

解説



4. オブジェクト指向データベースの応用

4.3 エンジニアリング業務支援とオブジェクト指向データベース†

石川 博†† 泉田 義男†† 川戸 信明††

1. はじめに

技術者の仕事であるエンジニアリング業務全体の生産性と信頼性を向上させるには、従来の作図中心の CAD (Computer Aided Design) や CAM (Computer Aided Manufacturing) だけでは十分ではなく、それ以外の工程の支援や関連する二次的な技術情報の管理が重要になりつつある。一方現在、ビジネス分野において有効性の確認されているリレーショナルデータベースは、そのテーブル構造のモデルが固定的であるがゆえに、複雑なエンジニアリング業務には向かないとされ、それに代わる次世代のデータベースとしてオブジェクト指向データベースに対する期待が高まっている。しかしながら、有効性の確認ができるほど、この分野におけるオブジェクト指向データベースの適用事例は多くない。そこでこの論文では、エンジニアリング業務の特徴を分析し、そこから重要な要件を洗いだした後、それら各要件に対するいくつかの試みについて紹介しながら、エンジニアリング業務支援に対するオブジェクト指向データベースアプローチの有効性をボトムアップに論じていきたい。2. では、エンジニアリング業務の特徴とその支援に必要な要件の洗いだしを行う。3. ではオブジェクト指向データベースについて簡単に説明し、第一の要件である設計データ管理に対するオブジェクト指向データベースの適用について述べる。続く 4., 5., 6. では知的設計支援、技術文書管理、協調的作業の支援という要件のそれぞれに対するオブジェクト指向データベースを用いた試みについて述べる。

2. エンジニアリング業務支援

技術者の行うエンジニアリング業務は、企画、設計、実験解析、製造、生産計画管理と多岐にわたっている。しかしながら計算機による支援が行われているのは、そのごく一部に過ぎない。その代表的なものは設計の一次情報を扱う CAD とそのデータを使う CAM である。CAD 以前の工程である企画や概念設計、及びエンジニアリング業務全般に係わる二次的な技術情報の管理はほとんど支援されていない。これらは最近唱えられはじめた EOA (Engineering Office Automation) という分野とも重なりをもつ。そこでエンジニアリング業務全体の生産性と信頼性を向上させるためにこうした業務の計算機による支援を行う必要がある。以下では業務の特徴を分析し、エンジニアリング業務支援にとって必要な要件を洗いだす。

第一に設計は、設計対象の一次的データの作成管理を行うことである。そのためには、設計データの意味を記述する必要がある。特に対象の構成要素間の関係や類似したデータ間の関係を表現できる必要がある。また設計データ間には、設計仕様としての制約が含まれるので、そうした設計制約の表現やその充足手段、さらに設計手順というものを表現できる必要がある。これには従来の作図を中心とした CAD のほかにいわゆる知的設計支援といったものが必要になる。さらに設計書は文書データベースとして格納され、設計対象と関連して管理されるべきである。このようにしてはじめて設計知識の資産化が可能になる。さらに通常の場合、複数の設計者が一つの設計対象を作り上げるという点で協調的作業の支援が不可欠となる。特に異なる視点で同一物を眺めるということが重要になる。

設計以外の部分では、設計データに付随した二

† Object-Oriented Database Approaches to Engineering Tasks by Hiroshi ISHIKAWA, Yoshio IZUMIDA and Nobuaki KAWATO (Artificial Intelligence Laboratory, FUJITSU LABORATORIES LTD.).

†† (株)富士通研究所人工知能研究部

次の情報（いわゆる技術文書）の作成、検索、管理が必要になる。設計者といえどもこの種の仕事が業務の多くを占める。特にマニュアル作成支援などは、データの量とデータの種類（文字数値、図面、実験データの表やグラフなど）及びそのライフサイクルの長さからいって重要な分野である。そうした異種メディアのデータは統合される必要があり、それらの関連を管理し、それを柔軟にアクセスできる必要がある。さらにこれらの情報は作成後に変更が起きる可能性があるため、関連するデータ間の一貫性を管理することが不可欠となる。当然ほかの技術者の作った情報を自分の見方で参照することも必要になる。

