The University of Osaka
Institutional Knowledge Archive

Title	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
Author(s)	赤木, 新介; 藤田, 喜久雄
Citation	日本機械学会論文集 C編. 1988, 54(500), p. 1017-1025
Version Type	VoR
URL	https://hdl.handle.net/11094/3250
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)
54巻500号(昭63-4)

論文 No. 87-0234 A

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*

赤木新介*1, 藤田喜久雄*2

Developing an Expert System for Engineering Design Based on the Object-oriented Knowledge Representation Concept

Shinsuke AKAGI and Kikuo FUJITA

An expert system is developed for engineering design based on the object-oriented knowledge representation concept. The design process is understood as determining the design parameters and their relationships which consist the design model. The design model is represented as a network in the computer system using the object-oriented knowledge representation. The system built with the above concept provides the following abilities: 1) flexible model building and easy modification, 2) effective diagnosis of the design process by using the rule-based knowledge representation, and 3) a hybrid function with numerical computations and graphics as well as AI Techniques by coupling the systems programmed with LISP and FORTRAN languages. The system is suitable for the wide domain of applications with the above various functions. Finally, the system's validity and effectiveness is ascertained by applying it to the basic design of a ship.

Key Word: Design Engineering, Expert system, Object-oriented System, CAD, Ship Design

1. 緒言

設計におけるコンピュータによる支援は，いわゆるCAD／CAM／CAEなどの分野として次第に確立されつつあるが，設計における情報処理の内容は多様であり，明確なアルゴリズムを持たない問題，いわゆるIll-structured Problemである部分が多い[1][2]．特に多様な設計要求を満たす設計解を求める機能設計では，試行錯誤的に要目や寸法，形状が定められるが，この過程におけるアルゴリズムは明確でなく，設計者は設計解を仮定しながら，それに修正を加えつつ評価を繰返し，満足解に近づける．このような処理は，設計の各部分に応じて自律分散的に行われる．しかし，設計対象が大型で複雑なシステムになると，このような処理を設計者の能力のみで行うことは容易ではなく，なんらかの支援が必要となる．このような手法としてAI(Artificial Intelligence)技術の応用であるエキスパートシステム[3]に期待が持たれる．エキスパートシステムは，工学分野でも広く応用されつつあるが，設計への適用は一般に難しいとされる[4][5]．これは，知識表現が一般に困難なためであり，設計自体の本質が十分理解されていないことにも起因する[6][7]．

本研究は，知識表現の手法としてオブジェクト指向の概念[8][9]を導入し，設計者を支援するための汎用設計エキスパートシステムを開発することを目的とする．1）設計知識をネットワーク表現に基づいたオブジェクトとして表すことにより，モジュール性を向上し，自律分散的な処理を可能として，設計プロセスを柔軟に支援する．2）ルール型知識表現を用いて設計結果に対する評価診断を行い，設計過程中において有効な設計指針を与える．3）FORTRANなどによる手続き型プログラムとの連係を可能とする．4）システムの構成を「汎用的な設計知識処理の部分」と「設計対象に固有な知識ベースの部分」に分離し，種々の設計対象への適用が可能なシステムとする．

なお，上記のオブジェクト指向の概念には，現在いくつかの立場がある[9]．言語自体の構成をデータ形オブジェクトとして表現したSmalltalk-80[10][11]，LISP言語において知識表現形式にオブジェクト指向を

* 昭和62年11月28日 関西支部第247回講演会において講演，原稿受付 昭和62年2月25日．
*1 正員，大阪大学工学部（〒565 吹田市山田丘2-1）．
*2 学生員，大阪大学大学院．

用いたFlavor[12]やLOOPS[13]，C言語におけるObjective-C[9]，さらにPROLOG系のESP[8]などが開発されている．本システムは，知識表現形式としての立場をとり，その機能として多重継承やFORTRANとの連係などに重点を置くもので，Flavorなどに近いが，LISP専用機ではなく汎用コンピュータ上に実現した点に特徴がある．

事例として，開発したシステムを船の基本設計へ適用し，本システムの有効性を検証する．なお，MacCallumは船の基本設計に関する処理をネットワークとして理解する研究[14]を行っているが，具体的な処理内容については明らかにしていない．

2．システムにおける設計モデル

本システムの基本となる設計処理のモデル化を行う．

2・1　設計プロセスのモデル化　　設計プロセスは，「①扱うべき設計モデルの作成」「②対象のモデルによるシミュレーション」「③シミュレーションによる結果の評価」「④評価結果に基づくモデルの修正」を繰返すプロセスである．[2] また，モデルによるシミュレーションは，例えば事例として取り上げる船の基本設計では後出の図１１のように，いくつかの段階に分かれ，各段階ごとに結果を評価し，必要ならば修正を行い，時には再び最初からシミュレーションをやり直しながら次の段階へと進む．すべてのシミュレーションが完了し結果が満足であれば，設計は終了する．この過程を図１のような設計・表示・評価・判断・修正の各フェーズの遷移としてとらえる．

2・2　設計におけるネットワークモデル　　上記のような設計処理の各段階では，設計条件を設定したり，要目や性能値を決定するための各処理が行われる．例えば，船の基本設計の場合では図５のように，ある設計パラメータを決定するために付随する他の設計パラメータの値を用いるという関係が再帰的に連鎖し，設計パラメータ間でネットワークが形成される．これを一つのネットワークモデルとしてとらえる．このつながりは複雑で設計状態に対応して動的に形成されるため，ネットワーク状につながるパラメータ間の関係を設計者の能力のみで処理することは困難である．そこでこのような問題を解決するために，オブジェクト指向の概念を導入する．詳細は後述(４章)するが，このような関係に対して，各設計パラメータをオブジェクト，設計処理をメソッド，パラメータ間の処理の連鎖をオブジェクト間のメッセージ・パッシングとしてとらえることにより，オブジェクト指向に基づいたモデル化を行う(図２)．さらに，このような設計プロセス中における設計結果の評価は，設計条件と設計結果の比較や法規の適用などを行う処理であり，これらをルール型知識表現により支援するとともに，システムの汎用化をはかる．

3．システムの構成と環境

本システムの構成を図３に示す．システムは①汎用設計知識処理システムと②設計対象固有の知識ベースに分れる．このようにシステムを分離することにより，一般のエキスパートシステム開発ツールと同様の機能を持ち，さらに設計問題固有の知識処理に対して特化することによって，設計に対する汎用ツールとすることができる．また，知識の追加や修正も容易となる．

