The University of Osaka
Institutional Knowledge Archive

Title	機能モジュール統合化による歯車伝達装置の設計支援システムに関する研究
Author(s)	藤田, 喜久雄; 赤木, 新介; 天木, 暢之
Citation	日本機械学会論文集 C編. 1994, 60(579), p. 3602-3609
Version Type	VoR
URL	https://hdl.handle.net/11094/2912
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

日本機械学会論文集 (C 編)
62 巻 579 号 (1994-11)

論文 No. 94-0339

機能モジュール統合化による歯車伝達装置の設計支援システムに関する研究*

藤田 喜久雄[†], 赤木 新介[†], 天木 暢之[‡]

Computer-Aided Design System for Gear Drive Devices with Functional Module Integratio

Kikuo FUJITA, Shinsuke AKAGI and Nobuyuki AMAKI

Design methods for gear drive devices are well described in textbooks and handbooks, since the type of such design problems are recognized as routine design. Though the design methodology for individual machine elements has been formalized, it is still difficult and time-consuming to arrange total systems, such as gear drive devices, which are composed of a set of machine elements. In this paper, we develop a computer-aided design system for the above design problems. In the system, design knowledge for the individual machine elements is modularized as “functional modules” in the manner of object orientation, and the operation and control knowledge for design are integrated with it based on the machine structure defined by the designer. The system can deal with several classes of gear drive devices and automate the various design calculation in such design processes. Finally, we show examples of its application to design problems of gear reducers.

Key Words : Machine Element Design, CAD, Design Engineering, Design Knowledge, Object Orientation, Functional Module, Gear Drive Devices

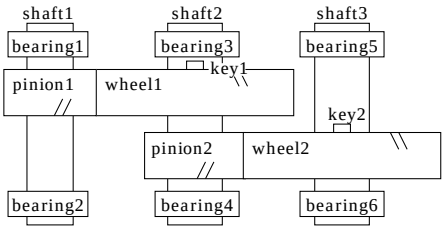
1 緒言

歯車減速機に代表される歯車伝達装置の設計過程においては，それを構成する歯車，軸，軸受，キーなどの機械要素についての対象知識，設計解を操作するための知識，さらに，設計過程を制御するための知識などの多種多様な知識が用いられる(1)．それらの内容のうち，特に個々の機械要素に関する対象知識については個別にその内容が整理され，設計過程についても定型的なルーチン設計であるものとされている．しかし，具体的なシステムとしての歯車伝達装置を設計するためには，上記の設計対象知識・設計操作知識・設計制御知識のそれぞれを全体として統合的に利用することが必要となる．

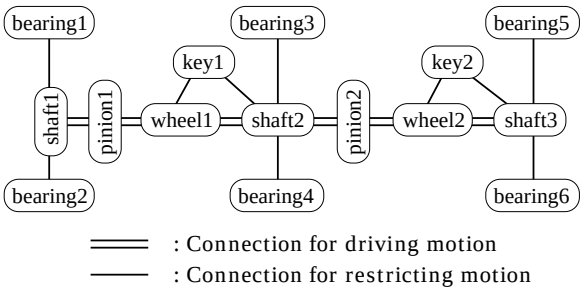
本研究では，歯車伝達装置を構成する様々な機械要素を具体的な設計対象と設計支援システムにおける機能的なモジュールとしてとらえ，設計者により定義される機械の構造(2)に従って，そのような装置を設計するために必要となる知識を自動的に統合化できるようにして，設計過程を有効に支援することのできるシステムをオブジェクト指向プログラミング(3)(4)の手法を基本として構築する．また，構築したシステムをいくつかの歯車減速機の設計問題に適用して，その有効性を検証する．

なお，定型的な設計問題に対する支援システムについては，設計処理の内容を設計実体についての諸属性の操作として把握できることから，オブジェクト指向を導入した試み(4)~(6)がなされており，各属性についての知識を各オブジェクトにモジュール化して記述することにより，有効に設計過程を支援することができる．本研究の試みは，個別の設計対象の構造にしたがって，そのようなオブジェクトを自動的に構成しようとするものであり，より適用性の高い支援システムとすることを目的としている．また，属性間の関係式を宣言的な記述から汎用的に操作しようとする立場から，制約プログラミングを導入した試み(7)もなされているが，本研究では，設計対象の領域を定めることにより，設計知識の内容を個々に具体的な手続きとして整理できることから，それらを直接的に利用することによって効率的な設計処理を実現するようにする．

*原稿受付 1994 年 3 月 9 日.
[†]正員，大阪大学工学部 (〒 **565** 吹田市山田丘 2-1).
[‡]学生員，大阪大学大学院工学研究科.



