知識情報処理に基づいた
基本設計支援システムの研究

平成元年12月

藤田喜久雄

知識情報処理に基づいた基本設計支援システムの研究

平成元年12月

藤田喜久雄

知識情報処理に基づいた基本設計支援システムの研究

目　次

第1章　緒論

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などの分野で次第に確立し，一部では定着しつつある．このような動きの起源は，ＭＩＴのＣＡＤプロジェクトにおける SKETCHPAD[1] に見られるが，設計におけるコンピュータ利用技術の変遷・発達はコンピュータそのものの性能に大きく係わっている．各種の２次元や３次元のＣＡＤシステム，ならびに設計シミュレーションのためのＣＡＥシステムなどは今まさに普及の過程にあり，これらの技術は，情報処理のレベルから言えば，比較的定型的な数値処理の技術を基盤としている．しかし，設計過程の合理化に対する要求はますます大きなものとなってきており，従来の設計支援技術の高度化に加えて，設計本来のシンセシス過程そのものを支援しようとする試みが行われつつある[2][3]．このような試みは，人間本来の知的な設計作業をコンピュータによって支援しようとするものであり，その基礎となるコンピュータ技術として，人工知能(Artificial Intelligence，ＡＩ)[4][5] の技術，特に知識情報処理の技術に大きな期待が持たれている．ＡＩ研究の歴史は古く，コンピュータの出現の直後にまでさかのぼるが，1980年代に入ってからのエキスパートシステム[6]の実用化・商用化を経て，実用可能な技術として定着しようとしている．このような技術の基礎は，数値シミュレーションなどの技術とは異なり，いわゆる記号処理の技術にあり，より高度で複雑な処理をコンピュータ上に記述できる点にひとつの特徴がある．それによって従来の技術では記述することが困難であった設計シンセシスの問題を記述できるようになることが期待されている[2]．

一方，設計シンセシスに関する研究も，コンピュータの利用とは別に古くからドイツなどを中心にして，設計処理の科学的解明，設計方法論の確立などの面から行われており[7]，また，我が国においても，吉川[8]は設計過程を解明すべく「一般設計学」を提唱している．しかし，これらの研究の多くはコン

ピュータの利用による設計支援システムの構築を前提としたものではなく，上述のようなより高度な設計支援の確立に直接結び付くものではなかった．これに対して，今日のコンピュータのハード・ソフトにわたる発達が，実際の設計支援を前提とした設計シンセシスに関する研究を可能にし，また，そのような状況が設計研究の新しい方向として注目されつつある．一方，ソフトウェア科学や人工知能の分野では，プログラミング言語は，計算や処理を実行するための方法をコンピュータに与える手段であることに加えて，方法論を具体的に記述するための高級な言語であるとの認識が一般化しつつある．以上のように，設計シンセシスに対する支援を考えるとき，ＡＩ技術に代表されるコンピュータ利用技術の進展によって，設計に関する議論を進め，それによって得られる知見を設計支援に生かしていくための手段が現実のものになりつつある．

本論文では，上述のような状況を踏まえて，設計過程の中でも，基本設計におけるシンセシスに対する具体的な支援を実現することを目的として，いくつかの基本設計問題を取り上げて，知識情報処理の技術に基礎を置いた設計支援手法を提案し，それをもとにして構築した設計支援システムについて考察する．

第２章では，設計支援システムに知識情報処理技術を導入しようとする立場から，設計問題の性質や種類について述べ，多様な設計問題に対して効率的に知識情報処理技術を導入しようとする観点や基本的な方法について，具体的な設計問題に関連させて考察する．このような検討を通じて，設計過程の中でも実用上の比重が最も大きく，有効な支援を実現することが期待できる基本設計の過程を取り上げることの意義を述べる．さらに，設計支援システムの実現に関して，設計支援手法やシステムの構築手法の基盤となる知識情報処理の技術として，知識表現や探索技法について述べる．以上のような考察をもとに，本論文では，具体的な基本設計の問題として，機能設計，プラント機器構成設計，基本配置設計を取り上げる．

第３章では，パラメトリック設計に代表される機能設計を取り上げる．機能設計は設計対象物に求められる多様な設計要求を満足するように，その要目や寸法，形状などを定めていく過程であり，その特質は，明確なアルゴリズムが

存在するのではなく，設計者の裁量によって，設計解を仮定してはその機能を評価し修正を加えていくことによって最終的な設計解を得ようとする点にある．このような過程に対して，設計対象を表現し設計処理を進めていく枠組として設計対象の構成要素を節点とするネットワークモデルをオブジェクト指向プログラミングのもとで提案し，機能設計における「設計・評価・再設計」の過程を柔軟に支援するシステムについて述べる．さらに，設計処理のなかで鍵となる再設計の過程に対して，設計者に有効な指針を与えるための機能について示し，また，設計対象を実際に即した形式で表現するために，このモデルに形状処理を融合する方法について述べる．加えて，構築したシステムを船舶の基本設計に適用して，その有効性を検証する．

