

アイコンックブラウザの開発

堀尾 祥久^{*}、津田 和幸^{*}、平川 正人^{**}、田中 稔^{**}、市川 忠男^{**}
広島大学大学院工学研究科^{*} 広島大学工学部^{**}

オブジェクト指向データベースにおける問合せ言語は、データベース操作に不慣れなユーザにとって解りにくいものになっている。したがって、初心者でさえ簡単に操作できるような親和性の優れたユーザインタフェースが必要とされている。本稿では、ユーザがアイコンを用いてデータベース内のオブジェクトを検索することができるアイコンックブラウザを提案する。アイコンはデータベース内のクラスあるいはオブジェクトを表現しており、問合せはアイコンに別のアイコンを重ねていくことで記述される。アイコンックブラウザは重ね合わされたアイコンを解釈することにより、データベース操作を行い、結果をユーザに表示する。

Development of Iconicbrowser

Y. HORIO, K. TSUDA, M. HIRAKAWA, M. TANAKA, and T. ICHIKAWA
Information Systems, Faculty of Engineering, Hiroshima University
Shitami, Saijo-cho, Higashi-Hiroshima 724, Japan

Query languages for object-oriented databases are complicated for the user unfamiliar with database manipulation. Development of friendly interfaces which can be easily manipulated by the novice user is strongly requested. This paper describes an iconicbrowser which enables the user to retrieve objects by means of icons. Icons represent classes and objects in the database. Queries are specified by overlapping one icon over another. The iconicbrowser interprets overlapped icons into the database operations, and then displays the result to the user.

1. はじめに

データベースの重要性が認識されるにつれて、一般のユーザがデータベースを直接操作する機会が増えてきている。しかしながら、データベースに関する知識をほとんど持たないユーザにとって、問合せを記述することは複雑で、かつ煩わしい仕事である。また、特にオブジェクト指向データベースのように複雑なデータを管理することができるシステムでは問合せの記述が複雑なるため、従来のシステムに比べて魅力的な機能が提供されていたとしても、それらの機能をユーザが十分に使いこなせないといったことが起こりうる。こうした問題を解決するために、親和性の優れたユーザインタフェースが必要とされている。

そのためのアプローチの一つとして、視覚情報をユーザインタフェースに用いる試みがある[1][2]。例えばQBE[3]では、表骨格(table skelton)と呼ばれる、関係構造を表現する枠を通して問合せを記述する。表骨格に関係名および属性名を与えた後に、定義域変数、定数、および演算子を表骨格に埋め込むことが問合せ記述となる。また、画面上に表示されたデータベーススキーマ上でデータベース操作を行うことができるブラウザも数多く提案されている[4][5]。しかしながら、視覚情報を何に対してどのように用いるかによって、その効果に大きな違いが生じてくる。

本稿において、我々はオブジェクト指向データベースに基づくアイコンックブラウザを提案する。

アイコンックブラウザでは、QBEとか従来のブラウザのようにデータベーススキーマに関する情報を視覚化するのではなく、操作対象そのものを絵シンボル(アイコン)として視覚化する。すなわち、アイコンはデータベース内の実体(オブジェクト)あるいは抽象概念(クラス)を表現する。ユーザの問合せは画面上でアイコンを重ね合わせることによって行われる。このとき、あるアイコンと重ね合わせることでできるアイコンはブラウザによって提示されるため、データベーススキーマの知識を持たないユーザでさえ、データベース操作が行える。また、ユーザの記述した問合せはシステム内において、それに対応したSQL風問合せ記述に変換され、その記述はアイコンの制約状態を保持する。

まず2章でオブジェクト指向データベースについて概観する。3章ではアイコンックブラウザの概要と具体的な検索方法について述べる。システム構成とアイコンの管理について4章で述べた後に、5章でまとめを行なう。

2. オブジェクト指向データベース

オブジェクト指向データモデルは、従来のデータモデルと比べて、複雑な構造を持つデータを柔軟に、かつ自然に表現することができる。データベースは現実世界の個々の物を表現するオブジェクトとオブジェクト間の共通な性質を表現するクラスとによって形成される。データベースのスキーマはクラス間のスーパー/サブ関係を示すクラス階層とオブジェクトのデータ構造を示すアグリゲーション階層によって定義される。

クラス階層は意味論モデルにおける汎化/専化の概念

を表現するとともに、性質の継承を実現する。図1にクラス階層の例を示す。ここで、VISUALクラスはElectrical_Appliancesクラスのサブクラスであり、VTRクラスはVISUALクラスとAUDIOクラスのサブクラスである。このとき、例えばVISUALクラスはElectrical_Appliancesクラスの性質を継承し、かつVISUALクラスに属するオブジェクトはElectrical_Appliancesクラスのオブジェクトとしても取り扱われる。

