

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ついて

石川博 泉田義男 川戸信明
株式会社富士通研究所

新しいデータベースの応用として、エンジニアリングなどの複雑かつ大規模な応用分野が重要になりつつある。例えばエンジニアリング業務全体の生産性と信頼性の向上には、従来の作図中心のCADやCAMだけでは十分ではなく、対象が持つ複雑な構造や関連を表現することと、設計制約等の手続き的知識を記述することが必要になる。こうした目標を達成する高度な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。この論文では、我々が開発したプロトタイプオブジェクト指向データベースシステム Jasmine をエンジニアリング業務支援に対して適用した事例を用いて、そのアプローチの有効性について論じる。

ON APPLICATION OF AN OBJECT-ORIENTED DATABASE SYSTEM TO ENGINEERING

Hiroshi Ishikawa Yoshio Izumida Nobuaki Kawato
Fujitsu Laboratories Ltd.
1015 Kamikodanaka, Nakahara-Ku, Kawasaki 211, Japan

Complex, large-scale applications such as CAD require advanced databases. For example, conventional CAD systems do not sufficiently improve the total productivity and reliability of engineering tasks. To this end, we must represent complex structures and behaviors of design objects directly. Object-oriented databases are expected to better meet such requirements than relational databases. In this paper we discuss an application of a prototype object-oriented database system called Jasmine to engineering to verify the validity of the approach.

1. はじめに

新しいデータベースの応用として、エンジニアリングやハイパーメディアといった複雑かつ大規模な応用分野が重要になりつつある。例えばエンジニアリング分野への応用を取り上げて、考えてみよう。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、それ以前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。その特徴は、対象が持つ複雑な構造や関連を表現することと、設計制約等の手続き的知識を記述することが必要になることである。こうした目標を達成するためのソフトウェア・プラットフォームとして高度なデータベースが必要とされている。

一方、リレーショナルデータベースは現在ビジネス分野において広く有効性が確認されているが、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、それに替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 (例えば概念としては [ATKI89] システムとしては Jasmine [ISHI90a] [AOSH90] [ISHI91a], GemStone [MAIE86], Iris [WILK90], Orion [KIM90], O₂ [DEUX90]) に対する期待が高まっている。そこでこの論文では、我々が開発したプロトタイプオブジェクト指向データベースシステムである Jasmine をエンジニアリング業務支援に対して適用した事例について説明しながら、オブジェクト指向データベースアプローチの有効性について論じていきたい。Jasmine については、マニュアル作成支援という部分分野への応用 [KUWA91] [YAMA90] もあるが、ここでは述べない。

本論文は以下のように構成されている。第2章ではエンジニアリング業務の支援に必要な要件として設計データ管理、知的設計支援、技術文書管理の3つについて述べる。続く3、4、5章で各要件に対するオブジェクト指向データベース Jasmine を用いた試みについて説明する。第6章で今後の課題について論ずる。

2. エンジニアリング業務支援という応用

この章ではエンジニアリング分野における要件について分析する。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渡っている。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。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、データベースという観点からエンジニアリング業務支援にとって必要な要件を洗い出す。

設計の第一義は、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の他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。設計知識の資産化にはこうした要件が前提となる。

エンジニアリング業務における設計以外の部分では、設計データに付随した二次的情報 (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。技術文書のデータ量は膨大であり、データの種類も多い (例えば文字数値、図面、実験データの表やグラフ等)。

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。この要件はPIMS(Product Information Management System)やEOA(Engineering Office Automation)といった分野と重なりを持つ。

以上の要件をまとめると、以下の3つになる。

- ・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。さらに複数の設計制約の充足の流れを制御するような設計手順の管理ができる。

- ・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。さらに関連するデータ間の一貫性制約の管理（維持）ができる。

これらの要件は、包括的ではないが重要である（例えばここでは触れないが、複数の設計者による協調的作業の支援も重要な要件の1つである〔ISHI91b〕）。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Jasmineをもちいた具体的試みを説明する。