以上のことをまとめると、以下の4つの要件になる。

- 設計データ管理

複雑なデータの構造（部品関係）を直接表現し、かつ類似したデータの抽象化を支援する。さらに大規模データの効率的な管理ができる。

- 知的設計支援

設計知識としての設計制約の記述、充足を含めた管理ができる。さらに関連するデータ間の一貫性制約の管理ができる。

- 技術文書管理

マルチメディアからなる技術情報及びそれらの間の関連の管理とその柔軟なアクセスができる。

- 協調的作業の支援

複数の技術者が同一データを異なる見方で参照できる。

そこで以下ではこれらの要件についてオブジェクト指向データベースをもちいた試みを紹介していく。

3. 設計データ管理

ここではまずオブジェクト指向データベースの基本的特徴をまとめておこう。オブジェクト指向データベースはオブジェクト識別子、複合オブジェクト、クラス階層と遺伝、ポリモルフィズム、メソッドの動的束縛、プログラミング機能、問い合わせ言語、永続的オブジェクトなどの機能を支援するデータベース管理システムと考える¹⁾。オブジェクト指向データベースは後述するすべての技術（サブシステム）の共通プラットフォームになりうる。

設計データ管理では、設計対象の複雑な部品関係を直接表現し、同じようなデータはまとめる抽象化の機能が求められている。これらの機能はオブジェクト指向データベースの複合オブジェクトとクラス階層で基本的には実現できる。たとえば IRIS というオブジェクト指向データベースを用いて、アセンブリ (Assembly) やその部品 (Part) をモデル化する試み²⁾がある。図-1 はそのクラス構造を表現したものである。Document はオンライン文書を表示している。Assembly は個々の Document, Part, subAssembly からなる複合オブジェクトである。Process は Assembly を入力・出力とする処理に関する情報を管理する。Document も Part も単独で Assembly となるため Assembly のサブクラスとして定義する。Assembly が sub-Assembly からなるという部品関係は、この例では Assembly のもつ再帰的な関係とみなし、Process の入出力関係をもちいて導出している。この研究では、同一のデータをオブジェクト指向データベースのほかにリレーショナルデータベースをもちいてモデル化もしており、両者の比較では、オブジェクト指向データベースのほうがスキーマの量の少なさと分かりやすさの点でリレーショナルデータベースより優れていると報告している。

さらに部品関係に付随して、複合部品と単体部品の間になり立つ一貫性制約の管理を行わなければならない場合がある。たとえば単体部品は複合部品が存在する間のみ存在できるというような制約を、Jasmine というオブジェクト指向データベースがもつデモン機能をもちいて実現するという試み³⁾がある。デモンはオブジェクト指向データベースの必須機能ではないが、属性やメソッドに付加していわゆる active database の実現に利用できる。この制約はすなわち複合部品の生成メソッドの after デモン（メソッド本体の起動後に起

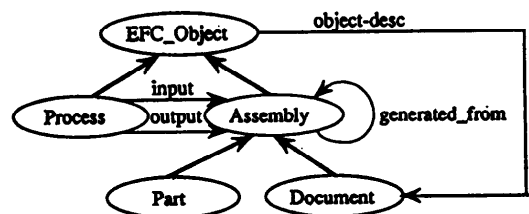


図-1 IRIS によるアセンブリとプロセスのオブジェクトモデル¹⁾

動されるデモン)と消去メソッドの before デモン(メソッド本体の起動前に起動されるデモン)として、それぞれ単体部品を生成するプログラムと単体部品を消去するプログラムを記述することにより実現できる。また複合部品の重量は単体部品の重量の合計として計算されるが、それも複合部品の重量属性の if-needed デモン(属性が参照されると起動されるデモン)として実現される(図-2参照)。これにより複合部品の重量は常に最新の重量に保たれる。

設計データの動的性質もオブジェクトのメソッドとして表現できる。たとえば図-2でモジュール式は歯車の適正なモジュール値を計算で求めるメソッドである。オブジェクト指向データベースではさらにメソッドを組み合わせて応用のプログラミングを行ったり、とくに後述するように Jasmine ではメソッドを問い合わせのなかで起動し、高度な検索などもできる。

複合部品	Super	COMPOSITE
Property	STRING	名称
単体部品	Part	multiple
FLOAT	重量	if-needed (self. Part. 重量. sum())
COMPOSITE-MEDIA 図面		