アグリゲーション階層はオブジェクトのインスタンス変数の定義に相当する。図2にElectrical_Appliancesクラス、TV_setクラス、およびVTRクラスのアグリゲーション階層を示す。Electrical_Appliancesクラスでは、インスタンス変数nameの値がStringクラスに属するオブジェクトであり、priceの値がIntegerクラスに属し、かつmakerの値がStringクラスに属するということが定義されている。TV_setクラスおよびVTRクラスでは、それぞれのクラスに定義されている性質だけでなく、図1に示したクラス階層に従って、Electrical_Appliancesクラスの性質も継承する。オブジェクトに対する操作はクラスのメソッドとして記述される。

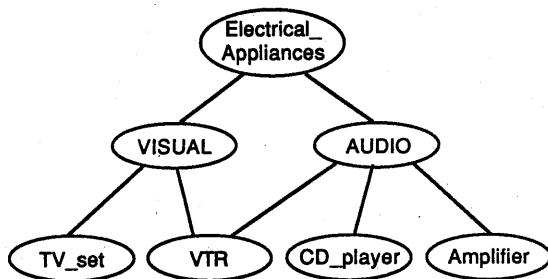


図1 クラス階層

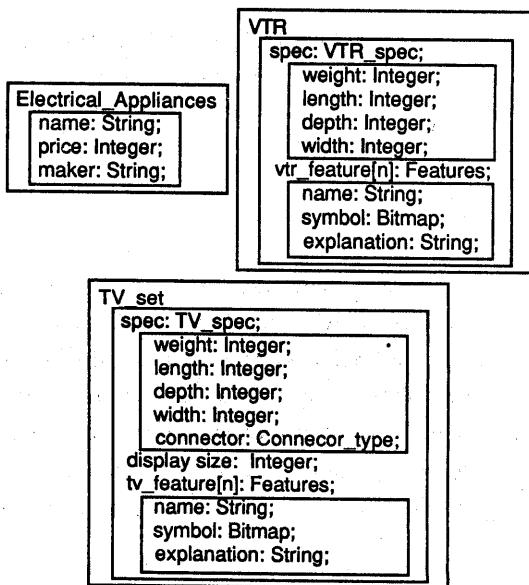


図2 アグリゲーション階層

3. アイコニックブラウザ

3.1 検索モデル

オブジェクト指向データモデルは現実世界を自然に表現できる記述能力を備えており、これまでのデータモデルと比較して、ユーザーの概念モデルとのギャップが小さい。しかしながら、データベースを操作するにあたってはデータモデルの理解が前提とされるため、一般のユーザーにとってはデータベース操作は依然困難であった。しかもオブジェクト指向データモデルの持つ柔軟なデータ表現能力はかえって問合せ操作を解りにくいものになっている。このため、データモデルをユーザーに意識させることなくデータベース操作が行えるようなユーザー親和性に優れたインタフェースが必要である。

本研究では、こうした問題を解決するにあたって、アイコンを用いてデータベースを操作するアイコニックブラウザを開発している。一般に、アイコンは現実世界に存在する事象を抽象的に表現するために用いられる。本稿で提案するアイコニックブラウザでは、オブジェクト指向データモデルにおけるクラスおよびオブジェクトをアイコンを用いて表現する。

オブジェクト指向データベースにおいては、検索は任意のクラスから所望のオブジェクトを選び出すためのアクセスパスを記述することと考えられる。例えば図1に示したスキーマにおいて、あるビデオデッキを検索することを考える。Electrical_Appliances クラスから検索を開始したとすると、それからVISUAL クラス、VTRクラスへと徐々にクラス階層を下がっていく。その後でユーザーはVTR クラスのインスタンス変数の値に関して制約を与えることで最終的に所望のオブジェクトを見つけ出す。すなわち、検索開始クラス、検索途中の制約付クラス、および選びだされたオブジェクトをそれぞれ状態と見なすと、データベース操作は図3に示すように、それらの状態の遷移と見なすことができる。図中において、円は状態を、線は状態間の遷移を表している。オブジェクト集合という状態はオブジェクト状態のアイコンの集りである。それぞれの線に付けられているラベルは状態遷移を引き起こす操作を示しており、これについては3.2で詳述する。