第４章では，エネルギープラントの機器構成設計を取り上げる．一般にプラントの機器構成設計は，あらかじめ与えられた各種の機器の中から，設計対象におけるエネルギー需要量を効率良く合理的に供給できるように，プラントの構成を決定し，それぞれの機器をカタログデータの中から選定する問題である．このような問題を扱う上での鍵は，各種のカタログデータや個々の設計候補を表現するデータベースの構成と，合理的なプラント候補の合成方法にある．本論文では，各種データの階層性やモジュール性，それらのデータに付随する様々な設計処理の性質，さらにプラント構成方式の階層的な分類に着目し，オブジェクト指向に基づいたデータベースの構成方法ならびに設計支援手法を提案する．また，以上の手法に基づいて，舶用動力プラント，ならびに，コージェネレーションプラントの設計を事例として取り上げ，具体的なシステムを構築し，その設計支援手法の有効性を検証する．

第５章では，発電設備などにおける大規模システムにおける基本配置設計を取り上げる．配置設計は設計対象を構成する各種の機器の位置を空間的な制約を満足し必要な機能を充足するように定める問題であり，その設計には多大な労力を要するため，コンピュータの援用による省力設計への期待は大きい．本論文では，配置設計の中でも，設計対象を構成する各機器のトポロジカルな位置関係を決定する基本配置設計の過程を取り上げ，知識情報処理技術のひとつ

である「制約指向」の考え方に従って，配置問題における各種の設計条件を制約として表現した上で，汎用的なアルゴリズムにより配置を行う手法を提案する．さらに，そのような問題の構成を吟味した上で，提案する方法論をオブジェクト指向プログラミングのもとでインプリメントして配置設計支援システムを構築する．また，このシステムを原子力発電所の基本配置設計に適用して，方法論ならびにシステム構築手法の有効性を検証する．

第６章では，本論文のまとめとして，取り上げた各設計問題に対する方法論ならびに支援システム構築法について総括する．

文　献

(1) Sutherland, I. E., SKETCHPAD: A Man-Machine Graphical Communication System, Technical Report 296, (1963), MIT, Lincoln Laboratory.

(2) 赤木，研究展望 設計分野におけるＡＩ応用技術，日本機械学会論文集Ｃ編，vol.55, no.516, (1989), pp.1848-1856.

(3)「知的ＣＡＤ」目指して飛躍する機械設計システム，日経ＡＩ別冊，エキスパート・システム最前線 ── 統合化への加速 ──，(1988)，日経ＢＰ社，pp.104-125.

(4) 白井・辻井，人工知能，(1982)，岩波書店.

(5) Rich, E., Artifical Intelligence, (1983), McGraw-Hill, (邦訳，廣田・宮村，人工知能，(1984)，マグロウヒル ブック).

(6) 例えば，Hayes-Roth, F., Waterman, D. A. and Lenat, D. B., ed., Building Expert Systems, (1985), Addison-Wesley, (邦訳，AIUEO，エキスパート・システム，(1985)，産業図書).

(7) 例えば，Pahl, G. and Beitz, W., Engineering Design, (1977), Springers-Verlag.

(8) 吉川，一般設計学序説 ── 一般設計学のための公理的方法 ──，精密機械，vol.45, no.8, (1979), pp.906-912.

第2章　設計支援技術とコンピュータ利用技術

2･1 緒 言

設計におけるコンピュータの利用は，ＣＡＤ／ＣＡＭ／ＣＡＥの各分野で確実に浸透しつつあるが，更なる利用技術の発展が望まれており，従来の枠組では扱うことが困難であった設計シンセシスの過程を有効に支援しようとする研究が様々な立場から行われつつある．このような研究のねらいは，人間本来の知的な設計作業を支援しようとするものであり，グラフィックスの技術などに基づいた製図支援ＣＡＤや３次元形状処理技術を核に各種の工学計算などを行おうとするＣＡＥの技術とはその性質を異にする．

本章では，次章以降で基本設計における具体的な問題を取り上げて，設計シンセシスに対する支援手法について議論するに先立ち，コンピュータによる設計支援の立場から，設計や設計過程の性質・種類について議論し，設計支援技術における知識情報処理技術の位置付けを明らかにする．まず，設計における情報処理をいくつかに分類し，それぞれに対して各種のコンピュータ利用技術を対応させることにより，設計本来のシンセシス過程に対する支援を考えていく上で知識情報処理技術が重要であることを示す．次に，設計シンセシスの具体的な内容について，それらをいくつかの種類やタイプに類別した上で，次章以降で取り上げる基本設計に対する個々の支援手法を中心にして，設計支援における知識情報処理の役割について具体的に論じる．さらに，以上の議論を踏まえて，知識表現と設計モデリングとの関係，ならびに，ＡＩ手法に基づいた探索技法と設計シンセシスとの関連について述べるとともに，本論文で一貫して用いるオブジェクト指向によるモデリング手法について示す．