3・1　固有知識ベースの構成　　知識ベースとして図４の構成を考える．これは，前述の設計知識のオブジェクトの他に，設計プロセスの進行を管理するためのメタ知識を表現した述語，検索対象のデータを表現したオブジェクト，設計結果の診断評価を行うためのルールなどからなるＡＩ知識ベースと，数値計算や図形表示を行うFORTRANプログラムや記号処理を伴う

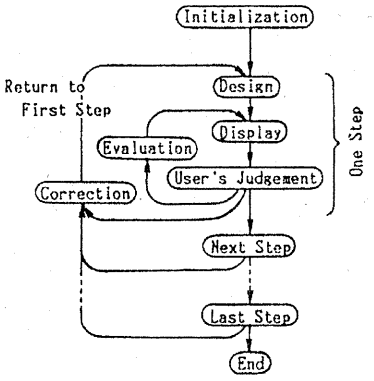


図１　設計プロセスのモデル化

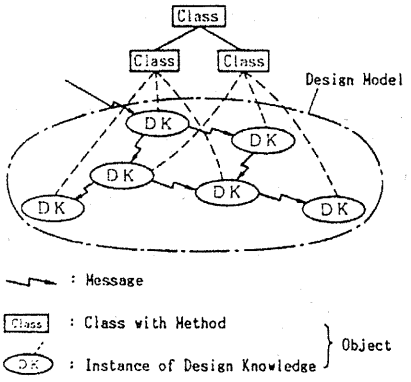


図２　設計知識のモデル化

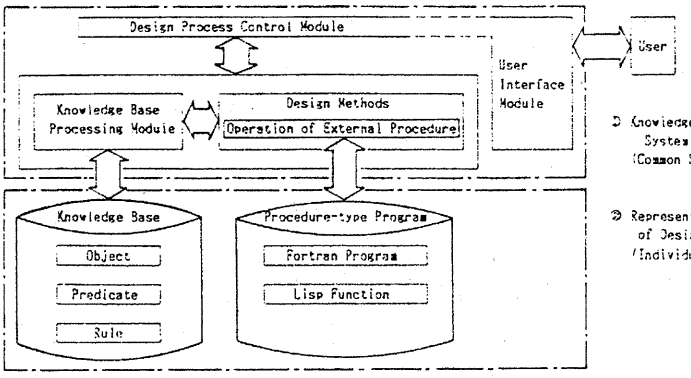


図3　システムの構成

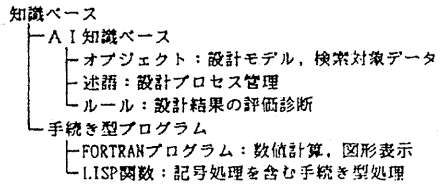


図4　知識ベースの構成

手続き的処理を行うLISP関数からなる手続き型プログラムから構成される．以上のような多様な知識表現により，設計対象に固有な種々の設計処理を行うための知識に対応することができる．

3・2　汎用設計知識処理システムの構成　　汎用設計知識処理システムは，知識ベースを操作し，設計者とインタフェースを取りつつ設計を行う部分である．この部分は，設計プロセスの進行管理とユーザインタフェースを行う部分，知識ベースの処理を行う部分，設計プロセスにおける基本処理を行うメソッドを定義する部分から構成される．手続き型プログラムの呼出しはメソッドの機能により行われる．

4．知識表現とその処理

4・1　設計知識のオブジェクト表現とその処理

設計モデルは，図2のように設計知識を細分化した個々の設計知識(パラメータ)のネットワークである．これは，システムが用意したクラスのインスタンスとして作成されたオブジェクトの集合として表現され，操作が行われる．ネットワークは具体的には，図5のような形となる．これは，船の基本設計における例(簡略化してある)で，主要寸法の決定から機能の推定の部分に至るものである．このようなネットワークに対して前述のオブジェクト表現を導入することにより，計算のアルゴリズムが各オブジェクト間のつながりと

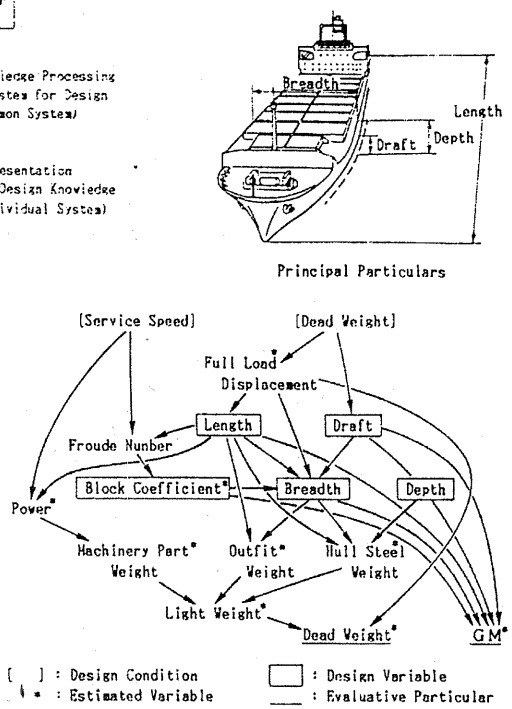


図5　設計におけるネットワークの一例

して動的に決定されるようになる．すなわち，設計パラメータ間の依存関係によって，あるパラメータの値が他のパラメータの値に依存して決定されるために，あるパラメータを修正した(一次的修正)時，その影響が及ぶパラメータも修正する(二次的修正)必要があるという関係を，メッセージの送信を監視することにより自動的に操作できるようになる．また，個々の知識の自律性が向上し，モジュール性も高くなり，知識の入れ換えなども容易になる．

オブジェクトに対する処理は，フレームを実現する手法[15]でもある属性リストや連想リストを用いて実現しており，オブジェクトの継承関係や変数，メソッドを記述するための構造と各種のメッセージを定義した上で，それに対するアクセスをメッセージの送信に相等するLISP関数SEND-MESSAGEに限定することにより，透明な知識表現を実現した．以下，このようなオブジェクト表現とその具体的な知識処理の内容について示す．

4・1・1　オブジェクトの構造と階層構造　　設計知識のオブジェクトは，1）演算や操作を行う設計手続きすなわちメソッドと，2）それに必要な情報や情報を格納する領域すなわち変数をそれぞれ一つの対象物にまとめたものである．これらのオブジェクトは，