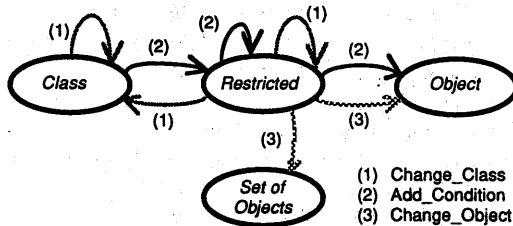


図3 検索における状態遷移



図4 アイコンの状態

提案するアイコニックブラウザにおいては、アイコンは図3に示しているように“クラス”、“制約付クラス”、および“オブジェクト”といった三種類の状態のうちいずれかを取る。すなわち、“クラス”状態にあるアイコンを“オブジェクト”状態に変えることがデータベースに対する問合せ作成に対応する。

アイコンの状態はその形態によって区別できるようになっている。VTRの場合を例にとりて、“クラス”、“制約付クラス”、“オブジェクト”それぞれの状態表現形態を図4に示す。“制約付クラス”状態のアイコンはクラスアイコンが別のアイコン上に重ねられていることで表現する。“オブジェクト”状態のアイコンでは、オブジェクトを識別するためのインスタンス変数の値がアイコン内に表示される。この変数を代表項目と呼ぶ。これ以降、状態の遷移が起こるアイコンを対象アイコンといい、そのアイコンが示すクラスを対象クラスと呼ぶ。さらに対象アイコンの状態を遷移させるアイコンを制約アイコンと呼ぶことにする。

3.2 基本操作

本節では、提案するアイコニックブラウザの具体的な操作方法について述べる。

システムの操作画面を図5に示す。画面は検索エリアとアイコンエリアとで構成されている。ユーザーは検索エリアにおいてアイコンを重ね合わせることでオブジェクトの検索操作を行なう。アイコンエリアには検索操作に用いることができるアイコンが表示される。システム起動時に表示されるアイコンを初期アイコンと呼ぶ。アイコンエリアにはデータベース内の全てのクラスに対応するアイコンが表示されているわけではない。ユーザーは検索操作を行ないやすくするために、必要なアイコンのみを初期アイコンとしてアイコンエリアに登録することができる。初期アイコンは3.1で述べた状態のうちいずれであってもよい。

ユーザーが初期アイコンの一つを検索エリアに取り出したとき、そのアイコンは対象アイコンになる。検索操作はこの対象アイコンに対して行なわれる。検索のための基本操作は大きく二つに分けられる。一つは検索に必要な

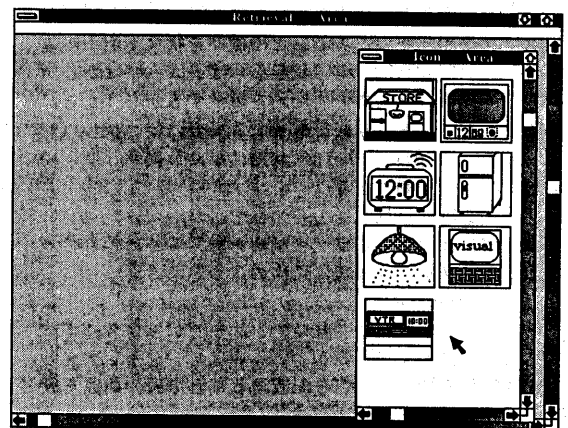


図5 操作画面

なアイコンあるいはアイコンが示しているオブジェクトのリストを表示させるためのもので、アイコン操作と呼ぶ。もう一つは対象アイコンの状態を変えるために用いる遷移操作である。

まず、アイコン操作について述べる。具体的には次の2つのコマンドが用意されている。これらのコマンドの実行は、特定のマウスボタンのクリックによって引き起こされる。

(1) Replace

この操作は、対象アイコンに対応したオブジェクトを画面上に表示させるためのものである。対象アイコンが“クラス”状態であればそのクラスに属するすべてのオブジェクトが、“制約付クラス”状態であればその制約を満たすオブジェクトが、“オブジェクト”状態であればそのオブジェクトがreplace windowと呼ばれるウィンドウに表示される。このとき、該当するオブジェクトは代表項目の値によって表示される。Replace操作をVTRアイコンに適用した例を図6に示す。replace windowにはVTRアイコンの代表項目であるビデオデッキの名前が表示されている。アイコンが“制約付クラス”状態にあるとき、指定された条件を満たしているオブジェクトは実線文字で表示され、そうでないものは点線文字で表示される。replace window上でユーザは表示されている値を直接選んだり、値の範囲を設定することができる。

(2) Icon_Set

この操作は対象アイコンに適用できるアイコンを表示させるためのものである。この操作が行われると画面上にpop-up icon menuが現れ、対象アイコンに適用可能なアイコンが表示される。この操作をVTRアイコンに適用した例を図7に示す。

遷移操作はアイコンの状態を変えるものである。図3に示されているラベル(1)、(2)、(3)はそれぞれ以下のChange_Class、Add_Condition、およびChange_Objectの操作に対応する。