2・2　設計における情報処理とコンピュータ利用技術

2・2・1　設計問題の性質と従来のコンピュータ利用

設計は要求される目標に対してそれを具現化する実体の像を生成する過程であり，与えられた設計要求に対して設計解を仮定しては，その設計解の有する機能を評価し設計要求を満足するかどうかを検討していくことにより進められていく．すなわち，設計における操作は設計目標と設計実体の像との間における写像関係の操作であり，そのような操作には，設計目標から実体の像を求めるという設計本来の操作と，それが要求される機能などを有しているかどうかを検討するという操作の二つが存在する．このうち，後者は具体的な物理対象に対してその振舞を求める作業であり，いわゆるアナリシスをベースとする順問題であるため，適切な物理法則を適用することができれば，原理的にはその解である機能などを求めることができる．これに対して，前者は振舞から実体を求める作業であり，いわゆるシンセシスをベースにする逆問題であるため，明確なアルゴリズムとして記述することが困難な場合が多い．しかし，設計の本質は前者のシンセシスの過程にあり，非常に高度で知的な作業が行われているため，その本質は未だに明らかにされていないように思われる．

一方，設計における情報処理の内容はシンセシス・アナリシスの両面において極めて多岐に渡り，そのなかには，シミュレーション，形状処理，最適化，検索などの様々な処理が含まれ，コンピュータによって扱いやすい部分からその利用が進められてきている(1)．例えば，シミュレーションについては，有限要素法などの技術に基づいたＣＡＥの技術は，もはや設計にはなくてはならないものとなりつつあり，また，形状処理についても，沖野による TIPS(2) の開発をはじめとして，今日では，３次元のソリッドモデラーなどの数多くの商用ツールが多方面で利用されている．さらに，最適化についても，赤木が総括しているように，いくつかの問題に適用され，設計における有効性が明らかにされている(3)．しかし，このような設計におけるコンピュータの利用の多くは，明確なアルゴリズムとして処理の内容を記述しやすい部分に対する支援

をねらったものであり，設計本来のシンセシス過程を有効に支援するには至っていない．また，最適化手法についても，一般にはシンセシス支援の有効な手段とされているが，設計問題への応用に関しては数学的な定式化などの点で本質的な解決には至っていない．

これに対して，設計本来のシンセシスの過程を何らかの形で支援しようとする立場から，インテリジェントＣＡＤ[4][5]といった名のもとで知識情報処理技術に基礎を置いた試みが行われつつある[6][7]．しかし，設計処理のなかには，本質的にコンピュータによる処理とは相入れない部分も存在し，有効な設計支援技術を確立し実際の問題に生かしていくためには，改めて，従来のコンピュータ利用技術と新たな技術として注目されている知識情報処理技術との相違や，それぞれの内容，さらに知識情報処理技術によって取り扱うことが可能になる設計処理の内容を検討しておくことが重要である．

2･2･2　コンピュータ利用技術の内容

コンピュータは，根源的にはアルゴリズムとデータ構造をもとにして動作する装置であり，設計支援に利用する場合にも問題に応じたアルゴリズムやデータ構造を与える必要がある．図2･1はコンピュータのプログラミングパラダイムの変遷[8]を示したものであり，アセンブリ言語にはじまり，科学技術計算には欠かすことができない FORTRAN，知識情報処理技術を支える LISP や Prolog などの言語，さらには各種のエキスパートシステム構築用のシェルまでもが，この流れの上にあるものとしてとらえることができる．このような状況は，コンピュータ自体のハードウェアの発達とともに，処理させようとする問題の特質に対応する形で，記述力がより高く柔軟性に富んだプログラミング環境へと変遷してきたものとみなすことができる．このような言語のなかでも，例えば，FORTRAN は言語としては古いものの，その特質が数値処理に向いているため今日でも広く用いられている．また，知識情報処理の分野でいわれる「知識表現」の技術は，エキスパートシステム構築の根幹を成すものであり，最も新しいものの一つである．その特徴は，その柔軟な制御構造や宣言的なデー