(a) 歯車伝達装置の一例



(b) (a) を表現するグラフ

図 1 歯車伝達装置の構造を表現するグラフ

2 機械の構造と設計知識

2.1 機械の構造とグラフ表現 緒言でも述べたように，歯車伝達装置は歯車，軸，軸受，キーなどの機械要素より構成されており，それらの間で運動や力を伝達しあったり，拘束しあったりすることにより必要な機能を具現化している．図 1はそのような状況を各機械要素とそれらの間の関係により示したものであり，図中 (a) に示す機械の構造が，“運動伝達エッジ”と “運動拘束エッジ” との 2 種類のエッジを用いることによって，各機械要素をノードとする (b) のようなネットワーク状のグラフとして表現されている[(2)]．このようなグラフによる表現は，複雑な 3 次元形状を有するであろう機械に対して，その構造や要素間のつながりを機能の面から簡潔に表現したものであり，図 1に示した機械の例に限らず，機械要素から構成される様々な機械システムを表現するための一般的な記法であると言える．

2.2 要素設計知識とシステム設計知識 一方，上記のようなグラフにより表現される歯車伝達装置の設計を行なうためには，個別のシステム要素の設計に加えて，システム全体としての整合性やバランスなどを調整するための設計を行なう必要があり，設計プロセスとしての特徴は，設計者の判断に基づいて「設計・評価・再設計」の過程を繰り返しながら調整を行なっていく点にある．

そのような過程のうち，前者の要素設計において用いられる知識内容は，機械要素毎の属性を決定するためのものであり，個々に設計計算式や設計規則の形で教科書やハンドブックなどにもまとめられているところである．したがって，個別の機械要素に対する設計処理については，それらの内容を適用することにより比較的容易に行なうことができる．しかし，実際の設計処理における内容は繁雑であり，有効な設計支援を行なうためには，設計計算そのものを具体的な設計対象に対して自動化していくことが求められる．

これに対して，後者のシステムとしての設計を行なうための知識は，個々の要素から対象システムの構造に対応した全体をまとめあげるために必要となる内容であり，個別の対象知識をそれぞれに適用した場合に生じる矛盾点などを調整したり，システムの構造に従った効率的な設計の手順を定めたりするための知識である．そのため，知識そのものの大枠については整理されてはいるものの，具体的な対象物に即して直接的に記述できるような内容ではなく，実際に設計過程において個々の設計事例に即した形で知識そのものを展開することが必要となる．

3 機械設計における機能モジュールと統合化

3.1 機能モジュール 上述のように，歯車伝達装置の設計問題では，システムの位相的な構造に基づいて個々の知識を統合的に組織化し，総合的に設計過程を支援することが必要であると考えられる．一方，設計対象に対する知識をはじめとする様々な知識内容は，前出の図 1に示したような機械構造に従って，個々の機械要素やそれらの間の接続関係に対して直接的に対応していたり，それら相互の構成的な関係をもとに説明付けられたりするものとして理解することができる．したがって，そのような内容を “機能的なモジュール” としてとらえ，個々の設計対象に対応させて知識内容を統合化することにより設計支援システムを構成することができれば，機械設計のための汎用的なツールとすることが期待できる．

本研究では，以上のような設計支援システムを実現する方法として，グラフとして表現できる機械の構造をもとに，オブジェクト指向によるプログラミング手法[(3)(4)]を導入する．すなわち，具体的な対象システムに依存しない知識内容については，設計支援システムにおけるライブラリとしてとらえ，個々の機械要素の種類に応じたクラスオブジェクトに定義するようにし，一方，図 1のようなグラフ表現におけるノードとエッジに対応するところの，具体的な設計対象に依存した

個々の設計対象実体やそれらの間の関係については，そのようなクラスのインスタンスオブジェクトとして定義するようにする．これにより，ライブラリ中の内容と具体的な設計対象の構造から実際的な設計知識の内容を導出できるメカニズムを実現する．つまり，オブジェクト指向における継承の性質により，クラスに記述された知識内容に基づいて，個別のインスタンスオブジェクトにおける処理を行なえるようにして，上述の汎用性を実現するようにする．

3.2 歯車伝達装置の設計知識 本項では，上述のような機能モジュールの統合化による設計支援システムの構成や知識処理の方法について述べる準備として，歯車伝達装置の設計において用いられる知識の内容について，長澤による設計知識の分類方法[(1)]をもとに整理しておく．

設計対象知識 … 個々の機械要素は寸法や材質などの様々な属性を有しており，それらを決定したり，評価したりするための知識を設計対象知識としてとらえることができる．そのような内容は，個別の要素に内在するものとしてとらえることができ，各機械要素などの種類に応じて共通化することができる．