タイミングギア	Super	複合部品
Property	歯車	Part
FLOAT	中心間距離	

歯車	Super	単体部品
Property	FLOAT	モジュール
INTEGER		歯数
Method	FLOAT	モジュール式 (F, sigma-k, y) {square-root(F/(5*pai*sigma-k*y))}

図-2 Jasmine による設計対象のオブジェクトモデルの例¹⁾

モジュール	dependent-parameter	F sigma-k y
design-method		retrieve-increase (標準モジュール)
constraint		モジュール>モジュール式 (F, sigma-k, y) advice (モジュール. increase)
increase		retrieve-increase (標準モジュール)

図-3 設計ゴールの例¹⁾

複合オブジェクトのアクセスに関しては、常に結合演算を必要とするリレーショナルデータベースに対して、直接的なリンクを利用するオブジェクト指向データベースのほうが、効率が良いと考えられるが、実際の報告はまだされておらず、今後の研究が待たれるところである。

4. 知的設計支援

設計における作図の支援を行う従来 CAD にはかなり高度なものがあるが、それだけでは十分ではない。設計作業には、属性に対する条件の集まりからなる設計制約の表現と充足が必要になる。設計制約は一つの属性の値に関する条件や、複数の属性間の条件として表現される。さらにこれらの設計制約を充足するための設計メソッドを表現でき、それに基づいて設計解を求めることができる推論機構を提供する必要がある。

こうした知的設計支援に対するオブジェクト指向データベースアプローチとして筆者らの研究³⁾を紹介する。設計対象オブジェクトのほかに制約を管理するオブジェクト(設計ゴールという)を定義する。設計ゴールには設計制約の記述とその制約の充足手段の記述及び充足失敗時のアドバイスを記述できる。設計制約としては、設計属性の間の制約条件を等式・不等式で表現したものが指定できる。充足手段としては、表(データベース)の検索、ほかの属性からの計算、生成、ユーザの入力が指定できる。失敗時のアドバイスとしては、自他の設計ゴールを呼び出すことができる。設計者が制約充足時のバックトラックを指定できるようになっているが、これは設計制約だけでなくその充足の手段や失敗時の処置方法についても設計者のノウハウの一部と考えられるためである。たとえば図-3は歯車のモジュールを決定するための設計ゴールである。そこにはモジュールの値は計算によって求めたモジュール値(ルイスの式に従う)より大きくなければならないという制約と、その充足メソッドとして標準モジュール表の検索をすること、制約充足失敗時のバックトラックゴールとして現状値より大きい値をとるよう自分のゴールにメッセージを送信することが記述されている。

5. 技術文書管理

技術文書の作成・検索及びそれにとまなう管理は、エンジニアリング業務のかかなりの部分を占める。したがってその部分を支援することは業務全体の生産性と信頼性を向上させることにつながる。技術文書の特徴は、マルチメディアデータ（たとえばテキスト、図、表など）とそれらの間の有機的な関連づけがあることである。よってこれらに対してはハイパメディアのアプローチが有効である。NoteCards⁴⁾はハイパメディアシステムの代表的なものであり、データをノードとリンクで表現する単純なモデルと、そのリンクをたどる検索の仕方がエンドユーザに受け入れられている。しかしながら、現状のハイパメディアシステムの多くは、複雑で大規模なエンジニアリング業務に適用しようとする、複合オブジェクトのモデル化、大規模ノードデータの効率的な管理、内容検索や構造検索を含む柔軟な検索機能、応用の進化にとまなう変化の吸収、プログラミング機能、分散協調、バージョン機能などの点で問題がある。