3. 設計データ管理

ここでは設計データ管理という要件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について説明する。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（汎化関係）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。例えばオブジェクト指向データベースシステムJasmineを用いて、部品とその構成要素をモデル化する試み〔ISHI90b〕〔ISHI91a〕がある。図1はそのクラス構造を表現したものである。ユニット（UNIT）の部品（Part）属性は、ユニットが複数の構成部品（COMPONENT）から構成されていることを示す。ピストン（PISTON）のSuper属性は汎化関係を直接表現し、この部品が上位の構成部品の属性を遺伝することを示す。このようにオブジェクト指向データベースは、スキーマの冗長性を少なくし、かつ部品関係を直接的に表現できて分かり易いという特徴を持つ。

ここで重要なことは、Jasmineは部品関係に対してシステム定義の意味付けを内蔵しないことである。なぜなら汎化関係に伴う遺伝が応用によらず一定の意味を持つものに対して、部品関係に対して各応用の与える意味付け、即ちどんな機能を期待するかということとは異なるからである。そのかわり、Jasmineではデモンとしてユーザが属性やメソッドに応用独自の意味付けを与えることを許している。デモンは一般にはオブジェクト指向データベースの必須機能とはされないが、属性やメソッドに付加していわゆるアクティブデータベース〔MORG83〕の実現に利用できる。即ち部品関係に付随して、ユニットと構成部品間に成り立つ一貫性制約の管理（維持）を行わなければならない場合に特に重要である。例えば構成部品はユニットが存在する間のみ存在できるというような制約（ただし流用設計では、部品が先に存在することが前提なのでこの制約は成り立たない）を、Jasmineが持つデモン機能を利用して実現するという試み〔ISHI90b〕〔ISHI91a〕がある。この制約はユニットの生成メソッドのafterデモン（メソッド本体の起動

後に起動されるデモン)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、それぞれ構成部品を生成するプログラムと構成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。またユニットの重量は構成部品の重量の合計に等しいという制約は、ユニットの重量属性の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1参照)。これによりユニットの重量は常に最新の重量に保たれる。このように、デモンは応用依存の意味である一貫性制約を維持するのに役立つ。このことは、リレーショナルデータベースや他のオブジェクト指向データベースでは難しい。

エンジニアリング業務全体を記述することにオブジェクト指向データベースは利用できる。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。例えば図1でピストンの最大曲げ応力(MaxBendingStress)メソッドはピストン頭部の受ける最大曲げ応力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はそうした応用依存の操作を定義できるだけでなく、検索・更新といった汎用的なデータ操作もオブジェクトとしてカプセル化して提供する。応用にはデータ操作が含まれるが、その中で最も重要なのがデータの柔軟な検索である。Jasmineは集合的な検索機能を有する操作言語を提供しているので、大量で多様な設計データの中から、必要な部品を効率的に条件検索できるようになっている。例えば以下は「11,000g以下の質量で、その部品としてコンロッドを持つユニットを求めよ」という問い合わせである。

```
UNIT where UNIT.Mass <= 11000
and UNIT.Part.Name = "conrod"
```

オブジェクト指向データベースJasmineでは後述するように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。

複合オブジェクト(例えばエンジニアリング分野における部品)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とくらべ、オブジェクトバッファ[ISHI91a]上で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方が、効率がよいと考えられる。このように定義された大規模な設計データはシステムにより前述したように2次記憶を用いて効率的に管理される。開発面而言えば、メソッドを問い合わせに指定することができ、オブジェクトの永続性を意識せずに応用をプログラミングできる点で応用開発の生産性が高くなる。またクラス階層を通した遺伝と複合オブジェクトの定義によるユーザ拡張性はインクリメンタルな設計に役立つ。

4. 知的設計支援

ここでは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について筆者らの研究[ISHI90b][ISHI91a]を紹介しながら、その有効性を具体的に論じていきたい。設計における作図の支援をおこなう従来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(作成)と充足が必要になる。設計制約は1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もとめることができる推論機構を提供する必要がある。設計制約は独立ではなく、互いに依存しているので、その充足の順序を含めて設計手順の管理も同時に必要である。