設計変数などの実体を表現したインスタンスと，それらに共通する概念をまとめたクラスとから構成される(前出図２)．クラスは図３のシステムの汎用部①により用意され，そこには設計における基本処理を行う(インスタンス)メソッドが定義されている．インスタンスはユーザが定義し，個々の設計パラメータに関する知識を(インスタンス)変数として保持する．設計処理は，図６のように，①インスタンスに対してメソッドを起動するメッセージを送信し，クラスから②継承されたメソッドが③そのインスタンスに対して働きかけるとともに，さらに④他のインスタンスに対してメソッドを起動するメッセージを送信することにより実行される．

以下，このオブジェクトに定義されるメソッドと変数(Variable)，およびクラスとインスタンスの階層関係などについて説明する．

（１）(インスタンス)メソッド　　メソッドには次のようなものがあり，その機能は以下の通りである．

DECIDE：設計パラメータの値すなわち変数VALUEの値を参照し，既に定まっていればその値をメッセージの送信元に返す．未定であれば，変数PROCEDUREに従い値を決定して表示し，その値を返す．

QUARRY：変数VALUEを参照しその値をそのまま返す．

DELETE：ある設計パラメータの値を削除したい場合，変数VALUEを参照し，定まっていれば，その値を消去し，さらにその値に依存して決定された他のパラメータの値も消去する．

SET-UP：ユーザにその値を入力するか，変数PROCEDUREにより決定するかを問い，入力する場合は入力を促進し，入力値を変数VALUEの値とする．パラメータ間に依存関係がある場合には上記のDELETEが作用する．

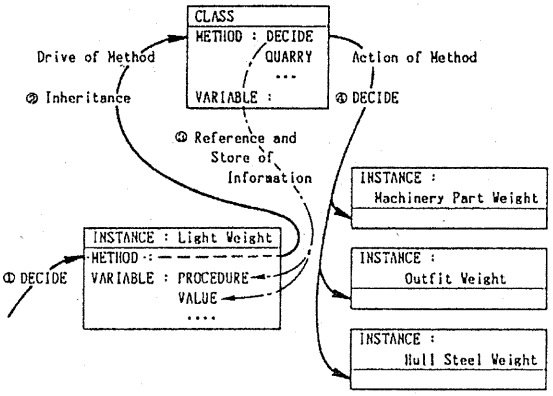


図６　オブジェクト間の継承と処理の連鎖

INPUT：ユーザにパラメータの入力を促進する．ユーザに確認しない点でSET-UPと異なる．

DISPLAY：変数VALUEを参照し，値が定まっていればその値を表示する．未定であれば未定であることを表示する．

これらのメソッドのうち，DECIDEとDELETEについては，４・１・３項でさらに詳細を示す．

（２）(インスタンス)変数　　変数には次のような名前のものがあり，各変数には以下の情報が格納され，メソッドによる処理の中で参照される．

PROCEDURE：設計パラメータの決定手続きの記述．

VALUE：設計パラメータの値．

RECORD：設計パラメータの値の履歴．

INPUT-SENTENCE：入力促進文．

DATA-RANGE：設計パラメータの値が数値であるか，そうでなければどのような値をとりうるかという情報．

以上のうち，PROCEDUREについては，４・１・２項で詳細を示す．INPUT-SENTENCEとDATA-RANGEは，パラメータの値を入力する時に用いる．VALUEとRECORDは，設計が進行する過程で設定される．

（３）クラス　　クラスには，上記のメソッドや変数を持つインスタンスの性質の違いにより，次の二つの種類がある．

①クラス"手続き"：設計パラメータの値が物理的関係などにより決定されるものと，実績値・標準値・入力値により暫定的に設定されるものに区分される．

②クラス"値"：設計パラメータの値の型による区分．すなわち，数値であるか非数値であるか，さらに，アトムとして表現されるかリストとして表現されるかにより区分される．

システムはこれらの二種類に対応して様々のクラスを用意する．前述のインスタンスは，上記①と②のそれぞれクラスから多重継承されるサブクラスに属するインスタンスとして作成する．

４・１・２　設計手続きの記述　　個々の設計パラメータの値の決定手続きは，変数PROCEDUREに記述されており，メソッドDECIDEにより操作され，設計パラメータの値を決定するために用いられる．

PROCEDUREには，他のパラメータの値やデータを引数として，①算術計算や論理演算，②条件式の記述，③設計候補の検索，④FORTRANあるいはLISPによる外部プログラムの呼出し，⑤知識ベースの操作などの手続きが記述できる．PROCEDUREを含めたインスタンスの記述例を図７に示す．これらは，船の基本設計に対するもので，(ａ)のフルード数は①の，(ｂ)の概略肥せき係数は②の，(ｃ)の類似船は③の例，(ｄ)の柱形

```
;
(INSTANCE (NAME FROUDE-NUMBER FN)
  (CLASS SNA)                                                    ...(a)
  (INSTANCE-VARIABLE
    (PROCEDURE ( R-VS // SQRT ( L * 9.808 ) )     )   )     )
;
(INSTANCE (NAME APPROX-BLOCK-COEFFICIENT A-CB)
  (CLASS SNA)                                                    ...(b)
  (INSTANCE-VARIABLE
    (PROCEDURE
      ( ( : DECIDE-BY-OTHER APPROX-BLOCK-COEFFICIENT )
          ` -->( A-DISP-F // ( 1.025 * L * DF * B ) )
        ( METHOD-OF-DPP * BY-TYPE-SHIP )
            -->( CB OF TYPE-SHIP )
                  MEAN-CB                          )        )
    (INPUT-SENTENCE
      " HOW MUCH IS BLOCK COEFFICIENT ? "  ) )                )
;
(INSTANCE (NAME TYPE-SHIP)
  (CLASS SNA)                                                    ...(c)
  (INSTANCE-VARIABLE
    (PROCEDURE
      ( SEARCH ( ABS ( DEAD-WEIGHT OF *** - R-DW )
                 + ABS ( VS OF *** - R-VS-KNOT )
                                  * R-DW // R-VS-KNOT ) )
      ( OBJECT-OF-SEARCH ( INSTANCE OF KIND-OF-SHIP ) ) ) ) )
;
(INSTANCE (NAME PRISMATIC-COEFFICIENT CP)
  (CLASS SNA)                                                    ...(d)
  (INSTANCE-VARIABLE (PROCEDURE ( $CP CP-CURVE  ) ) )        )
;
```