これらの課題に対してはハイパメディアのなかでも努力が払われている。その一つは Neptune というハイパメディアシステムの研究^{5),6)}であり、

大規模データの効率的な検索管理、及び分散環境でのデータの共有を可能とするために、HAM (Hypertext Abstract Machine) というハイパメディアのノード・リンクの格納とアクセス（リンクをたどる検索機能と条件指定による問い合わせ機能）を専門に行うデータベースシステムを作っている。ユーザは HAM の上に応用のレイヤを作成する。さらにユーザインタフェースの部分はオブジェクト指向プログラミング言語である Small-talk-80 を用いて実現している。図-4 はテキスト内のノードをたどって別のノードにアクセスしているところを示している。Intermedia というハイパメディアシステムの研究⁷⁾では Encore というオブジェクト指向データベースと InheritanceC というオブジェクト指向プログラミング言語を用いて、システムを実現しようとしている。図-5 はデータのブロック（ノード）とそのリンクをオブジェクトによって表現したものである。ブロックをテキスト中でハイライト表示したり、リンクを選択したりする手続きはブロックオブジェクトのメソッドとして定義され、メッセージ送信により Encore に格納されたオブジェクトが操作される。その結果、複雑なデータ構造をオブジェクトで直接表現できる点、一つのメソッド（メッセージ）

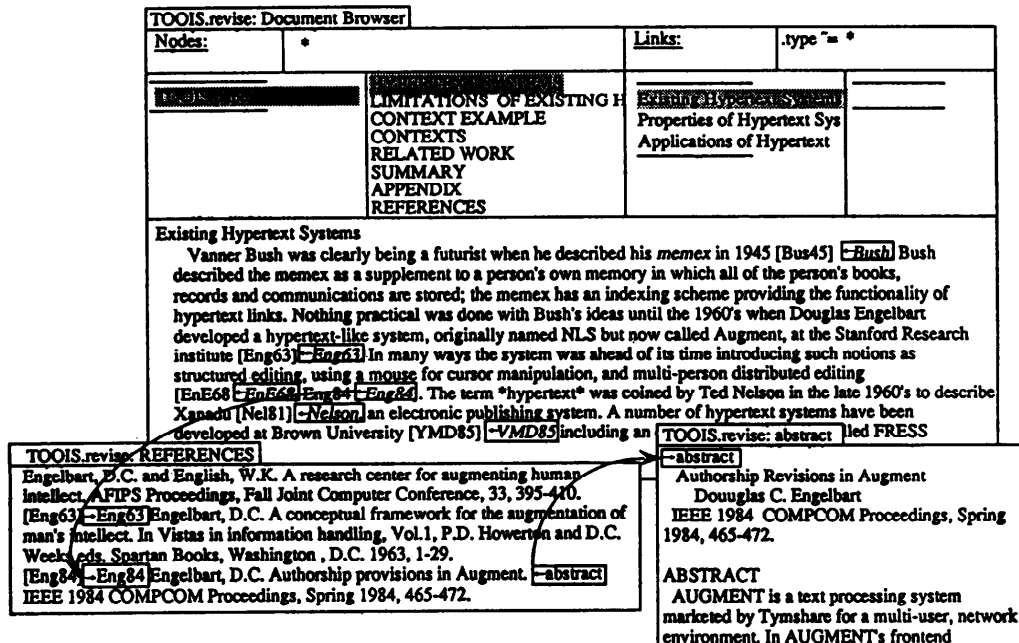


図-4 Neptune によるハイパメディアアクセスの例⁴⁾

によって複数の問い合わせと計算をひとまとめに実行できる点、プログラム中のデータとデータベースとの間のインタフェースにおけるインピーダンスミスマッチを小さくできる点、及び一貫性制御をメソッド起動時にできる点で、オブジェクト指向データベースを使うアプローチはリレーショナルデータベースを使うアプローチに比べ優れていると主張している。さらに Neptune と Intermedia はともに簡単なバージョン機能とユーザのプライベートな作業空間を作るための仕組みを備え、データの協調的利用ができるようになっている。

<pre> Define Class Block Superclasses: Object Properties: content: set of Object container_doc: Document marker: Marker explainer: Text create_date: Date modify_date: Date owner: User_ID key_words: set of Text links: set of Link Methods: initialize (B: Block) highlight (B: Block) delete (B: Block) choose_link (B: Block) </pre>	<pre> Define Class Link Superclasses: Object Properties: block1: Block block2: Block explainer: Text create_date: Date modify_date: Date owner: User_ID key_words: set of Text Methods: initialize (L: Link) follow (L: Link, B: Block) </pre>
--	--