設計制約は単にチェックするためだけにあるわけではない。制約条件が最初から満たされるわけではないので、制約が満たされない場合に、どうするかということもユーザが記述できなくてはならない。そこで設計対象オブジェクトの他に制約を管理するオブジェクト（設計ゴールという）を提供する。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、他の属性からの計算、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。推論は、設計ゴールとその間の依存関係からなるネットワークに基づいて行われる。推論の前処理で、設計ゴールのネットワークを作成し、その際にエラーチェック（例えばループの検出等）を行う。例えば図2の(a)は設計ゴールのネットワークの一部である。さらに図2の(b)はピストンの頭部肉厚を決定するための設計ゴールである。そこには制約の充足メソッド（DesignMethod）として初期値と増分から値を生成をすること、生成された頭部肉厚の値が満たさなければならぬ制約（Constraint）が記述されている。さらに制約(3)には、制約充足失敗時のバックトラックゴール（Advice）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か、またはピストン内径のゴールに値を減らすようにメッセージを送信することが記述されている。

我々が、ユーザがバックトラックを直接制御するというこの方式（これを知的バックトラックという）を採用した理由について述べる。別の方式として制約プログラミングという手法〔LELE88〕があるが、そこでは制約条件を記述するだけで、その後の制約充足はすべてシステムが行う。即ち制約を満たす解をみつけるメカニズムはブラックボックスである。しかしながら、設計においては制約充足の失敗後の処置もふくめて設計のノウハウと考えられるので、ブラックボックス化は不都合であり、そのかわり設計ゴールのように設計ノウハウをユーザが直接記述でき、システムがそれを解釈するという仕組みが必要である。もちろんオブジェクト指向データベースそのものは、制約管理機能を提供しないが、このようにオブジェクト指向データベースの拡張性を利用して制約をオブジェクトとして追加することにより実現できる。

設計対象を表わすオブジェクトと図面のオブジェクトとは相互に関連しあっている。即ち求められた設計解に基づき、設計対象に関連する図面（グラフィクスデータ）が生成される。さらに各部品間に成り立つ結合条件や運動制約などの幾何的制約を解決することにより、一連のアニメーションを作成することができ、概念設計における簡単なシミュレーションが可能となる。例えば図3の(a)はコンロッドとクランクの結合を定義する幾何的制約の例を示し、図3の(b)はそれらを解決して作成したアニメーションの1コマである。図面は、関連する設計データが変更されると、デモンによって自動的に変更される。

5. 技術文書管理

エンジニアリング業務のかなりの部分を占める技術文書の作成・検索及びそれにともなう管理という仕事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（例えばテキスト、図、表等）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ーメディアのアプローチが有効である。NoteCards〔HALA88〕はハイパー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、現状のハイパーメディアシステムの多くは、大規模で複雑なエンジニアリング業務に適用しようとすると、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、応用の進化に伴う変化の吸収、プログラミング機能等の点で問題がある。ここでは、ハイパーメディアの基本的機能のオブジェクト指向データベースによる実現方法と現状ハイパーメディアの問題点に対するオブジェクト指向データベースを用いた解決策〔ISHI90a〕について順に述べる。

オブジェクト指向データベースへのメディアの追加は容易に行うことができる。基本的なメディアとしてグラフィクス、イメージ、テキストが必要だが、それらはオブジェクトとして定義する。メソッドの長大データは、バイト列として格納する。Jasmineでは1ページに切れ目なく格納できるデータの最大長は、メディアクラス毎に定義することができる。メディアの操作はオブジェクトのメソッドとして定義する。

メディアの表示と構造・内容の間の対応関係は自明なものではない。つまり、メディアの表示とオブジェクトの構造や内容は関連するが、一応は別物と考えられる。例えば複合オブジェクトとしてのデータの表示には、属性名と属性値(様々な型がある)の組を表示する。そのためには複合したメディアの管理と表示が必要であり、それは部品関係を用いた複合オブジェクトとして管理される。

表示されたメディアをマウスで指示して、そのメディアや関連するメディアに直接アクセスすることが必要になる。この機能をメディアの直接操作といい、リンクをたどる検索・操作に対応する。この直接操作を行うためには、複合メディアの論理的構造の他に各メディアの位置情報(表示領域)からなる物理的構造も管理する必要がある。そのためにメディアオブジェクトのオブジェクト識別子とその表示領域の組をオブジェクトとして管理する。ウインドウ上でマウスクリックされた点を表示領域に含むオブジェクト識別子を前述のオブジェクトにメッセージを送って選択する。選択されたメディアオブジェクトを操作するには、それに対してメッセージを送ればよい。

