クラスと Dramatic クラスの両方に曖昧さを持って属する映画オブジェクトを生成する場合、本システムでは、まず Action クラスと Dramatic クラスのそれぞれにおいてインスタンスが生成される。次に、このクラスのインスタンスが生成され、先に作成されたインスタンス

を要素とし、`instance-of` 関連の曖昧さを示す度合と共に管理する。

3.3 Fuzzy-Class-Hierarchy クラス

このクラスは、クラス階層の曖昧さを管理する。このクラスのインスタンスが、ユーザの定義した各々のクラスに対応したノードとなり、クラス階層を表す木構造を形成する。

このクラスのインスタンスは、その対応するクラスのクラス名を示す文字列と、親クラスと子クラスとの間の `is-a` 関連の曖昧さを表す2つのリストで構成されている。(Fig. 3 参照)

リストはいくつかのタプルで構成され各タプルは、`is-a` 関連のあるクラスに対応するこのクラスのインスタンスへのポインタと、その曖昧さの度合を示す数値で構成される。

この方法により、`is-a` 関連の曖昧さは、ユーザの定義したクラスの内部構造と独立に管理することができる。

4 実現例

本節では、上記の概念に基づいて作成されたサンプルデータベースを紹介し、その基本的な機能について述べる。

今回作成したデータベースは、Postscript の画像データとテキストデータで構成される映画情報に関するデータベースで、SPARC ステーション上で OODBMS Versant[12] を用いて実現され、現在稼働中である。

映画情報データベースを作成するにあたり、まず、アクションや SF といったカテゴリの分類に従ってクラスを作成し、さらに分類の特殊化に従ってクラス階層を作成した。その際、分類の特殊化の関係が不明確な部分に曖昧さを持たせた。また、いくつかの映画オブジェクトを、複数のクラスに属するインスタンスとして、曖昧さを持たせて定義した。さらに、映画の作成された年、映画のストーリーの中で設定されている年代が不明確であることから、いくつかのファジイオブジェクトを定義し、それらを映画オブジェクトの属性

値として割り当てたり、検索条件に使用できるようにした。

例えば、Fig. 4 の例では、カテゴリ ActMovie に属し、作成された年が recent である映画を検索している。その結果として、ファジイオブジェクト *about1989* を属性値として持つ映画が得られ、min-max 法によって評価された値 0.9 が条件を満たす可能性として表示されている。また、結果 Image:2 のウィンドウ中の is-a ActMovie:0.75 および instance-of HardAct: 0.92 は、HardAct が ActMovie のサブクラスで、その is-a 関連の度合が 0.75 であり、この映画が HardAct クラスに属する度合が 0.92 であることを示している。

5 議論:ハイパーテキストと不確実性管理

本稿では、OODB におけるいくつかの不確実性について述べ、その管理方法を提案した。しかし、リンク関連の曖昧さについては、データモデルと実現方法が特殊であり、また、ハイパーテキストにおいて重要な概念であるとおもわれるリンクの抽象化と関連しているので、ここで詳しく述べることにする。

5.1 リンクの本質的化

リンクは情報の内容とは別に情報の構造を記述する機能を提供する。本稿では、リンクを次のように定義する。

リンクとは、あるポジションにおける情報オブジェクトの構成要素の集約とそれに対する情報オブジェクトの集約との間の関連である。

ここでポジションとはデータベースにおけるビューやスコープのようなもので、ハイパーテキストではユーザが現在みている文書に相当する。情報オブジェクトとは、オブジェクト指向モデルにおけるインスタンスオブジェクトに相当し、ハイパーテキストでは個々の文書が情報オブジェクトとなり、単語および単語の配列がその構成要素

になる。集約 (aggregation) とはある制約をみたすオブジェクトの集まりである。

上記の定義によれば、リンクは、出発点となるオブジェクト (ソース) の集約と、リンクを辿ることによって得られるオブジェクト (ターゲット) の集約の、2つの集約を決定する制約条件のペアを指定することで定義することができる。(例えば、文献 [10] の質問対リンクがこれに相当する。)

ここで、リンクへの曖昧さの導入として次の2つが挙げられる。

1. リンクの定義自体への曖昧さの導入
2. 制約条件の評価への曖昧さの導入

制約条件を満たすことに曖昧さがある場合、集約に属するオブジェクトの集合をファジイ集合に拡張することができる。このとき、各要素が集約に属する度合 (メンバシップ値) とリンクの定義自体の曖昧さを総合的に評価することにより、リンクの曖昧さを表わすことができる。

5.2 リンクの抽象化

個々のオブジェクトに対して固定的に定義されているリンクは、オブジェクトの修正/挿入/削除などの操作や、スキーマの変更に対する対応が困難である。質問対リンク [10] では、適用されるリンクと、リンクの対象となるオブジェクトを、質問対の実行により動的に決定することにより、より柔軟なリンクを実現した。ところが、依然次のような問題点がある。

- 質問の条件が厳密すぎる。
- 抽象的なリンクを実現しにくい。

以下では、不確実性管理の応用という観点から、これらの問題点を解決する方法を考察する。

リンクの柔構造化

例えば、次のような質問対リンクを考える。

```
ソース質問
select 'OODB'
from Paper
where
```

```
ターゲット質問
select *
from Book
where title contains 'object-oriented'
and publishedYear ≥ 1980
```