図-5 Intermedia におけるブロックとリンクのクラス定義

図-6 TextLink におけるリンク定義の例

現状のハイパメディアの問題点に対してオブジェクト指向データベースを使ったより進んだアプローチとしては、TextLink の研究⁹⁾と筆者らによる Jasmine を使った研究⁹⁾がある。TextLink では、データが大規模化したときにリンクを張る作業に膨大な労力を要するようになるという問題点を解決するため、ノード間に静的なリンクを張る代わりに、リンク自身をデータベースの問い合わせで表現するというアプローチを採用している。LDL と呼ばれるリンク記述言語を提供し、リンクのソースとターゲットを LDL を用いて表現する。たとえば図-6 は TextLink クラス及びそのサブクラスに属するテキストオブジェクトをその属性値の条件で制限してリンクのソースを定義している。リンクの記述では、クラスとそのサブクラスも同時に指定できるようになっているため、詳細化の方向へのクラス構造の進化に対しては、ある程度対応できるようになっている。一方筆者らの研究では、前述したように複合オブジェクトの制約に関しては Jasmine のデモン機能を用いて実現している。応用の進化にともなう変化の吸収という問題に対しては、メソッドの中に問い合わせを記述することができるようにして、仮想的なノードやリンクを実現できるようにしている。構造検索や内容検索に対しては、それぞれ専用のメソッドを問い合わせの中で起動することにより解決している。図-7 において(a)は仮想リンクの定義を、(b)はその利用を、また(c)と(d)はそれぞれ構造検索と内容検索の例を示している。また協調作業にともなう変更通知や変更の自動伝播に対してもデモン機能を利用した考えを提案している。

マニュアル管理という応用に対しては、Jasmine を用いた MASCOT システムの研究¹⁰⁾がある。文書の節、大項目、中項目、小項目、ブロックという部品階層をクラスとして表現し、それらはすべてテキストと図を内容としてもちうることから

- (a) procedural SCENIC_SPOT neighborhood () multiple
(SCENIC_SPOT where SCENIC_SPOT. within (self, 3)==true)
- (b) SHRINE. neighborhood () where SHRINE. Name=="The Heian Shrine"
- (c) OBJECT where OBJECT. Author. count ()==2
GUIDEBOOK. Section where
- (d) GUIDEBOOK. Section. Paragraph. contains ("This*")==true
and SECTION. Author=="ISHIKAWA"

図-7 Jasmine における仮想リンクの定義と利用及び、構造検索と内容検索の例

内容属性をもつ論理構造を上位クラスとして定義し、その属性を遺伝している。テキスト内で使う数字は、関連する表を検索する問い合わせで表現し、関連するデータ間の一貫性を維持するようにしている。図-8の(a)と(b)はマニュアルのクラス構造とそのインスタンスオブジェクトの例を示している。図-8の(c)はテキスト中で参照する数値を別の表から検索して埋め込む例を示している。さらにその値がオブジェクト指向データベースにない場合、外部のデータベースまたはファイルからデータを検索することも指定できる。オブジェクトが永続的かどうかを意識せずに応用をプログラミングできる点、問い合わせとメソッドを併せて使うことができる点、及び遺伝とポリ

モルフィズムが利用できる点で、応用システム開発における生産性の向上にオブジェクト指向データベースが寄与していると主張している。

マルチメディア文書の構造の表現に対してオブジェクト指向データベースを用いた研究¹¹⁾がある。図-9は、メモがヘッダと本体からなり、本体が段落、図形、画像からなるという部品関係を直接表現している。メモは文書のサブクラスとして定義されている。このオブジェクト指向データベースではクラス階層にそった遺伝のほか、部品関係にそった遺伝も許しており、この例では、段落が親部品である文書(メモ)からそのフォントサイズを遺伝してくる。オブジェクトに対する操作としては、メモは send、図形は draw という