図7　インスタンスの記述例

係数は④の例である．

以下，これらの中で比較的複雑な処理である⑩，④と共通的な手続きの記述について説明する．

（1）検索　システムの設計では，設計要目の参考値やプラントなどを構成する機器の要目などを，過去の設計実績や機器要目のカタログデータなどを検索して定めることが多い．例えば，船の基本設計では実績船を検索したり，主機関の候補を検索したりする．このような処理を行うために，本システムは，検索対象のデータをオブジェクト(フレーム)として表現し，与えられた評価基準に基づいて検索を行う機能を持つ[7]．

（2）外部プログラムの呼出し　設計処理の中には，手続き的な部分が多く，知識工学的な処理と手続き的なFORTRANプログラムやLISP関数との連係が必要である．外部プログラムの呼出しは外部プログラム名の前に識別子'$'あるいは'#'を付加することにより，他のPROCEDUREと同様に記述でき，知識工学的処理の中から自由に呼び出すことができる．

（3）共通的手続きの記述　設計のプロセスでは，種々の条件の下で同じ種類のシミュレーションを行うことが多い．そのような場合に対応できるように，システムは，相異なる複数の設計パラメータのオブジェクトの間で手続きを共有させる機能を持つ．

以上が，設計手続きの記述である変数PROCEDUREの機能と記述方法である．このPROCEDUREを記述したオブジェクト間でメソッドを起動するメッセージを送信することにより種々の設計処理が行われる．

4・1・3　設計パラメータ間のネットワーク操作

2・2節でふれたネットワークにおける設計パラメータの値の決定および修正は，メソッドDECIDEとDELETEを組合せて行われる．この時，図5のような設計パラメータの間のネットワーク状の依存関係を操作する必要がある．一次的修正に伴う二次的修正を実行する方法には，各パラメータの決定における時間的な前後関係による方法，二次的修正を行うための知識をあらかじめ作成しておき，それにより推論を行う方法，値の決定における依存関係を利用する方法がある[16]．ここでは最後の方法を用いる．

パラメータの値を決定するときには，変数PRCEDUREの記述に従いメソッドDECIDEにより，図8のように値を必要とする他のパラメータのオブジェクトに対して再帰的にメソッドDECIDEを起動するメッセージが送信される．これによりパラメータ間に依存関係が生じる．二次的修正は，このような設計パラメータ間の依存関係を自動的に記録しておき，メソッドDELETEによりそれをたどって旧データを消去しつつ実行される．

以上のオブジェクト表現により，設計モデルの「生成」・「シミュレーション」・「評価」・「修正」などの過程を柔軟に支援することができる．

4・2　評価診断知識のルール表現とその処理

オブジェクト表現により作成された設計モデルは，設計過程において設計条件と設計結果との比較，法規の適用，経験的知識による評価検討が行われ，リファインされて満足解に近づけられる．本システムは，これらの知識をルールとして表現し，求められた各設計パラメータの値をルールの記述と照合して，問題点があれば，それの意味する述語を作成する機能を持つ．ルール表現の例を図9に示す．これは船の基本設計におけるもので，「船体の載貨重量が要求載貨重量の1.05倍より大きければ，船体の載貨重量が大きすぎる」「二重底の高さが幅の1/16より小さければ，二重底の高さが小さすぎる」などの知識を表現している．このようにルールの記述中に設計パラメータのオブジェクト名を記述するだけでメソッドQUARRYによりその値が参照され，ルールにより評価が行われる．この機能によりユーザの判断に助言を与えることができる．

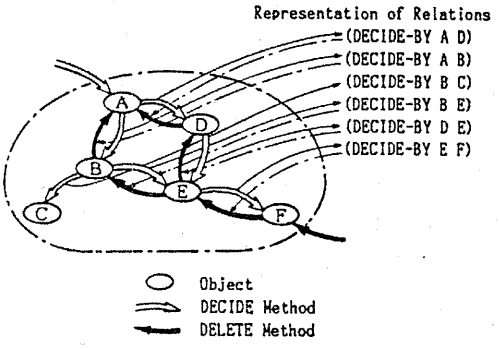


図8　ネットワークの操作

```
;
(RULE21-2 : IF ( &GREATERP DV ( 1.05 * R-DV ) )
            THEN ( &CREATE-PREDICATE
                     ( TOO-LARGE DV ( 1.05 * R-DV ) ) )  )
;
(RULE32-1 : IF ( &LESSP HEIGHT-OF-DOUBLE-BOTTOM ( B // 16.0 ) )
            THEN ( &CREATE-PREDICATE
                     ( TOO-SMALL HEIGHT-OF-DOUBLE-BOTTOM
                                          ( B // 16.0 ) ) ) )
;
(RULE34-2 : IF ( &GREATERP TRIM-ON-FULL-LOAD  0.30 )
            THEN ( &CREATE-PREDICATE
                     ( TOO-LARGE TRIM-ON-FULL-LOAD  0.30 ) ) )
;
(RULE34-3 : IF ( &LESSP   TRIM-ON-FULL-LOAD -0.05 )
            THEN ( &CREATE-PREDICATE
                     ( TOO-SMALL TRIM-ON-FULL-LOAD -0.05 ) ) )
;
(RULE36-1 : IF ( &LESSP    ( B // D ) 1.60 )
            THEN ( &CREATE-PREDICATE
                     ( TOO-SMALL ( B // D ) 1.60 ) ) )
;
(RULE36-2 : IF ( &GREATERP ( B // D ) 1.82 )
            THEN ( &CREATE-PREDICATE
                     ( TOO-LARGE ( B // D ) 1.82 ) ) )
;
```

図９　評価診断知識のルール表現

４・３　設計プロセス管理とユーザインタフェース

設計プロセス管理の機能により設計知識のオブジェクトや評価診断知識の操作が制御される．これにより図１に示した設計プロセスの進行が実現される．以下，これらの機能とそのために必要な知識表現について示す．

４・３・１　設計プロセス管理のための述語型知識

２・１節で述べたように設計のプロセスは，いくつかの段階に分れて進行する．このモデル化に基づき，設計プロセスの進行を管理し，設計結果を表示して判断の方向付けを行うためには，各段階で何を決定し何を表示するかを，あらかじめ設定しておく必要がある．以下の述語がこれを行う．

(STEP <N> <オブジェクト名>)：<N>番目の段階で，<オブジェクト名>のパラメータの値を決定する．

(DISPLAY <N> <オブジェクト名>)：<N>番目の段階で，<オブジェクト名>のパラメータの値を表示する．

(GRAPHIC-DISPLAY <N> <FORTRANプログラムを呼び出す記述>)：<N>番目の段階で，FORTRANプログラムを呼び出し，設計結果を表現した図形を表示する．