設計操作知識 … 設計を進めていく上で，設計対象における問題点を発見したり，それを解消するための修正を行なうための知識であり，具体的な内容については，機械の構造に従って，評価すべき項目とそれに対応した設計修正案の候補をあらかじめ列挙しておくことができる．

設計制御知識 … 設計を効率的に行なえるように，その順序などを定める知識であり，例えば，歯車伝達装置の設計においては，機械要素の種類や動力伝達における経路や順序に従って，円滑な設計が行なえる設計手順を定めることができる．

4 歯車伝達装置の設計支援システム

4.1 システム構成 上述のような機能モジュールの考え方と設計知識の内容に基づいた設計支援システムの具体的な構成を図2に示す．システムは，機械構造の入力部，初期化部，設計プロセス管理部，機能モジュールライブラリ，データベース部から構成されている．このような構成のシステムに対して，まず，ユーザである設計者が入力部の機能を用いて機械の構造を定義し，それに従って，初期化部が，機能モジュールライブラリ中の各クラスオブジェクトをもとにして必要な対象要素をワーキングステージ上のインスタンスオブジェクトとして定義し，また，設計プロセス管理部上に機械の構造に対応した設計手順などを定義する．それによって，ユーザは個々の設計問題における様々な設計処理を対話的に実行しつつ，設計が行なえるようになる．一方，データベース部には，標準的な設計の例や軸受などのカタログデータが保持されており，必要に応じてそれらを検索して具体的な設計を選定できるようになっている．

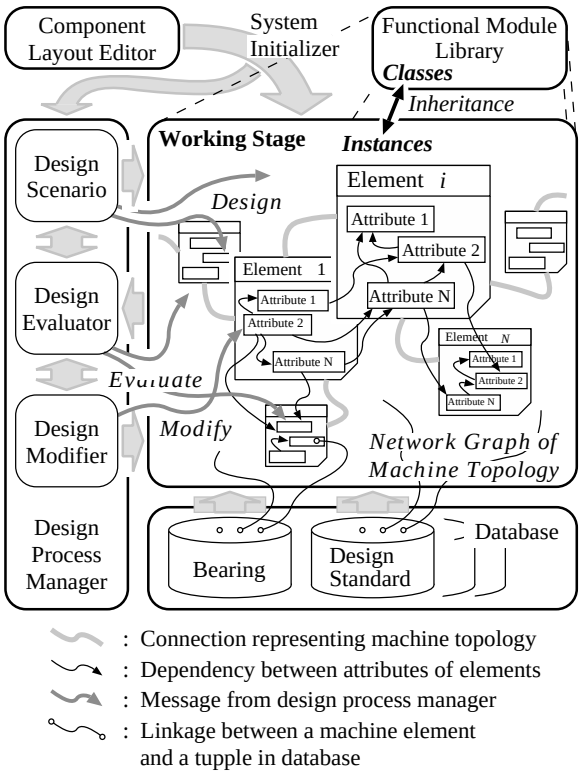


図2 設計支援システムの構成

なお，本システムはUNIXワークステーション上でSmalltalk-80[(8)~(10)] Release 4.1を用いて構築しており，知識表現や設計処理をオブジェクト指向プログラミングにより実現することに加えて，各種のユーザーインターフェース機能についても，マルチウインドウ上で視覚的に構成している．

4.2 クラスライブラリ 図3は，本システムで用いるクラスとその階層関係を示したものである．これらは個別の設計対象における設計処理に対するライブラリとして動作するものであり，主要なものとその役割は以下のとおりである[1]．

`ComponentLayout,ComponentLayoutView,`

[1] 以下では，Smalltalk-80の記法に従って，大文字で始まる文字列は，クラスオブジェクトあるいはクラス変数を，小文字で始まる文字列はインスタンスオブジェクトあるいはインスタンス変数を指すものとする．一方，単なるシンボルは#+で始まる文字列として表して，それらとは区別する．