オブジェクトは属性値そのものになるだけでなく、テキストの内容(例えば設計概説書)に含まれる場合もある。そこでテキスト内のオブジェクト(例えば関連する設計ゴール)を検索することも必要になる(図4参照)。これに対しては、上記のようにテキスト中のオブジェクトとその位置情報を管理する方式がある。我々はこの方式に対して、テキスト中のオブジェクトに対応する文字列とそのオブジェクトの識別子の対応表を、テキストと別に管理し、テキストの文字列をドラッグすると、対応表をもとに、対応するオブジェクトを検索する方法をとった。この方法は、オブジェクトの位置情報を管理する従来方式に比べ、テキストの変更に伴う位置情報の再計算をしなくてすむし、同一オブジェクトが複数あっても対応関係は1つですむという利点を持つ。

エンジニアリング業務は複雑で大規模であるためにハイパーメディアの直接操作(リンクアクセスを含む)だけでは不十分である。即ち複雑な条件や手続きを使った柔軟な検索が必要である。現状のハイパー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、筆者らによるJasmineを使った研究〔ISHI90a〕がある。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などの高度な検索に対しては、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5において(a)は仮想リンクの定義を、(b)はユーザインタフェースを通して仮想リンクを利用した例である。そこでは問い合わせもオブジェクトとして実現されている。アドホックな問い合わせは対話的に解釈し実行できる一方、定型的な問い合わせはコンパイルして効率的に実行できる。

エンジニアリング分野で日常的に起きる設計変更についても対処できる必要がある。即ち異なるメディ

アデータも互いに関連付けられているので、設計変更に伴う変更を正しく伝播する必要がある。我々は関連する異なるメディアデータ間の一貫性をデモンにより維持するようにしている。即ち設計データに対応する属性値（例えば頭部肉厚）を変更すると関連する図面データ（グラフィクス）も変更されるように、頭部肉厚属性に変更デモンを付加する。こうして設計変更の自動伝播を行うことができる。

さらに応用のためのユーザインタフェース管理機能を提供する必要がある。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としては、汎用のものを提供するだけでは十分ではない。ユーザが応用ごとにカスタマイズできる必要がある。我々は、システム定義のメディアとウインドウオブジェクトのライブラリを提供しており、システム定義のオブジェクトを、属性・メソッドの遺伝と新規追加によりカスタマイズすることによって、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容易に作成できる。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における研究〔DEUX90〕があるが、ライブラリはオブジェクトにより実現されてはいない。

6. おわりに

本論文で説明してきた研究は、CADやハイパーメディアといった複雑で高度な応用に対する新しいデータベースとしてオブジェクト指向データベースシステムが適していることを具体的に確認することを目指の1つとしている。そこで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 Jasmineを用いた具体的な試みについて制約の管理（作成，充足，維持，利用）を中心に述べてきた。ここで述べた課題は包括的ではないが、いづれも重要なものである。これらのアイデア（それを制約オブジェクト指向という）は、オブジェクト指向データベース Jasmineに100個以上のクラス（制約管理、三次元図形管理など）を追加することにより、知的設計支援のための試作システムHyperCAD〔ISH91a〕として実現されている（図6参照）。設計制約としては30個以上、幾何制約としては300個以上の制約を定義した。これによりオブジェクト指向データベースがエンジニアリング分野において、設計対象の複雑な構造と振る舞いのモデル化、高度な検索機能、機能拡張性などの点で適していることが確認できた。

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報告例〔KETA90〕〔KUWA91〕はきわめて少ない。そのため、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おこなうためにも、応用研究がシステムの研究開発とともに重要である。その意味でObaseプロジェクト〔NIKK90〕の発足は時宜を得たものであり、その動向は注目に値する。

今後の課題について述べる。制約に関しては現在数値的な制約を扱っているが、それらの他に今後はより定性的、原理的な制約の記述と充足も必要になる。即ち、設計制約についてはより概念的（言語的）な制約の扱いが重要となる。幾何制約では、よりハイレベルなやり方で制約を記述でき、それをもとに対象がどのように振る舞うかをおおまかに予測できる（シミュレーションできる）必要がある。メディアに関しては、現在グラフィックスをイメージに変換してアニメーションを行っているが、別手段としてビデオや光ディスクなどの動的メディアの利用が、システムの拡張性の確認のためにも重要である。さらにシミュレーションにしる動的メディアにしる、従来のデータベース応用とは異なるアクセスボタンを持った応用に相応しいデータの格納法やアクセス法の研究が重要である。その他、設計作業に伴うバージョンやトランザクション管理、複数のユーザによる協調的作業の支援等の応用に密着した課題が重要である。