これらの述語の一例を図１０に示す．これも船の基本設計に対するもので，後出の図１１に対応する．(a)の表現により最初に主要寸法が決定され，(b)により次に性能の推定が行われ，さらに(c)により主機関の選定が行われ，それぞれ結果が表示される．また，(d)によりステップ３では船体形状が表示される．

４・３・２　設計プロセスの進行の管理

図１に示した設計プロセスの進行は，順序的な遷移とユーザの判断による分岐により実現できる．以下，図１の各フェーズにおける処理を示す．

（１）初期化　ファイルとして作成された設計知識をロードするなどの設計を行うための前処理を行う．

（２）設計　述語STEPを参照し，メソッドDECIDEを用いて，現段階で決定すべきパラメータの値を定める．

```
;
 (STEP 0 LENGTH)            (STEP 0 BREADTH) (STEP 0 DEPTH)
 (STEP 0 APPROX-FULL-LOAD-DISPLACEMENT)
 (STEP 0 FULL-LOAD-DRAFT) (STEP 0 APPROX-BLOCK-COEFFICIENT)
 (STEP 0 MEAN-CB)
 (DISPLAY 0 KIND-OF-SHIP) (DISPLAY 0 REQUIRED-DEAD-WEGHT)        ･･･(a)
 (DISPLAY 0 REQUIRED-VS-KNOT)
 (DISPLAY 0 APPROX-FULL-LOAD-DISPLACEMENT)
 (DISPLAY 0 LENGTH)         (DISPLAY 0 BREADTH)
 (DISPLAY 0 DEPTH)          (DISPLAY 0 FULL-LOAD-DRAFT)
 (DISPLAY 0 APPROX-BLOCK-COEFFICIENT)
;
 (STEP 1 APPROX-GM-ON-FULL-LOAD) (STEP 1 APPROX-GM-ON-BALLAST)
 (STEP 1 APPROX-DEAD-WEIGHT)     (STEP 1 P-VALUE-ON-FULL-LOAD)
 (STEP 1 P-VALUE-ON-BALLAST)
 (STEP 1 NEEDED-VG-ON-GREN-FULL-LOAD)
 (STEP 1 APPROX-VG-WITH-TOP-SIDE-TANK)
 (DISPLAY 1 KIND-OF-SHIP) (DISPLAY 1 REQUIRED-DEAD-WEIGHT)
 (DISPLAY 1 REQUIRED-VS-KNOT)                                    ･･･(b)
 (DISPLAY 1 APPROX-FULL-LOAD-DISPLACEMENT)
      - - - - -
 (DISPLAY 1 NEEDED-VG-ON-GREN-FULL-LOAD)
 (DISPLAY 1 APPROX-VG-WITH-TOP-SIDE-TANK)
;
 (STEP 2 MAIN-ENGINE)
 (DISPLAY 2 A-PS-MCO) (DISPLAY 2 KIND-OF-MAIN-ENGINE)
      - - - -
 (DISPLAY 2 LENGHT-OF-MEIN-ENGINE) (DISPLAY 2 FUEL-CONSUMPTION)  ･･･(c)
;
 (STEP 3 BODY-PLAN)
 (GRAPHIC-DISPLAY 3
   ( $GRAPHSP LENGTH BREADTH FULL-LOAD-DRAFT DEPTH CP-CURVE
      L-V-L UPPER-DECK-FORM VERTICAL-SECTION-FORM BODY-PLAN ) )    ･･･(d)
 (DISPLAY 3 BLOCK-COEFFICIENT)     (DISPLAY 3 CM)
 (DISPLAY 3 PRISMATIC-COEFFICIENT)
 (DISPLAY 3 FULL-LOAD-DISPLACEMENT)
;
```

図１０　設計プロセス管理のための述語型知識

（３）表示　述語DISPLAY，GRAPHIC-DISPLAYにより現段階における設計状況を表示する．

（４）ユーザの判断　ユーザに対して判断を促し，その判断に応じて各フェーズに分岐する．

（５）推論(診断)　ルール型知識により設計結果を評価診断し，その結果を表示する．

（６）修正要目の入力と修正　ユーザに対して，どの要目を修正するかを問い，それに応じてメソッドSET-UP，あるいはINPUTにより修正を行う．

（７）終了　設計結果が満足と判断されたとき，設計を終了する．

４・３・３　ユーザインタフェース

設計におけるエキスパートシステムは，あくまでも支援システムであり，ユーザインタフェースが重要となる．以下，設計状況の表示とユーザからの値の入力について示す．

（１）設計状況の表示　設計の進行状況は，メソッドDECIDE中からメソッドDISPLAYを起動することにより表示される．さらに，表示のフェーズでは，述語GRAPHIC-DISPLAYにより図形の表示ができ，また，述語DISPLAYにより各オブジェクトの変数RECORDを参照し，修正を繰返すことにより各パラメータの値がどのように変化したかが感度解析的に表示される．一方，診断のフェーズでは，評価診断の結果の述語が英文に変換されて表示される．

（２）入力の簡略化　入力を容易にするため，数値以外の入力はできるだけ選択肢の中から選ぶ形式をとる．各オブジェクトの変数DATA-RANGEはそのために用いられる．

以上が，本システムにおける知識表現とその処理，および，それにより実現される機能である．本システムは大型コンピュータACOS1000上にUTILISP言語により記述されており，設計のプロセスを柔軟に支援することができる．

5．船の基本設計プロセスへの適用

本システムを船の基本設計に適用した事例[17]を以下に示す．

5・1　船の基本設計過程　　事例としてとりあげる船の基本設計[18]の過程は，要求される船のサイズや速力を満たし，かつ安全性や運航コストなどの評価基準を満足化するように，船の要目(長さ，幅，深さなど)を定めるプロセスである．この過程は，数理計画法による最適化とは異なり，多くの目標に対して設計者の自由な裁量による試行錯誤により総合的な最適化を行うプロセスである．このプロセスでは，図１１のように，要求される機能に対して船の要目を仮想し，種々のシミュレーションを行い，その結果をフィードバックし船の要目を修正するプロセスを繰返す．しかし，これらの処理は複雑に連鎖し，前出図５のネットワークが設計全体に対して形成される．このようなネットワーク状の関係を設計者の能力のみで処理することは困難であり，本システムの手法を導入する必要がある．