(1) Change_Class

この操作は対象アイコンの示すクラスを変更するためのものである。この操作は、対象アイコンに対象クラスのスーパークラス、サブクラス、あるいはシプリングクラスのいずれかに対応するアイコンが重ねられた時に起動される。対象クラスは重ねられた制約アイコンに対応したクラスに変わる。ここで、この操作を行なう前の対象アイコンが“制約付クラス”状態ならば、ユーザは新しい対象アイコンがそれまでの制約を継承するかどうかを決めることができる。制約を継承する場合には、新しい対象アイコンは“制約付クラス”状態になり、継承しない場合は“クラス”状態になる。

(2) Add_Condition

この操作は対象アイコンに制約を加えるためのものである。制約を加える方法には次の3つがある。

1) selection

replace windowにおいて、ユーザがマウスによって値を直接指定するか、あるいはレンジセレクタと呼ばれる“▲”、“▼”を配置することで値の範囲を設定する。

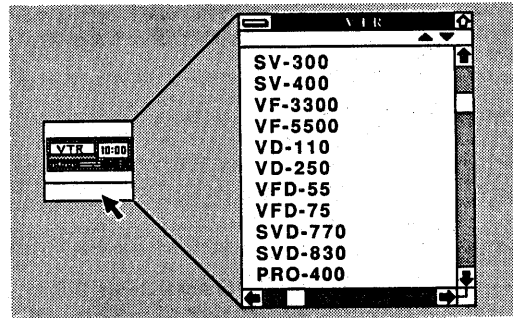


図6 VTRアイコンに対するReplace操作

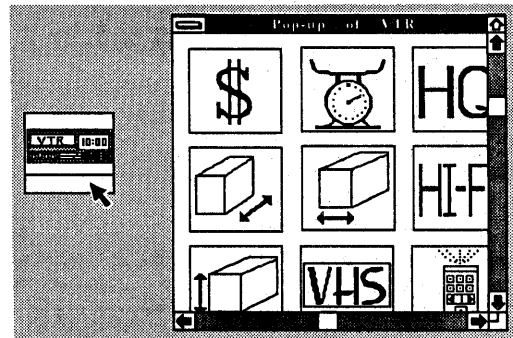


図7 VTRアイコンに対するIcon_Set操作

2) restriction

対象アイコンに、状態が“制約付クラス”か“オブジェクト”である別のアイコンを重ねる。重ねたアイコンが対象アイコンのインスタンス変数の一つに対応している時にrestriction操作と解釈される。対象アイコンは制約アイコンによって記述された値を持つオブジェクトに対応するアイコンになる。

3) join

対象アイコンが結合アイコンと呼ばれる特別なアイコンによって他のアイコンと結合された時、結合された2つのクラスのオブジェクトの値が、指定された比較演算子によって比較される。この操作の結果、対象アイコンは比較条件を満たすオブジェクトに対応したアイコンになる。比較するための値と比較演算子は結合アイコン中に記述する。

(3) Change_Object

この操作は、対象アイコンに順次条件を与えていき、条件を満たすオブジェクトの数が一つになったときに自動的に起動される。対象アイコンの状態は“制約付クラス”から“オブジェクト”に変わる。ユーザが直接この操作を起動することもでき、“制約付クラス”状態がアイコンをその制約を満たすオブジェクトの集り(“オブジェクト集合”状態)に展開される。

3.3 検索例

ここでは具体例を挙げて、オブジェクト検索がどのように行われるかについて述べる。例として、図1と図2で記述したスキーマ上で以下のような問合せを考える。

問合せ1

全てのビデオデッキを選び出す。

問合せ2

600ドル以下のビデオデッキを選び出す。

問合せ3

問合せ2の条件を満たしているビデオデッキの中で、ハイファイ機能のついたものを選び出す。

問合せ4

問合せ3の条件を満たしているビデオデッキの重量を表示する。その中から最も軽いものを選ぶ。

問合せ5

問合せ4で選ばれたビデオデッキに接続することのできるテレビを選ぶ。

検索操作は図5に示されている状態から始めると仮定する。アイコンエリアにVTRアイコンが存在するので、それを検索エリア内に対象アイコンとして選び出す。アイコンエリア内に目的のアイコンが存在しないときは、目的のアイコンを含むクラスを表現するアイコンから検索を開始し、VTRアイコンが見つかるまでIcon_Set操作ならびにChange_Class操作を繰り返す。VTRアイコンの検索途中で得たアイコンをアイコンエリアに置くことで初期アイコンとして登録することができる。アイコンエリアにアイコンを登録すると、次の検索操作の際には、そのアイコンから検索を開始することができる。