参考文献

- [AOSH90] Aoshima, M. et al.: The C-based Database Programming Language Jasmine/C, Proc. The 16th International Conference on Very Large Data Bases, pp. 539-551 (1990).
- [ATKI89] Atkinson, M. et al.: The Object Oriented Database Manifesto, Proc. The First International Conference on Deductive and Object-Oriented Database, pp. 40-57 (1989).
- [DEUX90] Deux, O. et al. : The Story of O₂, IEEE Trans. Knowledge and Data Engineering, vol. 2, no.1, pp. 91-108 (1990).
- [HALA88] 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).
- [ISHI90a] Ishikawa, H. : An Object-Oriented Knowledge Base Approach to a Next Generation of Hypermedia System, Proc. IEEE COMPCON Spring 90, pp. 520-527 (1990).
- [ISHI90b] 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).
- [ISHI91a] Ishikawa, H. et al. : An Object-Oriented Database : System and Applications, IEEE Pacific Rim Conference on Comm., Computers, and Signal Processing, pp. 288-291 (1991).
- [ISHI91b] 石川, 泉田, 川戸 : 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, 情報処理学会, 情報処理, vol. 32, no. 5, pp. 593-601 (1991).
- [KETA90] Ketabchi, M.A. et al.: Comparative Analysis of RDBMS and OODBMS: A Case Study, Proc. IEEE COMPCON Spring 90, pp. 528-537 (1990).
- [KIM90] Kim, W., et al. : Architecture of the ORION Next-Generation Database, IEEE Trans. Knowledge and Data Engineering, vol. 2, no.1, pp. 109-124 (1990).
- [KUWA91] Kuwano, N. et al.: Applications of Object-Oriented Databases to Publishing Systems, Proc. The Second International Symposium on Database Systems for Advanced Applications, pp. 421-429 (1991).
- [LELE88] Leler, W.: Constraint programming languages. Addison-Wesley, Reading, M.A. (1983).
- [MAIE86] Maier, D., et al.: Development of an object-oriented DBMS, Proc. the First OOPSLA, pp. 472-482 (1986).
- [MORG83] Morgenstern, M.: Active databases as a paradigm for enhanced computing environments, Proc. the 9th VLDB conference, pp. 34-42 (1983).
- [NIKK90] 日経AI, 1990, 9月10日, p. 10 (1990).
- [WILK90] Wilkinson, K. et al.: The Iris Architecture and Implementation, IEEE Trans. Knowledge and Data Engineering, vol. 2, no.1, pp. 63-75 (1990).
- [YAMA90] 山田、宮城、野村、「AI機能を利用したサービス資料の原稿作成システムの開発」、自動車技術、vol. 44, 1990