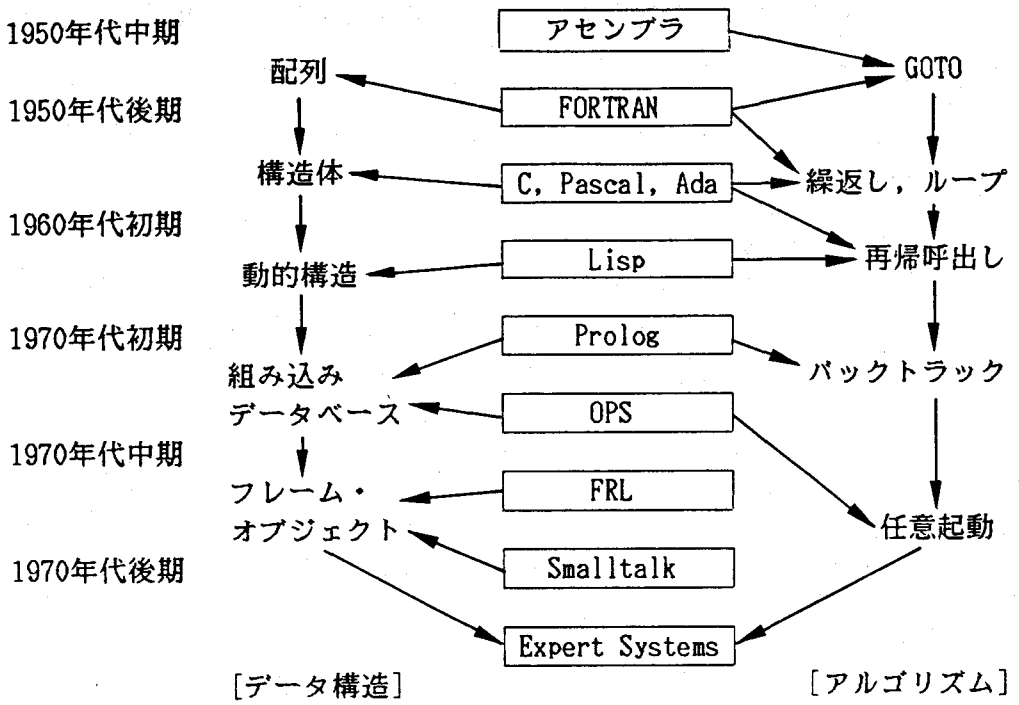


図2・1　コンピュータプログラミング言語の変遷[8]

タ構造によって従来のプログラミング言語ではカバーできなかった問題をカバーできるようになった点にある．また，従来のプログラミング言語がコンピュータそのものの構造に大きく依存したものであることに対して，知識表現による方法は，人間の認知や扱うべき問題の構成などから発生してきたプログラミング環境であり，そのため，単なるコンピュータ言語としての役割に止どまらず，取り扱うべき問題のモデリング手段や方法論を記述するための手段として認識されつつある[9]．

前項でも示したように設計におけるコンピュータの利用が定型的で数量的な処理にとどまってきた背景には，上記のような旧来のプログラミング環境にその一因があったことは否めない．今日における知識情報処理を援用した設計支援の試みにおいても，その記述能力の決定的な差異がひとつの契機になっている．一方，設計シンセシスの科学的な解明については，古くから，ドイツなどを中心に行われ[10]，我が国においても吉川[11]による研究がみられるが，実

際の設計支援を目指したものではなく，設計過程で行われている内容を一般的に文章表現する程度に止どまっている．しかし，最近の状況は，Dixon[12] が言うように，コンピュータ利用技術の進展によって，従来から行われていた設計の方法や理論に関する研究が計算可能なモデル（computational model）の構築を目標とするようになっており，より実用的な研究へと踏み出しつつある．言い換えれば，設計における支援技術に関して，従来のコンピュータ利用技術が，専ら，図形処理やシミュレーションなどのアナリシスをベースとする部分において数値処理を中心に援用されてきたのに対して，知識情報処理などの新しい技術が，設計シンセシスの内容を記述し支援していく上で，有効な手段であることが広く認識され，期待が持たれている．

次節では，以上のような点から，設計シンセシスの内容と，基本設計に対する支援手法について議論する．また，知識情報処理技術と設計支援との関連については，それを踏まえた上で2･4節で述べることにする．

2･3 設計シンセシスのタイプと設計支援

2･3･1 設計シンセシスのタイプとコンピータ化の度合

設計におけるシンセシスは，前述のように明確なアルゴリズムを記述することが困難な問題であり，いわゆる悪構造問題（ill-structured problem）[13] であるとされるが，その内容は個々の設計問題においてそれぞれに特徴となる一面を有している．そのため，その性質をとらえることにより，様々な設計問題はいくつかの種類やタイプに類別することができる．図2･2は，設計をプロセスという面から考え，それをいくつかのステップに分けて示したものであり，与えられた「設計仕様」に対して，「概念設計」･「基本設計」･「詳細設計」･「生産設計」の順に抽象度の高い処理から具体的な処理へと設計が進められていく様子を示している[6]．このような設計の各段階はそれぞれに異なった詳細度を持っているため，その処理内容も大きく異なったものとなる．それに対

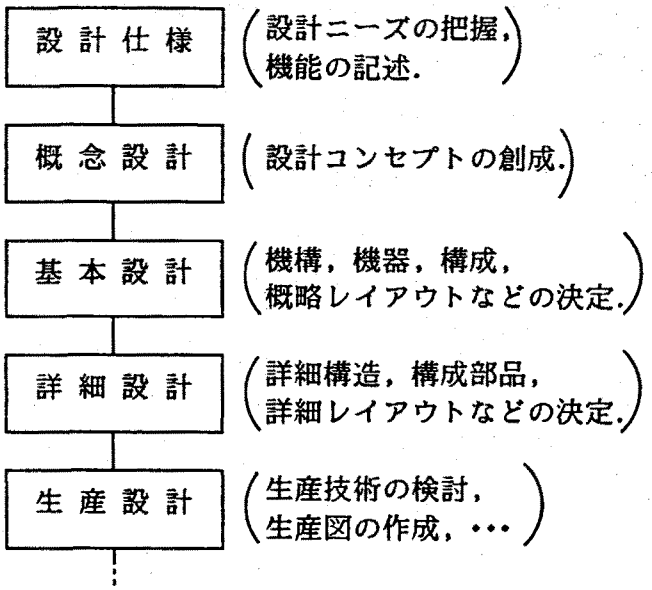


図2・2　設計のプロセス[6]