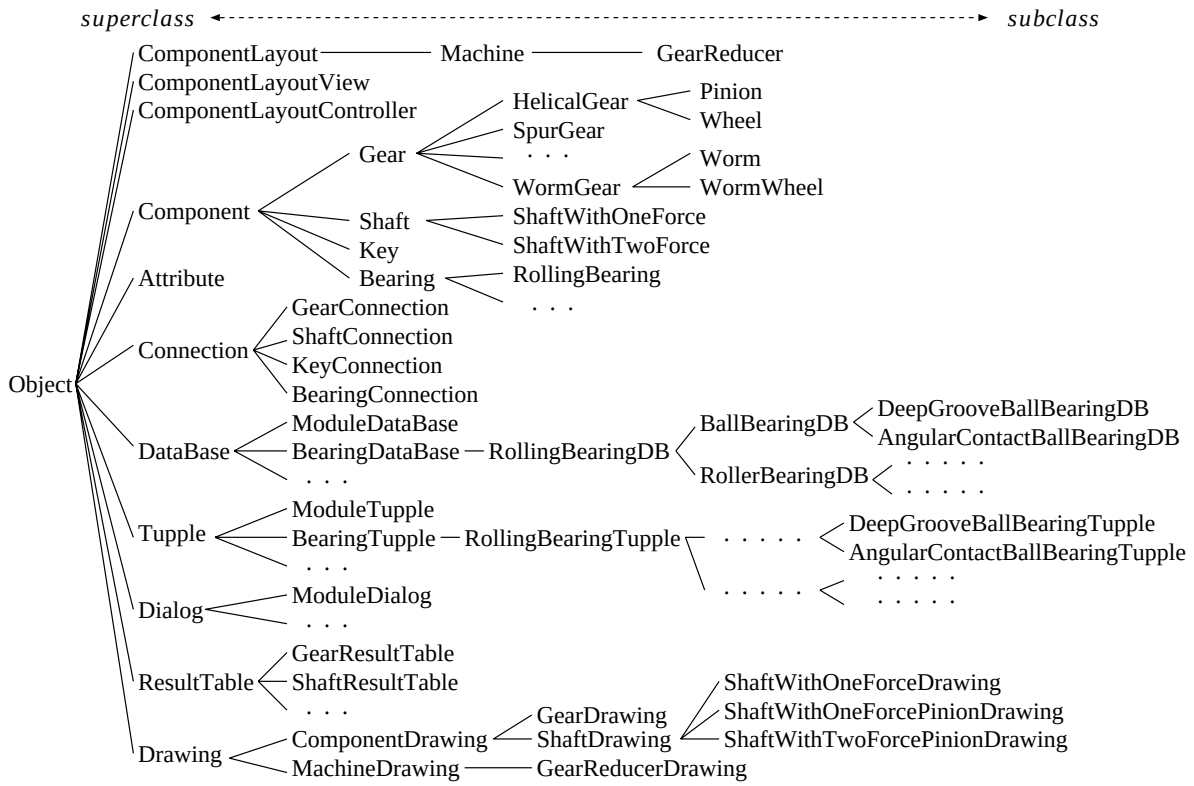


図 3　クラスライブラリの構成

`ComponentLayoutController` … 設計対象の機械の構造を入力したり，管理したりするためのオブジェクトであり，その構成は Smalltalk-80 における対話型オブジェクトの構成方式である MVC (Model-View-Controller) モデル[(10)~(12)]に準じたものとなっている．

`Machine` … 上記の `ComponentLayout` のサブクラスであり，定義された機械の構造に従って，設計プロセスを管理するオブジェクト．

`Component` … 機械を構成する歯車，軸，軸受，キーなどの要素を表現するためのオブジェクト．

`Attribute` … 機械要素の属性，例えば，歯車の歯数やモジュールなど，に関する情報である属性値やその決定方法などを保持・管理するためのオブジェクト．

`Connection` … ある機械要素からみて，運動伝達や運動拘束などの関係により接続されている要素を保持・参照するためのオブジェクト．

`DataBase` … 軸受などに対するカタログデータやモジュールなどの標準値などを保持したデータベースのオブジェクト．

`Tupple` … 上記のデータベースに含まれる個別の要素を表現するオブジェクト．

`Dialog` … 設計者にデータ入力や選択などの処理を促すなどのユーザーインターフェース処理を行なうためのオブジェクト．

`ResultTable`, `Drawing` … 各機械要素の要目表や部品図を表示するためのオブジェクト．

以下の項では，個々のオブジェクトにおける表現や処理の方法について述べる．

4.3　機械の構造を表現する方法　まず，上述のオブジェクトのうち，`Component` と `Connection` を用いて表現される設計すべき機械の構造を表現する方法について示す．

図 4は歯車が噛み合うことによる接続関係を示している状態の例であり，`aComponent` という要素オブジェクトの `connectedComponents` というインスタンス変数のスロットには接続関係を表現したオブジェクトである `aConnection` が保持されており，その `drivenComponent` というスロットには駆動側の歯車の名称がシンボルとして保持されている．これに対して，その駆動側の歯車が保持している `aConnection` の `drivingComponent` という スロットには被動側の歯車の名称が保持されており，相互に参照することができるようになっている．一方，同図下部の `anAttribute` と `aVariable` は，それ