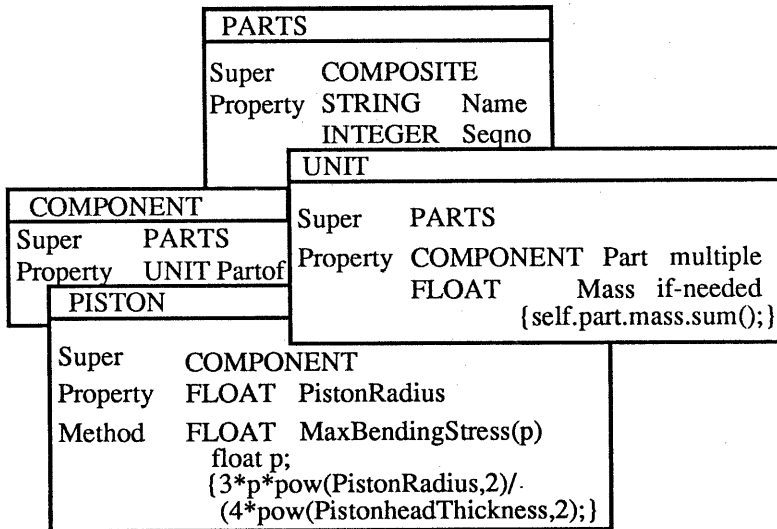
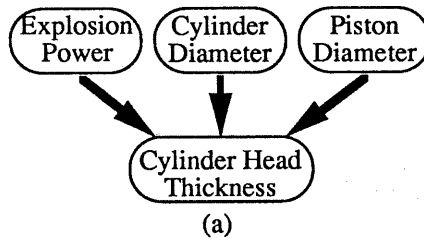
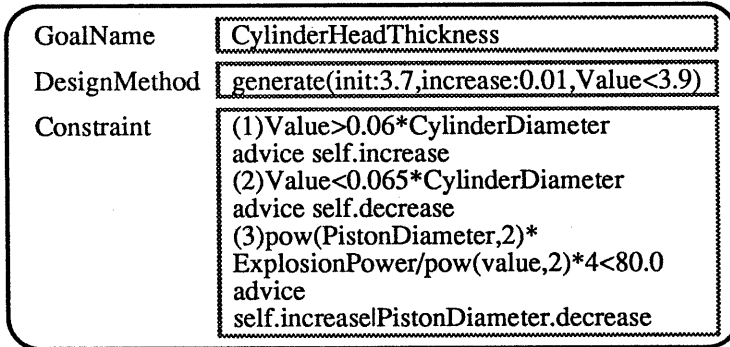


図 1. 部品クラスの階層の一部.



(a)



(b)

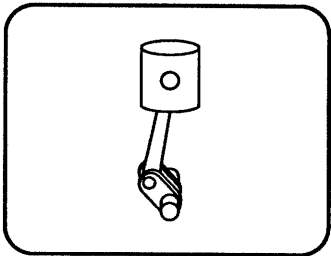
図 2. 設計ゴールの例 (a)設計ゴールの依存ネットワークの一部. (b)設計ゴールの定義の一部.

```

ConrodGeo.y= CrankGeo.CrankHead1.center.y
ConrodGeo.z=CrankGeo.CrankHead1.center.z+
CrankGeo.CrankArmThickness+
(CrankGeo.CrankPinLength-ConrodGeo.ConrodThickness)/2

```

(a)



(b)

図3. 制約に基づくアニメーションの例.
(a) 幾何的制約の一部. (b) アニメーション・スチール.

<p>EDITABLE_TEXT</p> <p>The pitch between the center cylinders at both ends is calculated as follows:</p> $l = x * r$ <p>where</p> <ul style="list-style-type: none"> l : the pitch x : magnification(3.6~4.2) r : crank radius 	<p>CRANK_RADIUS_GOAL</p> <table border="1"> <tr> <td>GoalName</td> <td>CrankRadius</td> </tr> <tr> <td>DesignMethod</td> <td>calculate(Stroke/2)</td> </tr> <tr> <td>Constraint</td> <td></td> </tr> </table>	GoalName	CrankRadius	DesignMethod	calculate(Stroke/2)	Constraint	
GoalName	CrankRadius						
DesignMethod	calculate(Stroke/2)						
Constraint							

図4. テキスト内に含まれるオブジェクトの検索例.

```

UNIT
method COMPONENT MajorComponent ( ) multiple
{self.Part where self.Part.Mass.sum() > self.Mass*0.1
groupby self.Part.Mass;}

```

(ユニットの主要部品を、その合計質量がユニットの10%を越える部品の集合として定義する。)

(a)

QUERY	
Target	UNIT.MajorComponent()
Condition	UNIT.Mass >40.0
<input type="button" value="run"/> <input type="button" value="cancel"/>	

(b)

図5. 仮想リンクの例. (a) 定義の一部. (b) 利用例.

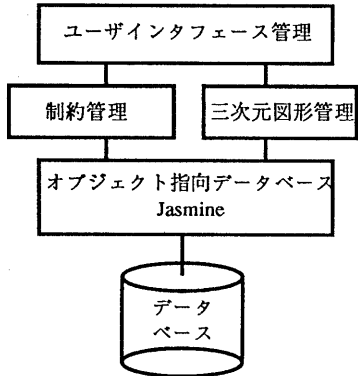


図6. HyperCADのシステム構成.