5・2　船の基本設計に対する知識表現　　船の基本設計に対する知識は前章で示した知識表現により知識ベースに登録される．

5・3　実行例　　図１２はシステムの実行例で，

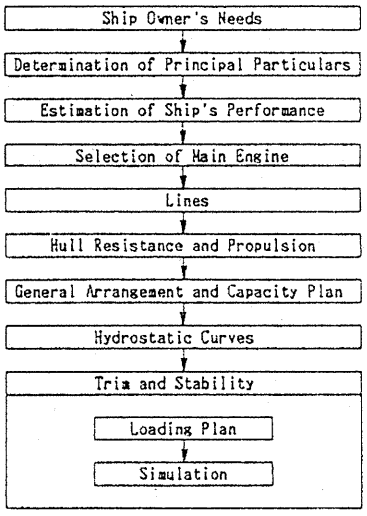
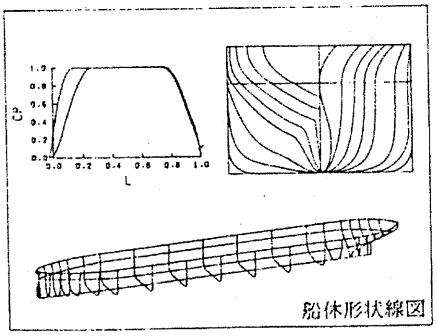


図１１　　船の基本設計過程

```
------------------------------ INITIALIZATION ------------------------------
    知識ベースがロードされ，初期化が行われる．
------------------------------ STEP 0 DESIGN START -------------------------
DECIDE FOLLOWING DESIGN VARIABLE
   LENGTH BREADTH DEPTH APPROX-FULL-LOAD-DISPLACEMENT FULL-LOAD-DRAFT
   APPROX-BLOCK-COEFFICIENT AGAN-CB-BY-WATSON
  ステップ０では長さ・幅・吃水などの船体主要目が決定される．
>>>  HOW DO YOU WANT TO DECIDE PRINCIPAL PARTICULARS ?
>>> ( 1 - BY-TYPES-SHIP , 2 - BY-STATISTICAL-DATA , )
   ... ? 2    船体主要目の決定方法を決める．
### METHOD-OF-DECISION-OF-PRINCIPAL-PARTICULARS = BY-STATISTICAL-DATA
>>>  WHAT KIND OF SHIP DO YOU WANT TO DESIGN ?
>>> ( 1 - CARGO-SHIP , 2 - BULK-CARRIER , 3 - CONTAINER-SHIP , )
   ... ? 2    船の種類を入力する．
### KIND-OF-SHIP = BULK-CARRIER
>>>  HOW MUCH WEIGHT IS REQUIRED DEAD WEIGHT [ TON ]  ?
   ... ? 38000.0    載貨重量を入力する．
### REQUIRED-DEAD-WEIGHT = 38000.
### APPROX-FULL-LOAD-DISPLACEMENT = 46790.40
### LENGTH = 187.3795
### FULL-LOAD-DRAFT = 11.10924
------------------- 中略 ------------------------------------------------
ステップ０の設計が進行し，結果が表示される．
それに対して診断を行った結果問題はなく，ステップ１へ進む．
------------------------------ STEP 1 DESIGN START -------------------------
DECIDE FOLLOWING DESIGN VARIABLE
   APPROX-GM-ON-FULL-LOAD APPROX-GM-ON-BALLAST APPROX-DEAD-WEIGHT
   P-VALUE-ON-FULL-LOAD P-VALUE-ON-BALLAST NEEDED-VG-ON-GREN-FULL-LOAD
   APPROX-VG-WITH-TOP-SIDE-TANK
   ステップ１では性能の推定が行われる．
>>>  HOW MANY SCREW DOES THE SHIP HAVE ?
>>> ( 1 - SINGLE , 2 - MULTI , )
   ... ? 1
### NUMBER-OF-SCREW = SINGLE
### APPROX-WATERPLANE-COEFFICIENT = 0.8882088
### APPROX-DRAFT-ON-FULL-LOAD = 11.10924
### APPROX-KB-ON-FULL-LOAD = 5.818456
### APPROX-BM-ON-FULL-LOAD = 5.415808
### APPROX-KG-ON-FULL-LOAD = 8.602866
### APPROX-GM-ON-FULL-LOAD = 2.631397
------------------- 中略（ステップ１の設計が進行し，結果が表示される）-------
------------------- DESIGN RESULTS ------ STEP = 1 ------ TIME = 1 ----------
 KIND-OF-SHIP ............ BULK-CA |
 REQUIRED-DEAD-WEIGHT .... 38000.0 |
------------------- 中略 ------------------------------------------------
 NEEDED-VG-ON-GREN-FULL-L  53199.9 |
 APPROX-VG-WITH-TOP-SIDE-  50713.4 |
-----------------------------------------------------------------------------
  D - NEXT STEP DESIGN , I - INFERENCE , C - CORRECT & UPDATE , E - END
-----------------------------------------------------------------------------
   ... ? I    設計結果に対する評価診断を行う．
------------------------------ APPLY DIAGNOSIS RULE ------------------------
 DEAD-WEIGHT WILL BE TOO SMALL AS COMPARED WITH REQUIRED-DEAD-WEIGHT .
 VG-WITH-TOP-SIDE-TANK WILL BE TOO SMALL AS COMPARED WITH
 NEEDED-VG-ON-GREN-FULL-LOAD .    診断の結果が表示される．
-----------------------------------------------------------------------------
  D - NEXT STEP DESIGN , I - INFERENCE , C - CORRECT & UPDATE , E - END
-----------------------------------------------------------------------------
   ... ? C    診断の結果により修正を行う．
>>> PLEASE INPUT VARIABLE NAME WANTED TO CORRECT !      ( E - RE-DESIGN )
   ... ? A-DISP-F    満載排水量を修正する．
>>> PLEASE INPUT VARIABLE NAME WANTED TO CORRECT !      ( E - RE-DESIGN )
   ... ? E
>>> SELECT VARIABLE NAME DECIDE BY OTHER MAIN DIMENSION VARIABLE .
>>> ( 1 - LENGTH , 2 - FULL-LOAD-DRAFT , 3 - APPROX-BLOCK-COEFFICIENT ,
      4 - BREADTH , )
   ... ? 1
------------------------ DELETE & SET-UP DESIGN VARIABLES ------------------
### P-VALUE-ON-BALLAST IS DELETED .    満載排水量の修正による
### APPROX-GM-ON-BALLAST IS DELETED .    二次的修正が行われる．
### APPROX-KB-ON-BALLAST IS DELETED .
------------------- 中略 ------------------------------------------------