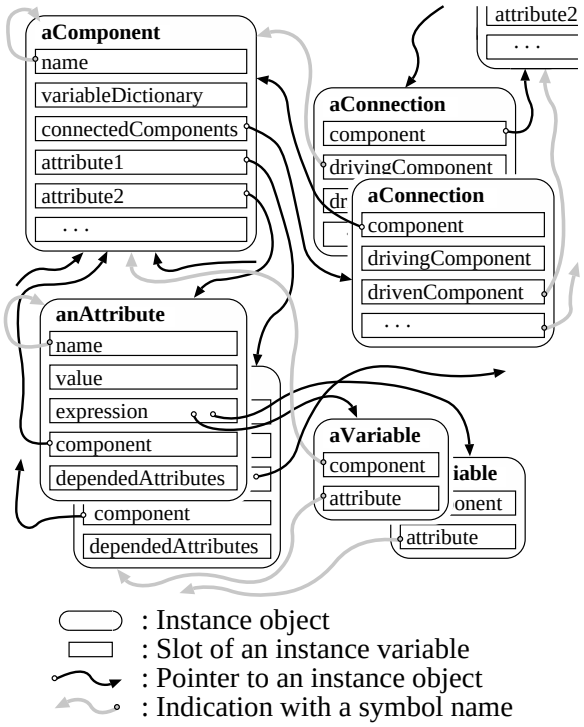


図 4　機械構造のオブジェクト表現

```
(#/ (#* (Variable component: #drivingGear
                  attribute: #rpm)
        (Variable component: #drivingGear
                  attribute: #numberOfTeeth) )
    (Variable component: #myself
              attribute: #numberOfTeeth) )
```

(a) 被動歯車の回転数の計算式

```
(#search: RollingBearingDB
   (Variable component: #myself
             attribute: #innerDiameter)
   (Variable component: #myself
             attribute: #radicalLoad)
   (Variable component: #myself
             attribute: #axialLoad)
   (Variable component: #myself
             attribute: #rpm) )
```

(b) ころがり軸受の検索処理

```
(#get: FormFactor
   (Variable component: #myself
             attribute: #numberOfTeeth) )
```

(c) 歯形係数値の参照処理

```
(#query: ModuleDialog
   (Variable component: #myself
      attribute: #moduleBasedOnBendingStrength)
   (Variable component: #myself
      attribute: #moduleBasedOnPittingStrength) )
```

(d) モジュール値の問い合わせ処理

図 5　設計対象知識の表現

ぞれ，aComponent というオブジェクトの属性を扱うためのオブジェクト，属性の決定方法の記述中に現れる属性指定子の内容を表すオブジェクトであり，図中にも示すように，それらについても相互に関連付けられている．

なお，このような各オブジェクトやそれらの間の関係は，ComponentLayout などのオブジェクトを介して，設計者がマウス操作によりビットマップディスプレイ上に描いたシステム構造の概略図をもとに，自動的に生成できるようにしている．

4.4　設計制御知識の表現と処理　設計を行なう手順は，上記のようにして表現される機械の構造から，以下のような基準に基づいて決定して，それにより効率的な設計が行なえるようにする．

- 機械要素の種類に従って「歯車 ⇒ 軸 ⇒ キー ⇒ 軸受 ⇒ ···」の順に設計を進めていく．
- 同種類の機械要素については，動力が伝達されていく経路の順序に従って，設計を進めていく．

このような内容については，図 4に示した aComponent や aConnection などのオブジェクトのスロットに保持されている各機械要素の種類を参照したり，要素間の接続関係を入力軸の側からたどっていき伝達経路を把握した上で，整列処理を行なうことによって，決定することができる．また，そのようにして決定される設計順序は，設計プロセスを管理するオブジェクトである aMachine に，機械要素の種類については，

```
( #Gear #Shaft #Key #Bearing )
```

という形式のリスト[2]で表現し，さらに，ある種類の要素ごとの順序についても，例えば #Gear については，

```
( pinion1 wheel1 pinion2 wheel2 )
```

というリストで表現しておくようにする．このような表現に基づいて，後述の設計対象知識や設計操作知識の適用順序が統括的に管理されるようになる．

4.5　設計対象知識の表現　次に，設計対象である各機械要素の属性を決定するための知識の表現方法について示す．属性の決定方法は anAttribute オブジェクトの expression スロットにリスト表現を用いて表現するようにする．図 5は，その具体的な記述例を示したものであり，以下に個々の内容について説明する．

[2] 以下で用いるリスト表現の形式は Smalltalk-80 の文法にしたがうものではなく，Lisp 言語流の括弧対による表現により，形式的にその内容を示したものである．

(1) 設計計算式による計算　様々な属性の多くは，四則演算を中心とした算術計算式により決定される．例えば，被動歯車の回転数は，$n_2 = n_1 \cdot z_1/z_2$ なる計算式により決定できるが，そのような内容は図 5 (a) のような形式で被動歯車オブジェクトの回転数属性オブジェクトに保持されている．このうち，‘#/’ と ‘#*’ は四則演算子を，(Variable ⋯) なる形式で記述されている他の部分は，それぞれの演算で用いられるべき他の属性項目を指し示している．