表2・1　設計プロセスと設計問題のタイプ[6]

	概念設計	基本設計	詳細設計
設計問題のタイプ	創成形設計		
	知的検索形設計		
	試行錯誤形設計		
		最適化形設計	

応して，各種の設計問題は以下のようなタイプに類型化することができ，上記の各設計プロセスにおける問題は，表2・1に示すようにそれぞれのタイプに対応する[1][6]．

・創成形：与えられた設計仕様に対して設計対象の構成そのものを定める設計操作であり，可能な構成を創成する処理が中心となる．このような問

題は概念設計において特徴的であり，典型的な問題としては，メカニズムを創成する問題などがあげられる．

・知的検索形：創成形と同様，設計対象の構成を定める設計操作であるが，知的検索形の問題においては，パターン化された構成をもとにして，設計仕様に合致する優れた構成を検索する処理が中心となる．このような問題は概念設計から基本設計において特徴的であり，典型的な問題としては，プラントや油圧システムなどのシステムを構成する問題などがあげられる．

・試行錯誤形：設計対象の構成を具体化していく設計操作であり，設計解の仮定とシミュレーションによる評価を繰り返して行うことにより，多様な設計目標を満足する設計解を具体化する処理が中心となる．このような問題は基本設計において特徴的であり，典型的な問題としては，各種の配置問題や配管問題，船舶や航空機などの機能設計などがあげられる．

・最適化形：明らかにされた設計対象の構成や構造に対してその詳細を定める設計操作であり，手順的に設計解を得ることができたり，部分的には数理計画法に基づいた最適化手法を比較的容易に適用することができる．このような問題は詳細設計において特徴的な問題である．

なお，このような分類は個々の設計対象に対応するものではなく，設計プロセスの進行とともに，様々な操作が行われることにも対応している．例えば，プラント設計においては，機器構成設計の部分は知的検索形の問題であり，配置設計や配管設計は試行錯誤形の問題，プラント構造物の設計などは最適化形の問題として理解することができる．

以上のような各タイプの設計問題におけるコンピュータ化の度合については，創成形の設計は，その本質が非常に高度で知的な作業であり，いわゆる創造を伴う場合が多いため，現在のところ，ほとんどコンピュータ化されていない．知的検索形の設計については，知識情報処理の技術により各種の設計知識やパターンとなる構成を表現して，有効な支援を行うことに期待が持たれる．また，試行錯誤形の設計に対しては，既に一部でコンピュータ化が進められ，また進

められつつもある．しかし，設計解に対する総合的な評価や設計対象の記述などの点で従来のコンピュータ利用技術による支援には限界があり，知識情報処理技術に期待が持たれる．一方，最適化形の設計はアルゴリズム化して取り扱うことが容易であり，既にコンピュータ化の度合が高い．しかし，大量の設計データの統合化された取扱いなどの点で今後の課題も大きい．

2･3･2　基本設計における設計シンセシスの支援

上記のような設計シンセシス過程の分類やその処理内容，さらに，前節で示した従来のコンピュータ利用技術などを振り返って整理すると，従来のコンピュータ利用技術が適用されてきた部分は，設計の内容を手続きとして記述することが比較的容易な部分であり，また，それぞれに独立した個別的な処理の部分でもある．それに対して，従来の利用技術を適用することが困難であった部分は，非定型的な処理の部分や数値処理とは馴染まないような処理の部分，ならびに設計者との協調した設計処理の部分であり，知識情報処理技術による支援に期待が持たれている．

基本設計における知的検索形の設計や試行錯誤形の設計についても，従来のコンピュータ利用技術では十分な支援が行われていなかった．これらのタイプの問題も，上述のように，非定型的な処理や設計者との協調した処理などを必要とする部分であり，知識情報処理による支援が期待でき，設計シンセシスの面では設計過程の中でも基本設計の比重が最も大きいことから，その実現による効果は大きい．これに対して，基本設計におけるシンセシスに対する具体的な支援手法として，以下のような立場からの支援が有効であると考えられ，期待が持たれる．

(1) 設計対象の操作過程に対してモデルを確立し，それをコンピュータ上に記述することによって，設計者に対して有効な設計対象の操作環境を提供することによる支援．

(2) 設計処理の内容をモデル化し，設計解の生成過程を明確にした上で，それをコンピュータ上に記述することによって，設計解を自動的に生成し

て，設計者に提示することによる支援．

(3) 設計において行われる複雑な操作に対応して，設計データの記述とそれに対する操作を統合化することにより，有効なデータベース技術を確立し，様々な設計データを総合的に操作できるようにすることによる支援．