この定義では、文字列'OODB'が正確に選択されなければリンクの対象にならないが、例えば、'object base'など、関連のある文字列が選択されても、このリンクが適用されることが望ましい。また、ターゲット質問の「出版年が1980年以降」のような明確な条件でなく、「最近出版された」のような漠然とした条件の方が、リンクの意図をより忠実に表現できるかもしれない。

このようにリンクに不確実性を持たせることにより、リンクの対象になる可能性のあるオブジェクトの集約をより適切に指定することが可能になったり、曖昧さの度合を用いることでリンクの対象となるオブジェクトに優先順位をつけることが可能になる。

リンクの抽象化

通常、リンクは互いに独立に定義され、それらの間の関係を示す方法がないが、質問対の条件の記述の形式が類似していたり、より一般的な、あるいはより特殊なリンクがあることが考えられる。ここでは、リンク間の関連の1つとして抽象化を考察する。

(A) 制約の汎化に基づくリンクの抽象化

リンクは制約を示すことにより定義されるので、制約に汎化関係があれば、リンクにもその関係が影響することが考えられる。

(i) 変数の使用による制約の汎化

例えば、選択された文字列から、それをキーワードとする文書へのリンクは、次の質問対により定義することが考えられる。

```
ソース質問
select X
from
where
```

```
ターゲット質問
select *
from
where keyword contains X
```

このように、変数を用いることで無数のキーワードに対するリンクを実現することができる。この質問対は、キーワード1つ1つに対する具体的な質問対を抽象化して、それらのリンクの形式を記述していると考えられる。

また、変数Xのように、ソース質問とターゲット質問の両方で共通に使用できれば、より質問の記述能力が高くなるであろう。

(ii) ファジイオブジェクトによる制約の汎化

ファジイオブジェクトは、その候補となるオブジェクトの識別性を失わせる汎化的な機能を持つ。よって、変数を使用する場合と同様に、具体的に候補になっているオブジェクトに対するリンクをより抽象化できるであろう。さらに、そのリンクが適用されるとき、ファジイオブジェクトの持つ曖昧さによって、リンクの曖昧さも表現できるであろう。

(B) リンク構造の要約に基づくリンクの抽象化

ハイパーテキストを使用している際、自分の見ている文書から辿ることのできる情報がどのような構造をしているかを、大まかに知りたいことがある。このような場合、情報構造を要約した目次のようなものがあれば、いま見ている文書がどのように構成されているかを知ることができるであろう。

例えば、リンクのソースとなる文字列をエントリとして目次を作成し、さらに辿ることのできる文書に対して再帰的に目次を作成すれば、情報構造を見通すことができ、何度か辿ることによって得られるオブジェクト、およびそのパスを知ることでもできる。さらに、リンクに曖昧さを表現する

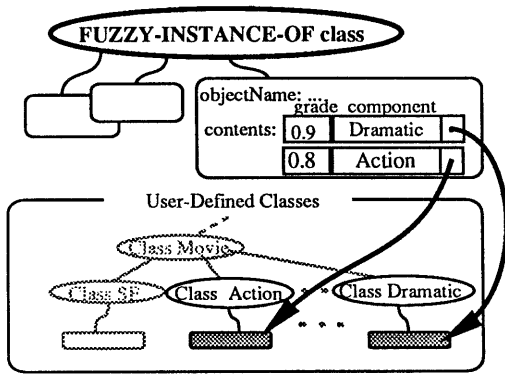


Fig.2 FUZZY-INSTANCE-OF class

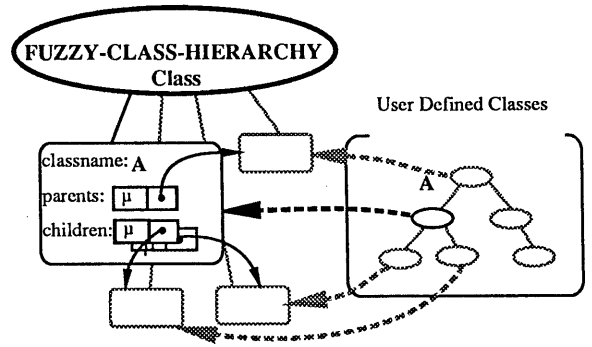


Fig.3 FUZZY-CLASS-HIERARCHY Class

Query Window

choice: fuzzy set selected.

Select _____

From ActMovie

Where

Year

recent _____

segment value: non fuzzy

execute

Image: 1

Poss. Of Satisfying Condition : 0.900000

is-a ActMovie : 0.800000

Title : Indiana Jones

Year : about1909

Director : George Lucas

Leading Actor : Halison Ford

Nation : USA

Image: 2

Poss. Of Satisfying Condition : 0.800000

is-a ActMovie : 0.750000

instance-of HardAct : 0.920000

Title : Die Hard

Year : about1908

Director : John McLean

Leading Actor : Blues Wills

Nation : USA

Fig.4 検索結果