ビデオデッキの名前は図6に示されているように、VTRアイコンにReplace操作を適用することで得られる。ここではVTRアイコンの代表項目としてビデオデッキの名前が指定されており、ビデオデッキの名前が表示される。これらの操作は問合せ1に対応する。

問合せ2は次のように記述される。まずVTRアイコンのpop-up icon menuからPriceアイコンを選び、VTRアイコンにPriceアイコンを重ねる。次に、PriceアイコンにReplace操作を適用し、値の範囲を設定することで条件付けを行う。値の範囲指定には、図8に示されているように、上向きあるいは下向きの三角形を用いる。この操作の結果、600ドル以下の値段を持つビデオデッキの名前がVTRアイコンのreplace windowに表示される。

問合せ3を記述するためには、先の問合せで得られたVTRアイコンにpop-up icon menu内のHi-Fiアイコンを重ねる。図9に示しているように、VTRアイコンにReplace操作を適用すると現在の制約を満たすビデオデッキの名前を見ることが出来る。条件を満たすオブジェクトは実線文字で示されている。

選ばれたビデオデッキの重量は該当するVTRアイコンに重量に対応するアイコンを重ねることによって知ることが出来る(図10)。問合せ3を満たすビデオデッキの重量は実線文字で表示されている。ここで、図中に示されているように、最も重量の軽いビデオデッキを指示すると、そのVTRアイコンは"オブジェクト"状態のアイコンに変わる。本システムにおいて、オブジェクトに対応するアイコンはデータベース内のオブジェクトへのインタフェースとして機能する。すなわち、このアイコンを通じてオブジェクトに対してメッセージを送ることができる。オブジェクトでは、送られたメッセージに対応したメソッドが起動される。例えばメッセージ"!appear

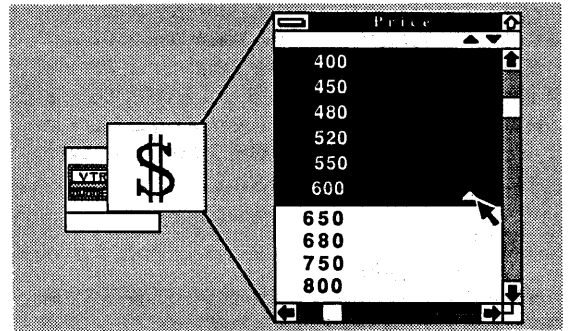


図8 問合せ記述 (問合せ2)

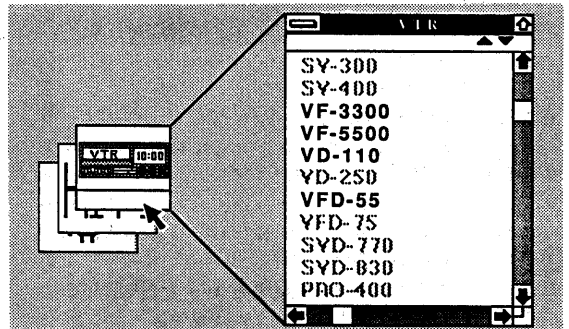


図9 問合せ記述 (問合せ3)

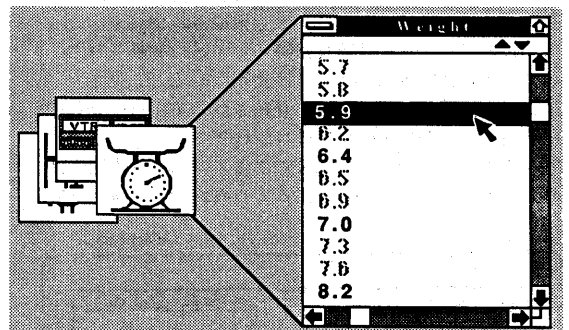


図10 問合せ記述 (問合せ4)

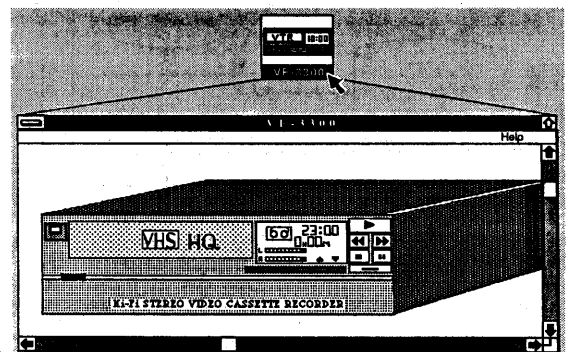


図11 ビデオデッキの外観の表示

nce'がオブジェクトアイコンに送られると、図11に示されているようにビデオデッキの外観が表示される。

問合せ5は、TV_setアイコンを検索エリアに選び出し、図12に示すように、TV_setアイコンとVTRアイコンとの両方の上にConnector_Cableアイコンを重ねることによって記述できる。この操作の結果、選ばれたビデオデッキに接続できるテレビが表示される。