(1)については，前述の「試行錯誤形の設計」に対する支援として有効であり，それによって設計者は柔軟に設計解を操作しながら設計を進めていくことができるようになる．(2)については，前述の「知的検索形の設計」に対する支援として有効であると考えられ，機械的な処理により設計解を合成しながら，いわゆる設計知識により設計解を絞りこんでいくことにより設計過程を支援することが期待できる．(3)については，各種の設計問題における設計解の表現や各種の設計データの統合的な取扱いに対して期待が持たれる．また，本論文で取り上げる各設計支援システムについても，第３章で取り上げる船舶の基本設計などの機能設計に対する支援手法は，ネットワークモデルによって設計者に有効な設計環境を提供しようとするものであり，上記の(1)や(3)に対応するものである．第４章で取り上げるエネルギープラントの機器構成設計に対する支援手法は，知的検索形の問題に対して，組合せにより設計解を合成するとともに，一方では，様々のカタログデータなどを有効に表現して処理しようとするものであり，上記の(2)や(3)に対応する．さらに，第５章で取り上げるプラントの配置設計に対する支援手法は，探索によって設計解を自動的に提示していき，設計者による試行錯誤を支援しようとするものであり，上記の(1)や(2)に対応する．

以上のような支援を実現するためには，高度のモデリング能力や問題解決能力が必要であり，前述のようにその手段として知識情報処理の技術が不可欠なものとなっている．このような背景には，知識情報処理技術が単なるプログラミング言語としてではなく，方法論の記述や対象物のモデリングの手段としても有効であることがあり，したがって，知識情報処理技術を設計支援に生かしていくためには，知識情報処理技術によって何が可能になるかをあらためて認

識することが重要である．次節では，このような点から知識情報処理の技術を整理しておく．

2･4　設計支援における知識情報処理技術

2･4･1　設計と知識情報処理技術

知識情報処理技術[14]は，人工知能[15]に関する研究のうち，応用に重きを置いた知識工学による技術として位置付けられ，いわゆる知識と呼ばれる高度な情報をコンピュータ上に蓄積し，推論によって様々な問題の解決に利用しようとする技術である．このような技術を現実の問題に適用して，多くのエキスパートシステムの開発が行われており[16]，設計問題を対象にした試みも見受けられる．また，今日における人工知能に関する研究は大きな広がりをみせており，人間の認知メカニズムの解明を目的とする認知科学に関する研究や，より高度な問題解決のための様々な論理に関する研究なども行われている．一方，設計の分野においても，知識情報処理に基づいた設計支援の試みが数多く行われている[4][6][7]．しかし，設計というもの自体がたいへん高度で知的な作業であり，その本質的な内容を明らかにすることは容易ではなく，前節で列挙した立場から知識情報処理の技術をとらえることが必要である．

前述の立場から設計支援を実現するためには，ひとつには，高度なモデリングをコンピュータ上に記述することが不可欠であり，そのような技術として「知識表現」に期待が持たれる．ＡＩ技術における知識表現は，問題を解決するために必要となる知識や事実をコンピュータ上に記述していくための枠組であり，認知科学的な研究などに基づいて「ルール」・「セマンティックネットワーク」・「フレーム」などの知識表現手法が提案されている[17]．しかし，2･2･2項でも述べたように，これらの知識表現の手法は文字通りの「知識」を表現するための手段としてのみならず，今日ではより一般的で高級なプログラミ

ング言語やモデリングの記述方法として認識される傾向にある．このような方法の特質として以下のような項目をあげることができる[17]．

(1) 記述力：表現すべき知識を人間にとって理解しやすい形式で，しかも宣言的に記述することができる．

(2) モジュール性：記述されるシステム（プログラム）の全体が個々のモジュールから構成されるようになるため，システムの構築や記述内容の追加や変更に対して容易に対応することができる．

(3) 柔軟性：取り扱おうとする問題の構成に対応して，柔軟に適切なシステムの構成をとることができ，効率的な問題解決を実現することができる．

このような性質は，設計シンセシスにおける支援を考えた場合，各種の設計知識はもちろんのこと，設計モデルや設計データベースの構成方法としても有効であり，知識表現は設計支援において核となる技術のひとつであると考えられる．なかでも，「オブジェクト指向[18][19]」に基づいた手法は，知識工学の分野のみならず，様々な情報処理の分野でプログラミング技法としても注目を集めており[20]，設計支援の分野においても大きな関心が持たれている．本論文の次章以降で論じる各設計支援システムにおいても，各システム構築の基本としてオブジェクト指向に基づいた手法を用いるが，設計支援技術におけるオブジェクト指向の有効性については，次項であらためて論じることにする．

以上のような知識表現に加えて，さらに，設計解の合成や検索といった点からは，ＡＩ手法における「探索」の技術[20]に期待が持たれる．探索は，問題の定義が与えられたときにその解を求める手段であり，古くから研究が行われるとともに，いくつかの基本的なアルゴリズムが提案されているほか，各種の問題の解決に適用されて，各方面で有効な手段となっている．しかし，旧来の探索が対象としている問題は設計における問題に比べてその構成は単純であり，従来の探索技法をそのまま設計問題に適用することはできない．したがって，上述の知識表現と同様，個々の設計問題の内容を吟味した上で適切な探索の手法を適用する必要があり，エキスパートシステムにおける経験的な知識との融合なども有効な手段の一つである．なお，ＡＩにおける探索とは別に，このよ