(2) データベースの検索・参照処理　機械要素の設計では，カタログデータを検索したり，標準的な寸法値を参照したりする処理が不可欠であり，そのような内容については図 5 (b) や (c) のような形式で保持されている．前者は，ころがり軸受のデータベースである ‘RollingBearingDB’ を，内径，ラジアル荷重，アキシアル荷重，回転数の値をもとに検索して，具体的な軸受を選定する処理を表現しており，後者は，歯形係数の値を決定するために自身の歯数の値をもとに ‘FormFactor’ というデータベースを参照する処理を表現している．

なお，具体的な検索処理の内容やデータベースの構成については，4.9項で後述する．

(3) 設計者との対話処理　以上の処理のほか，機械要素の設計においては，様々な設計計算により求められる参考値をもとに，設計者自身が具体的な属性値を決定する場合もある．図 5 (d) は，曲げ強さから計算されたモジュール値と面圧強さから計算されたモジュール値をもとに，‘ModuleDialog’ というインターフェース用のオブジェクトを介して，実際のモジュール値を設計者に入力させるための記述である．

なお，このようなインタラクティブな処理については，Smalltalk-80 の機能を用いて，ウインドウを介した視覚的な入力処理が行なえるようにしている．

4.6　設計対象知識による処理　以上の各対象知識の内容は，それぞれ，クラスライブラリ中のクラスオブジェクトにより，具体的な設計対象の要素に対応したインスタンスオブジェクトを生成する際に，付随した属性オブジェクトのスロットに記述される内容である．その内容はあくまでも個別の機械の構造に特化したものではなく，具体的に実際の設計処理を行なう時点において，機械の構造に従った解釈と処理が行なわれる．

例えば，ある要素のある属性の値を決定する処理は，以下のようにして行なわれる．すなわち，ある機械要素のオブジェクトに対してある属性の値を参照するメッセージが送信されると，まず，属性オブジェクトの value スロットを調べ，既に値が決定されていればその値を直ちに返し，未決定であれば expression スロットの記述をもとにその値を決定する．その際には，expression スロットの内容を解釈して，参照すべき属性のオブジェクトを aConnection オブジェクトなどによる表現を通じて特定し，それらに対して同様のメッセージを再帰的に送信することにより，計算に必要となる引数の値を求める一方，記述内容を Smalltalk-80 において実行可能なメッセージ式に展開することにより評価して，自身の値を決定し，それを value スロットに格納するようにする．

なお，以上のような再帰的なメッセージ送信による属性の決定処理過程で発生する属性間の依存関係については，4.8項で述べる再設計処理において参照できるように，各 anAttribute インスタンスの dependedAttributes スロットにその値を用いて決定された属性のオブジェクトを保持させておく．

4.7　設計操作知識の表現と処理　以上のような設計知識に基づいて様々な設計項目の決定が行なわれると，そのような内容が適切であるかどうかを評価し，必要に応じて設計修正を行なう必要がある．具体的な評価項目の内容は，大きく，

- 「実際に決定されたピッチ円直径がその算出に用いられた仮定値と異なる」などの特定の機械要素における属性間の関係における不具合．
- 「軸の直径が歯車の歯底円直径よりも大きい」などの接続関係を直接的に介した要素間にわたる問題点．

の二つに分類することができる．このような内容は，前者については，その処理内容が個別の機械要素の内容に閉じているため，そのような aComponent オブジェクトのスロット情報として，後者については，設計プロセスを管理する aMachine オブジェクトのスロット情報として，それぞれ，評価項目と修正案の方針とを対応させて保持しておくようにする．

図 6はその具体的な記述の例であり，評価すべき項目については，図中 (a) のように，設計対象知識の記述方法と同様の形式で表現し，また，そのような不具合が生じた場合の対応方法と修正すべき属性については，それらの対を辞書を用いて図中 (b) のような形式で記述した上で，複数個の代替案を優先度に従って配列中に保持させておく．このような知識の記述に基づいて，個々の要素の設計が完了した時点で，適用可能な評価項目が検索され，問題点があれば，設計者に対して，どの修正案を採用するかを問い合わせた上で，

```
(#> (Variable component: #shaft
             attribute: #diameter)
    (Variable component: #revolvingComponent
             attribute: #rootCircleDiameter) )
```

(a) 評価すべき項目の記述

```
(Dictionary new
   at:  #messageForDesigner
   put: ’;u?t$rA}$d$9$3$H$G!";u<V7B$rBg$-$/$9$k’;
   at:  #resetAttribute
   put: (Variable component: #revolvingComponent
                  attribute: #numberOfTeeth) )
```