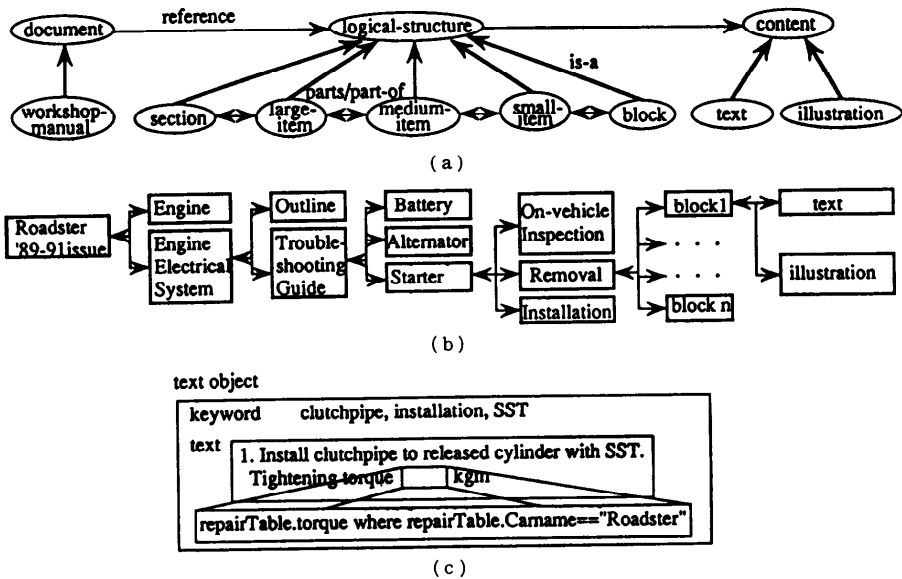


図-8 Mascot におけるマニュアルのクラス構造とインスタンスの例、及び問い合わせを含むテキストオブジェクトの例¹¹⁾

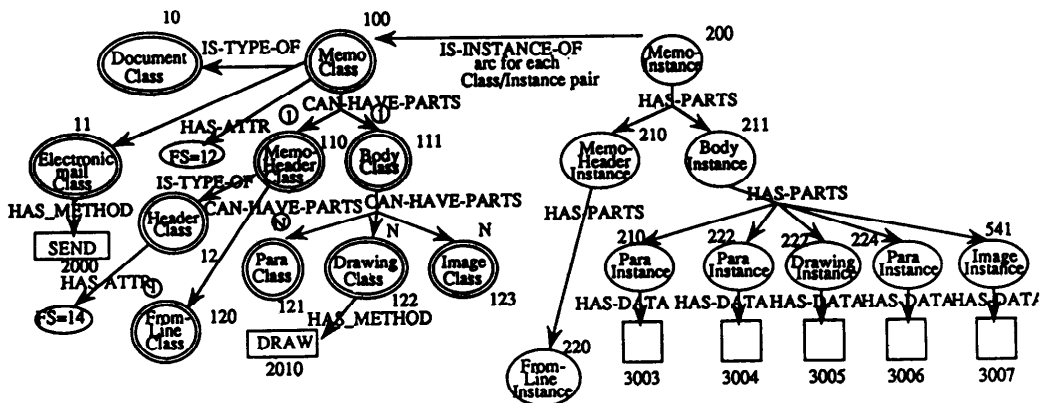


図-9 Orion によるマルチメディア文書のクラス構造とインスタンスの例¹¹⁾

メソッドをもつ。

オブジェクト指向データベースを用いた応用システムのユーザインタフェースの研究としては、O₂ とよばれるオブジェクト指向データベースの研究¹²⁾がある。O₂ には LOOKS というユーザインタフェースジェネレータが備わっており、ユーザが複合オブジェクトの表示と編集を行うインタフェースを、システム定義の汎用の対話用コンポーネントをカスタマイズすることによって容易に作成できる。表示では複合オブジェクトの展開

の深さを指定することができる。編集時のコピーやペースト機能では、データの型チェックを行うことができる。さらにメソッドの対話的な起動やデータベースのブラウジングや問い合わせを行うことができる。図-10 は複合オブジェクトの表示の例である。LOOKS は、O₂ における値とオブジェクトを生成編集するための機能を提供するユーザインタフェースサーバとして実現されており、クライアントの応用とはリモートプロシジャコールで通信を行う。