うな探索手法のひとつとして，数理計画法をあげることができる．数理計画法は，対象とする問題を数学的な形式で記述することができ，その性質が一定の条件を満たしている場合には，有効なシンセシスの手段であり，設計問題においても既に最適化手法の適用が行われている[3]．しかし，設計における問題の多くは定式化や数学的な解法などの点でそのような範疇に入らず，ＡＩ手法による探索手法の適用や両者の融合などに期待が持たれる．

2･4･2　オブジェクト指向と設計支援技術

オブジェクト指向の考え方の起こりは古く 1960 年代後半の研究にまでさかのぼることができ，その背景には各種の研究成果が含められているが，今日では多方面でその展開がみられ，情報処理に関する様々の分野で取り入れられている．例えば，プログラミング言語の世界では，言語自体がオブジェクト指向で構成されている Smalltalk-80[22][23] をはじめとして，LISP における Flavors[24] や CLOS (Common Lisp Object System)[25]，Prolog 系の ESP，Ｃ言語における Objective-C や C++ などがある．また，その適用は，オペレーティングシステム，データベース，シミュレーション，知識表現などの多岐に渡っている[20]．オブジェクト指向の考え方の基本は，従来のプログラミング技術の考え方では基本的に「データ」と「手続き」を分離して考える傾向が強かったのに対して，両者を分離せず「データ」とそれを操作する「手続き」を一体化して取り扱おうとする点にあり，これを基本としたオブジェクト指向の具体的な構成要素には以下のようなものがある[18]～[20]．

(1) オブジェクト：オブジェクトは，対象とする問題に現れる物理的なものや概念的なもののそれぞれに対応するプログラムの構成要素であり，データと手続きを一体化したモジュールである．オブジェクトはそれぞれに内部状態を持っており，それへのアクセスは自分自身の手続きによって自分自身が行う以外に許されず，ほかのものが勝手に外から操作することができないようになっている．また，オブジェクトにおいて手続きを実行する媒体をメソッドと呼ぶ．

(2) メッセージ：オブジェクトに対して処理を依頼し，メソッドを起動させる手段がメッセージである．このメッセージは，何をさせるかを表した標識と処理の中で必要となる情報を含んでいるのみで，具体的な処理の内容はメッセージを受けたオブジェクトに記述されているメソッドの内容によって定まる．

(3) クラスとインスタンス：同じような性質や機能をもったオブジェクトが存在する場合に，そのような共通となる性質や手続きを記述する枠組がクラスである．具体的な実体に相当するオブジェクトは，これと区別してインスタンスと呼び，インスタンスはいずれかのクラスに属する．

(4) 継承（インヘリタンス）：継承とは，より上位のクラス（スーパクラス）がもつ性質や機能をより下位のクラス（サブクラス）が受け継ぐという性質である．これによって，下位のクラスを定義する場合には，上位のクラスを指定して，上位のクラスは持たないが下位のクラスにおいて必要となる性質や機能のみを付加的に記述するのみで良くなる．

以上のような構成により，オブジェクト指向に基づいたプログラミング環境は，高度なモジュール性を有したものとなっており，また，システムが個々の部品の集合として構成されるようになるため，システムのプロトタイピングや段階的なシステムの構築などの面で有効な手段となっている．さらに，上記のような構成は，現実世界の様々な問題の構成に比較的容易に対応付けることができるため，扱おうとする対象問題のモデルを自然な形でコンピュータ上に記述することができる．

このため，オブジェクト指向は，設計支援技術の分野においても，対象モデルの記述やデータベースの構成などの点から注目されている[26][27]．すなわち，オブジェクト指向は，設計における複雑な対象や高度な処理内容を取り扱うとともに，自然で直感的な設計モデルをコンピュータ上に記述するための手法して有効であると考えられている．また，設計におけるデータベースについても，その性質が基本的に従来のデータベース技術では扱いにくいことが認識されており，オブジェクト指向に関心が持たれている．このほか，オブジェク

ト指向は，設計支援システムの構築の面からも有効な手段である．設計支援システムの開発過程では，その支援手法が最初の段階で明確に定まっている場合はむしろ少なく，システムを構築しつつ手法そのものも洗練化させていく必要がある．オブジェクト指向プログラミングを用いた場合，設計支援手法の具体化や詳細化に伴って，クラスを定義したりメソッドを追加したりすることにより，各種のオブジェクトを詳細化し成長させていくことにより，全体としての整合性を維持しつつ，システムを構築していくことができるようになる．

2･5 結 言

本章では，基本設計における具体的な問題を取り上げる準備として，設計支援システムに知識情報処理の技術を導入しようとする立場から，一般的に，設計問題の性質や種類について考察し，設計支援技術に知識情報処理を導入する必要性やその立場を明らかにした．その上で，知識情報処理技術の中で，知識表現や探索の技術を設計支援技術に生かしていくことが必要であり，なかでも，オブジェクト指向によるモデリング手法が有望であることを示した．