### APPROX-FULL-LOAD-DISPLACEMENT IS DELETED .
>>>  HOW MUCH WEIGHT [ TON ] IS FULL LOAD DISPLACEMENT ?
>>>   PRE-VALUE  1: 46790.4
   ... ? 47990.4    満載排水量の値を入力する．
------------------- 中略 ------------------------------------------------
### LENGTH IS DELETED .
------------------------------ STEP 1 DESIGN START -------------------------
DECIDE FOLLOWING DESIGN VARIABLE
------------------- 中略 ------------------------------------------------
  再び，ステップ１の性能の推定が行われ，結果が表示される
------------------- DESIGN RESULTS ------ STEP = 1 ------ TIME = 2 ----------
                          PAST  1                    NOW
 KIND-OF-SHIP ............          |  | BULK-CA |
 REQUIRED-DEAD-WEIGHT ....          |  | 38000.0 |
 REQUIRED-SERVICE-SPEED ..          |  | 14.5999 |
 APPROX-FULL-LOAD-DISPLAC  46790.4 |C-| 47990.3 |
 LENGTH ..................  187.379 |-| 192.185 |
------------------- 中略 ------------------------------------------------
   ... ? C    さらに，修正を行う．
  問題点がなくなるまで，このような修正を繰り返す．
  ３回の修正を行った結果が以下の表である．
------------------- DESIGN RESULTS ------ STEP = 1 ------ TIME = 4 ----------
                          PAST  1       PAST  2       PAST  3        NOW
 KIND-OF-SHIP ............          |  |            |  |            |  | BULK-CA |
 REQUIRED-DEAD-WEIGHT ....          |  |            |  |            |  | 38000.0 |
 REQUIRED-SERVICE-SPEED ..          |  |            |  |            |  | 14.5999 |
 APPROX-FULL-LOAD-DISPLAC  46790.4 |C-|            |  | 47990.3 |-| 47535.3 |
 LENGTH ..................  187.379 |-|            |  |            |  | 192.185 |
 BREADTH .................          |  |            |  |            |  | 27.1517 |
 DEPTH ...................          |  | 15.2181 |C-|            |  | 15.8999 |
 FULL-LOAD-DRAFT .........          |  |            |  |            |  | 11.1092 |
 APPROX-BLOCK-COEFFICIENT           |  |            |  | 0.80765 |C-| 0.79999 |
 APPROX-GM-ON-FULL-LOAD ..  2.63139 |-| 2.67485 |-| 2.40384 |-| 2.41406 |
 APPROX-GM-ON-BALLAST ....  3.69716 |-| 3.96671 |-| 3.65723 |-| 3.66159 |
 TYPE-OF-MAIN-ENGINE .....  DIESEL  |  |            |  | DIESEL  |  | DIESEL  |
 APPROX-PS-MCO ...........  8650.56 |-|            |  | 8598.05 |-| 8543.51 |
 APPROX-LIGHT-WEIGHT .....  8957.73 |-| 9285.61 |-| 9374.23 |-| 9370.40 |
 APPROX-HULL-STEEL-WEIGHT   7482.95 |-| 7792.36 |-|            |  | 7840.98 |
 APPROX-OUTFIT-WEIGHT ....  854.906 |-|            |  |            |  | 887.088 |
 APPROX-MACHINERY-PART-WE  609.364 |-|            |  | 608.163 |-| 602.324 |
 APPROX-DEAD-WEIGHT ......  37532.6 |-| 38704.7 |-| 38615.1 |-| 38184.9 |
 P-VALUE-GM-FULL-LOAD ....  6.18646 |-| 6.13599 |-| 5.58867 |-| 6.57469 |
 P-VALUE-GM-BALLAST ......  6.31074 |-| 6.25518 |-| 6.62529 |-| 6.62234 |
 NEEDED-VG-ON-GREN-FULL-L           |  |            |  |            |  | 53199.9 |
 APPROX-VG-WITH-TOP-SIDE-  50713.4 |-| 52014.0 |-|            |  | 53661.0 |
-----------------------------------------------------------------------------
  D - NEXT STEP DESIGN , I - INFERENCE , C - CORRECT & UPDATE , E - END
-----------------------------------------------------------------------------
   ... ? I
------------------------------ APPLY DIAGNOSIS RULE ------------------------
    診断により，設計結果には問題はないと判断される．
-----------------------------------------------------------------------------
  D - NEXT STEP DESIGN , I - INFERENCE , C - CORRECT & UPDATE , E - END
-----------------------------------------------------------------------------
   ... ? D    次の段階の設計を行う．
------------------------------ STEP 2 DESIGN START -------------------------
    ステップ２では主機関が選定される．以下，省略．
```