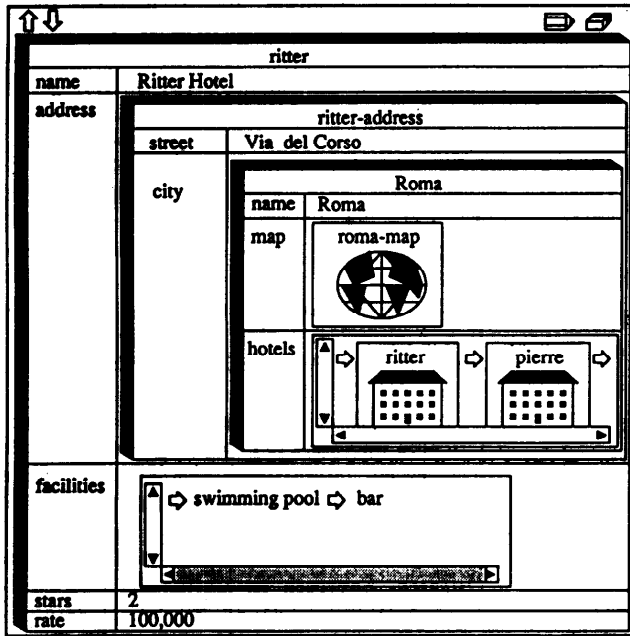
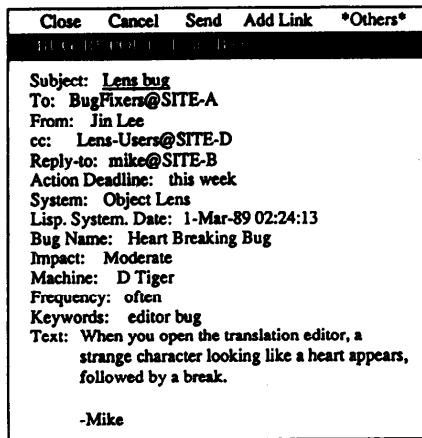


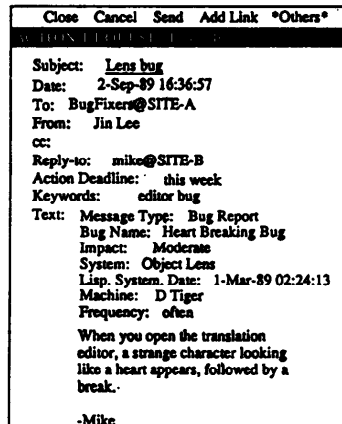
図-10 O₂ における複合オブジェクトの表示の例¹¹⁾

6. 協調的作業の支援

一般には複数の設計者が協調的に作業して一つの対象を設計する。その前提となるのはほかの設計者の設計したデータの参照である。しかしその際に作成した側と参照する側が同一の見方をするとは限らない。そこで同一オブジェクト（設計データでも技術情報でもよい）を異なる見方で参照できるメカニズムが必要になる。これには自律分散型データベースの研究¹³⁾や構造をもったメッセージ送信システムの研究¹⁴⁾が参考になる。ここでは後者の例として MIT の ObjectLens を取り上げる。クラス階層をいくつかのフラグメントにわけ、そのフラグメントの集合をユーザグループに対応させる。そこで一つのグループのユーザが別のグループの作成したオブジェクトを参照し



(a)



(b)

図-11 ObjectLens におけるオブジェクト変換の例¹⁴⁾

た場合に、もしまったく同一の構造をもつオブジェクトが自サイト内にはない場合は、参照されるオブジェクトの属性の部分集合をもつクラスを自サイトに探し、そのクラス（それは元のクラスの上位クラスになる）のオブジェクトに変換する。変換先の上位クラスにない属性の値は、その特別な属性（テキスト属性）のヘッダ部分に属性名と属性値という対の形で、テキストに変換されて挿入される。もちろん陽にオブジェクトを別オブジェクトに変換するルールも記述することができる。図-11の(a)は元のオブジェクト、(b)は自動的に上位クラスに変換された結果のオブジェクトである。

7. おわりに

エンジニアリング業務の支援を行うために解決すべき課題を述べ、それらに対するオブジェクト指向データベースを用いた具体的な試みについて述べてきた。ここで述べた課題は包括的ではないが、いずれも重要なものであり、具体例も代表的なシステムや研究の成果に基づいている。しかしオブジェクト指向データベースは、現在システムそのものの研究開発が行われている段階であり、それを実際の問題に適用した例はきわめて少ないため、エンジニアリング業務支援に対するオブジェクト指向データベースによるトータルなアプローチを描くことはできなかった。そこでオブジェクト指向データベースアプローチの有効性を実証するためだけでなく、システムの問題点の発見とその改良を行うためにも、今後とも実例に関する報告がでてくることが強く望まれる。ここではほとんど触れなかったものに、CAD そのものに対するオブジェクト指向データベースを用いたアプローチ（統合データベースやバージョン管理などの技術を含む）があるが、これは本特集の別のところで解説されているのでそちらを参照していただきたい。

参考文献

1) Atkinson, M. et al. : The Object Oriented Database Manifesto, Proc. The First International Conference Deductive and Object-Oriented Database, pp. 40-57 (1989).