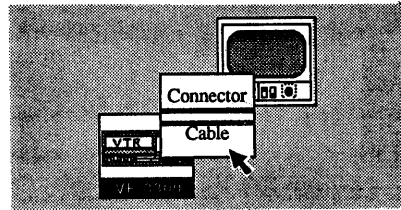


図12 問合せ記述 (問合せ5)

4. 実現

4.1 システム構成

提案するアイコンブラウザは、我々の研究室で開発しているオブジェクト指向データベースシステムMODE [6] 上に実現する。MODEはサーバ/クライアント形式のシステムである。サーバはオブジェクトマネージャと呼ばれ、クラスとオブジェクトの管理を行なう。一方、クライアントはオブジェクトスペースと呼ばれる。オブジェクトスペースはアプリケーションシステムを開発するための環境である。アイコンブラウザはそのアプリケーションシステムの一つとして実現されており、検索操作に応じてメッセージがオブジェクトスペースを通してオブジェクトマネージャに送られる。オブジェクトマネージャはデータベースを操作し、検索結果をオブジェクトスペースに返す。オブジェクトスペースではアイコンブラウザを通して結果をユーザに表示する。

アイコンブラウザのシステム構成を図13に示す。アイコンブラウザはInitial_Icon_Storage (IIS)、Icon_Interpreter (II)、Display_Manager (DM)、およびQuery_Generator (QG) から構成されている。アイコンブラウザを起動した時、IISに管理されているアイコンがアイコンエリアに表示される。ここで、IISには初期アイコンとして登録されたアイコンだけが管理される。それ以外のアイコンはアイコンブラウザのみがアクセスすることのできるオブジェクトとしてオブジェクトデータベースに蓄えられており、使用時に

オブジェクトデータベースから一時的なアイコンとしてアイコンブラウザ内に取り込まれる。アイコンの選択やアイコンの重ね合わせなどといったイベントはDMによって管理されている。それらのイベントはIIとQGに送られる。IIはイベントを解釈し、データベースからオブジェクトを検索するためにオブジェクトマネージャにメッセージを送る。オブジェクトマネージャから送られてきたデータはDMに送られ、画面に表示される。QGはアイコンによる問合せ操作に対応した内部的問合せ表現を生成する。

4.2 アイコンの管理

全てのアイコンはICONSクラスのオブジェクトとしてオブジェクトデータベース内に管理されており、各アイコンはオブジェクト識別子と呼ばれるデータベース内での唯一の値を持つ。ICONSクラスはサブクラスとしてClass_IconクラスおよびJoin_Iconクラスを持つ。

Class_Iconクラスは対象アイコンまたは制約アイコンになりうるクラスアイコンを管理する。クラスアイコンは以下の7つ組で定義される。

{イメージ, クラス名, パス, 代表項目, 問合せ, 制約集合, 結合アイコン}

各フィールドはそれぞれ次の意味を持つ。

イメージ: アイコンが画面上に表示される時の2次元イメージ。

クラス名: アイコンが対応するクラスの名前。

パス: アイコンが制約アイコンとして用いられているときの対象アイコンからのパス。

代表項目: アイコンが対応するクラスのインスタンス変数の一つで定義される。このインスタンス変数の値がreplace windowに表示される。

問合せ: アイコンの現在の状態を表現するための述語表現による問合せ。これはオブジェクトデータベース内ではnullであるが、適用された検索操作に従ってQGによって埋められる。

制約集合: アイコンのpop-up icon menuに表示されるアイコンの集合。表示されるアイコンの対応するクラスへのパスとそのアイコンの状態とが記述される。状態は"クラス"か"オブジェクト"のいずれかである。

結合アイコン: 結合アイコンのオブジェクト識別子

例として、VTRクラスのためのアイコンの定義を図14(a)に示す。VTRアイコンの代表項目はビデオデッキの名前で、図6に示されているように、その名前がreplace windowに表示される。pop-up icon menuが呼びださ