(b) 修正代替案の記述

図 6　設計操作知識の表現

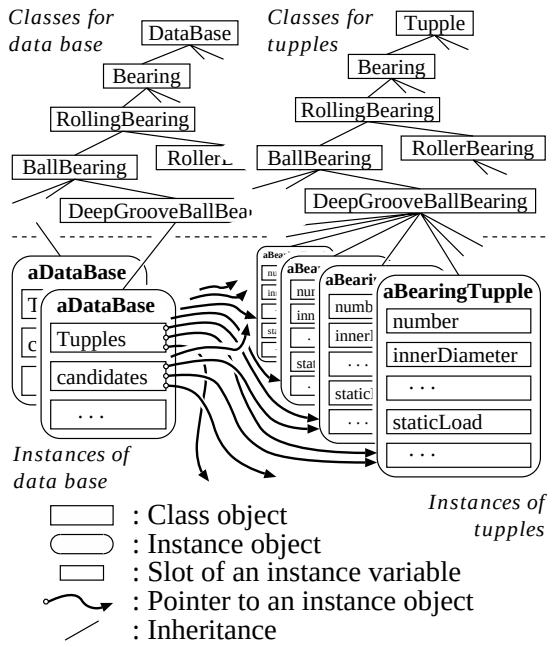


図 7　データベースの構成とその検索・参照処理

選択された項目に対して，次項で述べるような再設計処理を行なうようにする．

4.8　再設計処理　前項のようにして，ある設計項目が変更された場合，それに関連した他の設計項目を連動させて変更する必要がある．そのような依存関係は，4.6項で示したように各 `anAttribute` インスタンスの `dependedAttributes` というスロットに保持されているため，そのような情報を用いて，関連のある属性項目の値を再帰的に消去していくようにする．その上で，再び設計処理を行なうことによって，再設計処理が行なえるようになる．

4.9　データベースの構成とその検索・参照　機械要素の設計では，軸受などの部品や歯形係数などの標準値をデータベースから検索する必要がある．図7は，軸受のデータベースの例をもとに，本システムにおけるデータベースの構成と検索方法を示したものであり，前出図 3に示したクラス階層に従って，図中左側にはデータベースを操作するためのオブジェクトが，右側にはそれらと対応する形で個別のデータを表現したオブジェクトが示されている．ある特定のデータベースのオブジェクトは，対応する検索対象データの集合を `Tupples` というスロットに保持しており，設計条件から与えられる検索条件に従ってそれらを絞り込んで，`candidates` に保持し，さらに，設計者がその中から適切なものを選択することにより，軸受などの選定ができる．

なお，以上の過程においては，個別の軸受毎に検索条件に対応した寿命計算などを行なう必要があり，そのような計算の内容は軸受の種類に従って異なってくるが，それについても，オブジェクト指向における差分プログラミングの機能[(10)(12)]により，各軸受クラスオブジェクトにメソッドとして計算式を記述するようにして，柔軟に対応することが可能である．

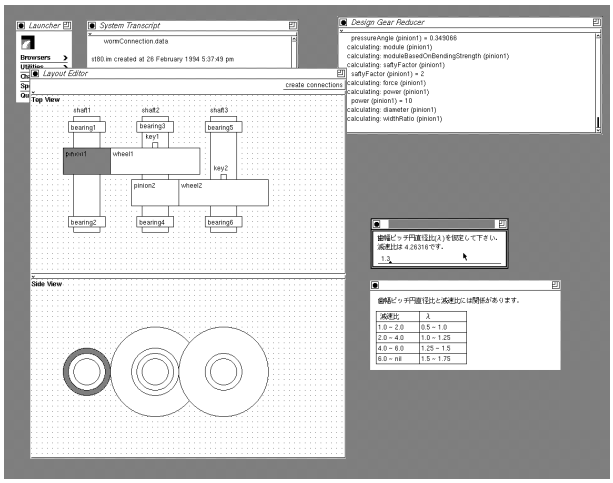
5　適用例

最後に，本システムを用いて 2 段はすば歯車減速機の設計を行なった事例を示す．まず，設計者は図 8 (a) の①に示されているウインドウを用いてマウス操作により設計すべき対象の構造を定義する．それに続いて，②部に示すような形式で，設計処理を行なう上で必要となる内容についての問い合わせや，関連する事柄がウインドウ上に表示され，設計者がマウス操作などにより対応しながら，設計を進めていく．さらに，各機械要素の諸元が図 8 (b) の③部のように決定され，最終的に減速機の設計が完了すると，その結果が④部などのように寸法の入った部品図などとして表示される．