- 2) Ketabchi, M. A. et al. : Comparative Analysis of RDBMS and OODBMS : A Case Study, Proc. IEEE COMPCON Spring 90, pp. 528-537 (1990).
- 3) 石川, 泉田, 川戸 : エンジニアリング分野におけるデータベースフレームワークについて, 情報処理学会, データベースシステム研究会, 78-17 (1990).
- 4) Halasz, F. G. : Reflections on NoteCards : Seven Issues for the Next Generation of Hypermedia Systems, ACM CACM, Vol. 31, No. 7, pp. 836-852 (1988).
- 5) Delisle, N. and Schwartz, M. : Neptune : A Hypertext System for CAD Applications, Proc. ACM SIGMOD, pp. 132-143 (1986).
- 6) Delisle, N. and Schwartz, M. : Contexts—A Partitioning Concept for Hypertext, ACM Trans. Office Information Systems, Vol. 5, No. 2, pp. 168-186 (1987).
- 7) Smith, K. E. and Zdonik, S. B. : Intermedia : A Case Study of the Differences between Relational and Object-Oriented Database Systems, Proc. ACM OOPSLA 87, pp. 452-465 (1987).
- 8) 平山, 西川, 難波, 田中 : リンク定義言語を有するハイパーテキストシステム : TextLink-III, 情報処理学会, データベースシステム研究会, 78-7 (1990).
- 9) Ishikawa, H. : An Object-Oriented Knowledge Base Approach to a Next Generation of Hypermedia System, Proc. IEEE COMPCON Spring 90, pp. 520-527 (1990).
- 10) Kuwano, N. et al. : Applications of Object-Oriented Databases to Publishing Systems, Proc. The Second International Symposium Database Systems for Advanced Applications, pp. 421-429 (1991).
- 11) Woelk, D., Kim, W. and Luther, W. : An Object-Oriented Approach to Multimedia Databases, Proc. ACM SIGMOD, pp. 311-325 (1986).
- 12) Deux, O. et al. : The Story of O₂, IEEE Trans. Knowledge and Data Engineering, Vol. 2, No. 1, pp. 91-108 (1990).
- 13) Heimbigner, D. and McLeod, D. : A Federated Architecture for Information Management, ACM Trans. Office Information Systems, Vol. 3, No. 3, pp. 253-278 (1985).
- 14) Lee, J. and Malone, T. W. : Partially Shared Views: A Scheme for Communicating among Groups that Use Different Type Hierarchies, ACM Trans. Information Systems, Vol. 8, No. 1, pp. 1-26 (1990).

(平成2年12月11日受付)



石川 博 (正会員)

昭和31年生。昭和54年東京大学理学部情報科学科卒業。同年(株)富士通研究所入社。以来関係データベース、データベースの自然言語インタフェース、オブジェクト指向データベースの研究開発に従事。現在オブジェクト指向データベースのエンジニアリング分野への応用に興味をもつ。ACM, IEEE 各会員。



川戸 信明 (正会員)

昭和23年生。昭和46年東京大学工学部電子工学科卒業。昭和51年同大学院工学系研究科電子工学専門課程博士課程修了。工学博士。同年(株)富士通研究所入社。以来、論理装置のCADの研究開発に従事。AI技術や並列処理技術のCADへの応用に特に興味をもつ。電子情報通信学会、日本ソフトウェア科学会、人工知能学会、IEEE 各会員。



泉田 義男 (正会員)

昭和27年生。昭和50年横浜国立大学工学部電気工学科卒業。昭和52年同大学院工学研究科電気工学専攻修士課程修了。同年(株)富士通研究所入社。以来、リレーショナル・データベース、自然言語インタフェース、知識表現ツールの研究開発に従事。現在、オブジェクト指向データベースとその応用及びデータベース・マシンの研究を行っている。電子情報通信学会、人工知能学会、ACM, IEEE 各会員。