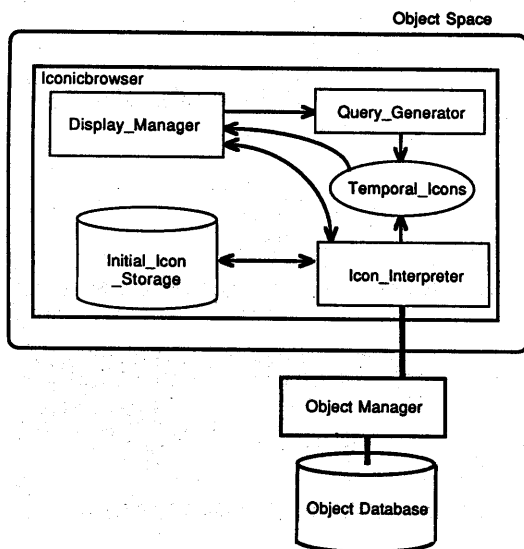


図13 システム構成

れたとき、制約集合のフィールドに定義されたパスVTR.priceおよびVTR.spec.weightが指しているクラス、つまりこれらのクラスに対応する値段および重量のアイコン、および結合アイコンのフィールドに定義された識別子を持つアイコンがpop-up icon menuに表示される。また、VTRクラスのスーパークラス、サブクラス、およびサブリングクラスに対応するアイコンは自動的に表示される。

一般に、オブジェクトの表現にはクラスのアイコンイメージが用いられる。しかしながら、クラスに属するオブジェクトをそれぞれ異なるアイコンイメージで表示したい場合がある。このような場合には、Class_IconクラスのサブクラスであるObject_Iconクラスの定義が用いられる。オブジェクトアイコンは識別子をClass_Iconクラスの記述に加えた形で定義する。Featuresクラスのオブジェクトに対応するHi-Fiアイコンの記述例を図14(b)に示す。ここで、Oid00875はオブジェクト識別子である。

ICONSクラスのもう1つのサブクラスであるJoin_Iconクラスは結合アイコンを管理するために用いられる。結合アイコンは以下の4つ組で定義される。

{イメージ, 結合クラス, 結合パス, 演算子}

各フィールドはそれぞれ次の意味を持つ。


イメージ: 結合アイコンが画面上に表示される時の2次元のイメージ。

結合クラス: 結合アイコンによって結合される2つのクラス。


結合パス: 結合演算によって比較される2つの変数へのパス。

演算子: 結合演算で使われる比較演算子。


例として、図12で用いたConnector_Cableアイコン

```
{
   , VTR, VTR, VTR.name, ,
  { (VTR.price,C), (VTR.spec.weight,C),
    (VTR.spec.length,C), (VTR.spec.depth,C),
    (VTR.spec.width,C), (VTR.vtr_feature,O) },
  {Oid00726}
}
```

(a) クラスアイコンの例

```
{
   , Features,
  VTR.vtr_feature, Features.name,
  SELECT Features FROM Features
  WHERE Features.name = "Hi-Fi" ,
  { (Features.name,O) }, , Oid00875
}
```

(b) オブジェクトアイコンの例

```
{
   , {TV_set, VTR},
  {TV_set.spec.connector,
  VTR.vtr_feature[?].name}, IN }
}
```

(c) 結合アイコンの例

図14 アイコンの管理

の記述を図14(c)に示す。結合クラスはTV_setクラスとVTRクラスである。結合パスはTV_set.spec.connectorとVTR.vtr_feature[?].nameである。ここで、"?"はワイルドカードを示す。演算子はINである。これらの情報から、TV_set.spec.connectorの値がVTR.vtr_feature[?].nameの値と同じであるテレビが選ばれる。

4.3 問合せの内部表現

提案したアイコンックブラウザにおいては、アイコンの重ね合わせによって記述された問合せはシステム内部でSQL風の形式に変換される。システム内部で生成された問合せ表現は、初期アイコンとして登録されたアイコンを対象アイコンとして用いたときのアイコンの初期状態を形成するために用いられる。さらに、これはMODE上の他のアプリケーション記述に用いることもできる。この中間言語の構文を図15に示す。

3.3で説明した問合せは中間言語を用いると次のように表現される。

問合せ1

```
SELECT VTR.name FROM VTR
```

問合せ2

```
SELECT VTR.name FROM VTR
WHERE VTR.price LE 600
```

問合せ3

```
SELECT VTR.name FROM VTR
WHERE VTR.price LE 600
```

```
AND VTR.vtr_feature[?].name EQ "Hi-Fi"
```

問合せ4

```
SELECT VTR.spec.weight FROM VTR
WHERE VTR.price LE 600
```

```
AND VTR.vtr_feature[?].name EQ "Hi-Fi"
```

問合せ5

```
SELECT TV_set.name FROM TV_set
WHERE TV_set.spec.connector
IN (SELECT VTR.vtr_featur[?].name FROM VTR
WHERE VTR.name EQ "VF-3300")
```