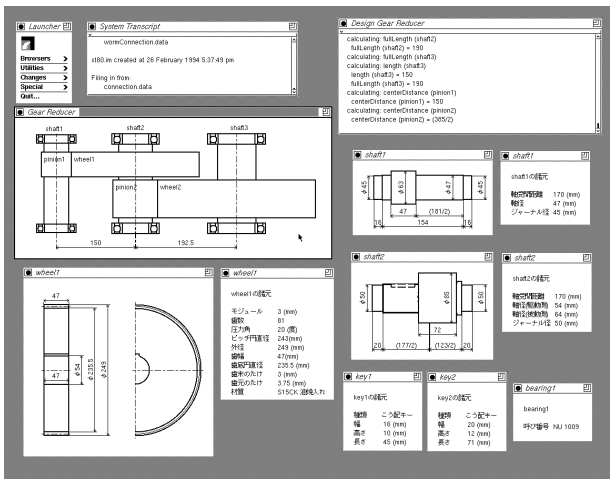
このほか，本システムは，図 9にも示すようにウォーム歯車を用いた場合や段数の異なる場合などにも対応でき，ユーザが定義する機械の構造に従って様々な歯車伝達装置の設計過程を支援することができる．

6　結 言

本研究では，歯車伝達装置の設計問題に対して，その構造がネットワーク状のグラフを用いることにより表現できることに着目した上で，装置を構成する個々の機械要素を設計支援システムを構成する上での機能モジュールに対応させることにより，様々な機械の構造を有する装置の設計過程を幅広くインタラクティブに支援することのできるシステムを構築した．本シス



(a) 設計計算の実行



(b) 設計結果の表示

図 8　設計支援の一例 – 2 段はすば歯車減速機の設計

テムは，オブジェクト指向の概念に従って，そのような機能モジュールに関する知識をクラスオブジェクトに記述しておいた上で，具体的な設計対象を個々の構成要素に対応したインスタンスオブジェクトの集合として表現するようにし，様々な設計処理をクラスから継承することによって実現したものである．このようなシステムの構成方法は，複雑なシステムを設計するような場合においても，個別の要素に関するよく整理された設計知識を統合化していく上で，有効な設計支援システムの構成方法であると考えられる．

文　献

(1) 長澤，設計エキスパートシステム，情報処理，**28**-2, (1987), 187.

(2) 吉川, 機械のトポロジ, 精密機械, **38**-12, (1972), 30.

(3) 例えば，Booch，G.，*Object Oriented Design with Applications*，(1991)，The Benjamin / Cummings Publishing.

(4) 赤木・藤田, オブジェクト指向に基づく設計エキスパートシステムの研究, 機論, **54**-500, C (1988), 1017.

(5) 藤田・赤木, 機能設計における設計対象のモデリングと形状モデルとの融合, 機論, **57**-535, C (1991), 1058.

(6) 中島・馬場・加藤, オブジェクト指向設計知識・データ表現システム, 情報処理学会論文誌, **32**-5, (1991), 635.

(7) 車・横山・岡部, 機械系 CAD のための設計変数・制約ネットワークにおける制約充足の一方式, 機論, **59**-568, C (1993), 394.

(8) Goldberg, A. and Robson, D., *Smalltalk-80: The Language and Its Implementation*, (1984), Addison-Wesley.

(9) Goldberg, A. and Robson, D., *Smalltalk-80: The Interactive Programming Environment*, (1984), Addison-Wesley.

(10) 上谷 (編著), 統合化プログラミング環境 — Smalltalk-80 と Interlisp-D —, (1987), 丸善, 185.

(11) 赤木・鷹津, オブジェクト指向言語による配管設計支援システムの研究, 機論, **54**-499, C (1988), 787.

(12) 赤木・鷹津, オブジェクト指向言語による配管設計支援システムの研究 (第 2 報), 機論, **55**-520, C (1989), 3087.

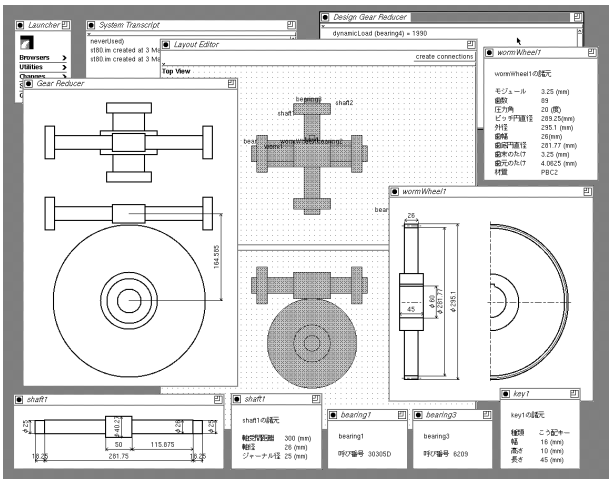


図 9　設計支援の一例 — ウォーム歯車減速機の設計