図１２　実行例



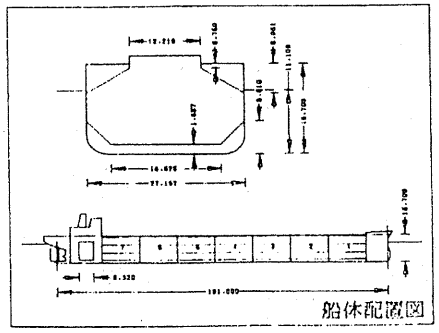


図１３　船の形状と配置の表示

ばら積み貨物船の基本設計を行った例で，実績値による性能の推定値をもとに修正を繰り返し主要寸法の決定を行ったものである．図のように設計は対話的に進行し，設計者は設計結果に対する診断の表示をもとに設計モデルの修正を行う．また，設計プロセス中の設計要目の履歴が感度解析的に表示され設計者の判断を支援する．この表中では，ある要目を増加させた場合 Ｃ＋ の記号で示され，その影響による他の要目の増減が ＋ または － で示される．このようにして次第に要目が固められる．さらに設計が進行すると図１３のように船体の形状や配置が表示される．

６．結　言

本研究では，設計モデルの表現手法としてオブジェクト指向の概念を導入し，汎用性を持つ設計エキスパートシステムを開発した．ここで用いた手法により，

（１）　設計知識をオブジェクトとして表現することにより，知識のモジュール性や自律性が向上し，知識の生成・追加・修正が容易となる．

（２）　設計処理をオブジェクト間のメッセージ・パッシングにより行うことにより，設計知識におけるネットワーク関係が操作でき，効率的な設計モデルの修正が可能になる．

（３）　ルール型知識により設計モデルの妥当性を検討し，モデルの修正を適切に助言することができる．

（４）　設計のモデリングにおける多様な知識表現，すなわちオブジェクト(フレーム)・述語・ルールのいずれの表現にも対応できる．

（５）　FORTRANなどによる外部の手続き型プログラムとの連係により，数値計算・最適化・図形表示などの処理と知識工学的な処理との融合が可能になる．

（６）　システムの構成を汎用的な設計知識処理部分と設計対象に固有な知識ベース部分に分離することにより，種々の設計対象への適用が可能なシステムである．

以上の諸機能によって，固有の設計知識の記述が容易になり設計プロセスを柔軟に支援することができる．

また，事例として船の基本設計へ適用することにより，開発したシステムおよび手法の有効性を確認した．

文　献

(1) Dixon,J.R., and Simmons,M.K., ASME *Computers in Mech. Eng.*, 2-1, (1983), 10.
(2) 赤木，機械の研究，38-1，(昭61)，養賢堂，1.
(3) 例., Hayes-Roth,F., Waterman,D.A., and Lenat,D.B., *Building Expert System*, (1983), Addison-Wesley.
(4) 日経コンピュータ(別冊)，ＡＩ(人工知能)：実用化への夜明け，(昭61)，日経マグローヒル.
(5) Sriram,D., and Adeg,R.,el., *Applications of Artificial Intelligence in Engineering Problems*, I.II, 1st Int. Conf., (1986), Springer-Verlag.
(6) 赤木，工学設計における知識工学の応用，大阪大学知識科学研究会第3回研究会資料，(昭61)，9.
(7) 赤木ほか4名，機論，53-486，C (昭62)，512.
(8) 例., 鈴木編，オブジェクト指向，(昭60)，共立出版.
(9) Cox,B.J., *Object Oriented Programming : An Evoluationary Approach*, (1986), Addison-Wesley.
(10) Goldberg,A.J., and Robson,D., *Smalltalk-80 : The Language and Its Implementation*, (1983), Addison-Wesley.
(11) Goldberg,A.J., and Robson,D., *Smalltalk-80 : The Interactive Programming Environment*, (1984), Addison-Wesley.
(12) Weinreb,D. and Moon,D., *Object, message Passing, and flavors, LISP machine manual*, (1981), Symbolics Inc.
(13) Bobrow,D.G. and Stefik,H., *The LOOPS Manual*, working paper, Memo KB-VLSI-81-13, (1983), Xerox Corp.
(14) MacCallum,K.J., Proc. 1st Int. Marine System Design Conf., (1982), 1.
(15) 例えば，Winston,P.H. and Horn,B.K.P., *LISP*, (1981), Addison-Wesley.
(16) 吉田ほか3名，日立評論，67-12，(昭60)，日立製作所，971.
(17) 赤木，藤田，関西造船協会誌，206，(昭62.9)，関載予定.
(18) 富田，船舶基本設計論，(昭57)，丸善出版サービスセンター.

討　論

〔質問〕　小島俊雄
〔工業技術院機械技術研究所〕

設計問題とオブジェクト指向概念との関係を対応づけ，具体的な課題に適用，評価されたことに敬意を表す．

貴論文は設計問題の中でも，いわゆるパラメトリック設計を中心に考察されていたが，設計プロセスには，設計対象の構造を生成したり解析・評価する場面も存在する．そこで以下について質問する．

（1）　貴システムの開発・保守担当にはどのような分野の専門家を想定しているか，またシステムの利用者はどのような人か．

（2）　貴システムのユーザは，オブジェクト指向概念を直接意識しないで作業することが可能か．

〔回答〕　ご指摘のように，本システムはいわゆるパラメトリック設計を対象としており，設計対象の構造そのものの生成が重要な要素となる問題には，別途対応する必要がある．

（1）　本システムは，設計解を内部で完全に生成するのではなく，ユーザとの対話を通じて設計の過程を支援するものであり，ユーザといえども対象とする問題を十分理解した専門家である必要がある．また，システムの開発・保守は，対象とする設計問題を熟知した人が本システムにおける設計モデルなどを理解したうえで行うことが望ましい．

（2）　ユーザがオブジェクト指向概念を意識する必要はないが，設計知識を知識ベースに記述する開発者は，本システムにおける知識のモジュール化の意義，特にメッセージパッシングによる設計の進行を十分に理解する必要がある．

〔質問〕　新野秀憲〔東京工業大学工学部〕

（1）　図1のモデルは設計項目を逐次，一連のプロセスの中で決定・評価する場合には有効だが，多くの機械システムのように多様な設計要求を同時に満足する解を導出する際には有効に作動しないと考えられるがいかがか．

（2）　貴論文の「はん用設計知識処理システム」は，一般のエキスパートシステム開発用ツールと考えて良いか．

（3）　設計プロセス管理のために各段階で何を決定し何を表示するかをあらかじめ設定しておくことは，設計プロセスを柔軟に支援することにならず，システムのフレキシビリティを著しく減ずることにならないか．

〔回答〕　（1）　ご指摘のように，多様な設計要求を同時に満足する解を直ちに導出する場合には有効に作動しないと考えられるが，このような問題の解決は，設計・評価・修正の過程の繰返しを通じて，各種の設計要求を一つ一つバランスをとりながら満足させていき，設計解を導出するという手法によらざるを得ない．本システムは，設計パラメータ間の依存関係を自動的に処理することにより，上記の過程における設計者の判断を支援することをねらったものである．

（2）　「はん用設計知識処理システム」の部分は，一般のエキスパートシステム開発用ツールに相当する知識ベースの処理を行う部分に加えて，設計問題に固有な処理を行うためのクラスの定義やメソッドの定義を含んでいる．例えば，数式が容易に記述できるとか，FORTRAN プログラムとのリンケージにおいて，大形数値計算や図形処理に対応できるように，大量のデータの受け渡しが可能となっている．これにより，本システムは事例として取り上げた船の基本設計に類する各種の設計問題に対するツールとなっている．

（3）　設計プロセスの進行は，述語としてあらかじめ設定された手順に従って行われるが，これはユーザとの対話のためのものであり，本質的でかつ細部にわたる管理は，設計パラメータに対応させたオブジェクト間のメッセージパッシングにより，パラメータ間の依存関係に従ってフレキシブルに行われる．したがって述語は，固有の設計対象ごとに自由に設定することができるし，また，設計プロセスの各段階の順序を変更することも可能である．