4.4 内部表現の生成手順

画面上の問合せ操作がIIによって受け取られた時、QGはその問合せに対応した中間言語表現を生成する。生成手順について以下に述べる。

アイコンが検索エリアに置かれて対象アイコンになった時、アイコンの問合せフィールドは'SELECT select_path FROM from_clause'で埋められる。select_pathとfrom_clauseはアイコンのクラス名フィールドの記述に置換される。Replace操作が対象アイコンに適用された場合には、select_pathは対象アイコンの代表項目フィールドの記述に置換される。また、Replace操作が制約アイコンに適用された場合には、select_pathは制約アイコンの対象クラスに関するパスフィールドの記述に置換される。

Add_Condition操作が対象アイコンに適用された時、'WHERE where_clause'が対象アイコンの問合せフィールドに追加される。さらに、条件がselectionであるならば、'path comparator constant'が追加される。pathは

```

query_expression ::= SELECT select_path expression
select_path      ::= path
path             ::= class_name [variable]
variable        ::= . instance_variable | variable . instance_variable
instance_variable ::= variable_name | variable_name['number']
number          ::= numeral | ? /* ? indicates wild card */
expression      ::= from_clause [where_clause]
from_clause     ::= FROM class_name
where_clause    ::= WHERE conditions
conditions      ::= boolean_term | conditions OR boolean_term
boolean_term    ::= boolean_factor | boolean_term AND boolean_term
boolean_factor  ::= [NOT] boolean_atom
boolean_atom    ::= scalar_expression comparator scalar_expression |
                  scalar_expression comparator subquery_expression

scalar_expression ::= constant | path
comparator        ::= EQ | NE | GT | GE | NGT | LT | LE | NLT | IN
subquery_expression ::= (SELECT select_path expression)

```

図 15 中間言語の構文

対象アイコンの代表項目フィールドの記述に置換される。comparatorとconstantはユーザの指定した演算子と値にそれぞれ置換される。条件がrestriction 操作によって記述されたならば、制約アイコンの間合せフィールドのwhere_clauseを対象アイコンの間合せフィールドに追加する。そして、select_path を対象クラスに関する制約アイコンのパスフィールドの記述に置換する。条件がjoinであるならば、'scalar_expression operationsubquery_expression'が対象アイコンの間合せフィールドに追加される。scalar_expression は対象クラスに関する結合アイコンの結合パスフィールドの記述に置換され、operation は結合アイコンの演算フィールドの記述に置換される。subquery_expression は、制約アイコンの間合せフィールドのselect_path を、結合アイコンで定義されたもう1つの結合パスで置換した記述に置き換えられる。

5. おわりに

本稿では、アイコンを用いてオブジェクト指向データベースの検索を行なうアイコンックブラウザを提案した。アイコンはデータベース内のクラスあるいはオブジェクトを表現する。間合せはアイコンに別のアイコンを順次重ねていくことによって記述される。システムは重ね合わせられたアイコンの組み合わせからその意味を解釈し、データベース操作を行なう。

提案したアイコンックブラウザはユーザの立場から見るとき、以下の3つの利点を持つ。

- 1 ユーザはアイコンのイメージから間合せの対象を容易に判断できる。
 - 2 ユーザはデータベースとそのスキーマに関する知識を持っていなくても間合せが行える。
 - 3 アイコンックブラウザはアイコン上の操作に従ってSQL風の間合せ表現を生成する。生成された間合せは他のアプリケーションに適用できる。
- しかしながら、今後解決しなければならない問題がいくつか残されている。それらの問題を以下にまとめる。

- 1 結合アイコンの定義されていないアイコン間でのjoin操作
- 2 OR条件で制約を記述する場合のアイコン表現

3 アイコンの定義を支援するアイコン定義ツールの提供

参考文献

- [1] M. C. Shu, Visual Programming, Van Nostrand Reinhold Company, New York, 1988.
- [2] S.K. Chang, T. Ichikawa, and P.A. Ligomenides (eds.), Visual Languages, Plenum Press, New York, 1986.
- [3] M. M. Zloof, "QBE/OBE: A Language for Office and Business Automation," IEEE Computer, Vol. 14, No. 5, pp. 13-22, May 1981.
- [4] M. Caplinger, "Graphic Database Browsing," Proc., ACM SIGDIS Conference on Office Information Systems, pp. 113-121, 1986.
- [5] J. A. Larson, "A Visual Approach to Browsing in a Database Environment," IEEE Computer, Vol. 19, No. 6, pp. 62-70, May 1986.
- [6] 津田, 山本, 平川, 田中, 市川, "MOREに基づくオブジェクト指向データベースシステム," 情報処理学会データベースシステム研究会資料, 88-DB-(6), 1988.