文 献

(1) 赤木，ＡＩ技術によるシステム設計論，(1988)，コロナ社．
(2) 沖野，自動設計の方法，(1982)，養賢堂．
(3) 赤木，設計論とコンピュータ利用技術（2），機械の研究，vol.38，no.2，(1986)，養賢堂，pp.273-277．
(4) 吉川・伊藤 編，特集 インテリジェントＣＡＤ，コンピュートロール，no.25，(1989)，コロナ社．
(5) 吉川・冨山 編，インテリジェントＣＡＤ(上)，(1989)，朝倉書店．
(6) 赤木，研究展望 設計分野におけるＡＩ応用技術，日本機械学会論文集

C編, vol.55, no.516, (1989), pp.1848-1856.
(7) 「知的ＣＡＤ」目指して飛躍する機械設計システム, 日経ＡＩ別冊, エキスパート・システム最前線 ── 統合化への加速 ──, (1988), 日経ＢＰ社, pp.104-125.
(8) Parsaye, K. and Chignell, M., Expert Systems for Experts, (1988), John Wiley & Sons.
(9) Abelson, H. and Sussman, G. J., Structure and Interpretation of Computer Programs, (1985), MIT Press.
(10) Pahl, G. and Beitz, W., Engineering Design, (1977), Springer-Verlag.
(11) 吉川, 一般設計学序説 ── 一般設計学のための公理的方法 ──, 精密機械, vol.45, no.8, (1979), pp.906-912.
(12) Dixon, J. R., On Research Methodology towards a Scientific Theory of Engineering Design, AI EDAM, vol.1, no.3, (1987), pp.145-157.
(13) Dixon, J. R., and Simmons, M. K., Computers That Design: Expert Systems for Mechanical Engineers, Computers in mechanical engineering, vol.2, no.3, (1983), ASME, pp.10-18.
(14) 例えば, 特集 知識工学, 情報処理, vol.25, no.12, (1985).
(15) 例えば, Rich, E., Artifical Intelligence, (1983), McGraw-Hill, (邦訳, 廣田・宮村, 人工知能, (1984), マグロウヒル ブック).
(16) 例えば, Hayes-Roth, F., Waterman, D. A. and Lenat, D. B., ed., Building Expert Systems, (1985), Addison-Wesley, (邦訳, AIUEO, エキスパート・システム, (1985), 産業図書).
(17) 石塚, 知識の表現と利用, 知識ベース入門 (大須賀 編) 第1章, (1986), オーム社, pp.13-55.
(18) 鈴木 編, オブジェクト指向, (1985), 共立出版.
(19) Cox, B. J., Object Oriented Programming: An Evolutionary Approach, (1986), Addison-Wesley.
(20) 大特集 オブジェクト指向プログラミング, 情報処理, vol.29, no.2, (1988).
(21) 例えば, 白井・辻井, 人工知能, (1982), 岩波書店.
(22) Goldberg, A. J., and Robson, D., Smalltalk-80: The Language and Its Implementation, (1983), Addison-Wesley.
(23) Goldberg, A. J., and Robson, D., Smalltalk-80: The Interactive Programming Environment, (1984), Addison-Wesley.

(24) Weinreb, D. and Moon, D., Object, Message Passing, and Flavors, LISP machine manual, (1981), Symbolics Inc.
(25) Keene, S. E., Object-Oriented Programming in COMMON LISP, (1989), Addison-Wesley.
(26) 木村，オブジェクト指向によるＣＡＤ／ＣＡＭのためのモデリングとデータベース，情報処理，vol.29，no.2，(1988)，pp.368-373.
(27) オブジェクト指向によるＣＡＤ／ＣＡＭ用データベース 小特集，システム制御情報学会誌，vol.33，no.8，(1989).

第3章　ネットワークモデルによる機能設計支援システム

3･1　緒 言

機能設計は，多様な設計要求を満足するように，設計対象物の要目や寸法，形状などを定める過程であり，典型的な「試行錯誤形」の問題である．このため，一般には明確なアルゴリズムは存在せず，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り返し，満足解に近づけることにより，最終的な設計解を求める．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．一方，このような問題に対する支援手法のひとつとして最適化手法が用いられてきたが，数学的な定式化の問題や複数の設計目標間の総合化などの点でなお困難を伴っている．

本章では，機能設計の過程における処理の特質が設計対象を構成する様々の要素間の関係を操作している点にあることに注目することにより，そのような要素を節点，要素間の依存関係をアークとするネットワークモデルを提案し，それを基本としてオブジェクト指向プログラミング[1]のもとで構築した設計支援システムについて述べる．まず，機能設計に関する分析から，その設計過程や処理の特質について述べた上で，ネットワークモデルについて示し，このモデルがオブジェクト指向プログラミングによって実現できることを示す．次に，設計支援の立場から，設計対象の表現や設計プロセスの記述，設計評価知識の表現方法を示す．さらに，機能設計のひとつであるパラメトリック設計の過程を支援するための機能として，やはりネットワークモデルに基づいた感度解析による機能や最適化手法に基づいた機能を提案して，設計者による「設計・評価・再設計」の繰返しを有効に支援するための方法について述べる．また，設計問題には欠かすことのできない形状処理をネットワークモデルを融合して，

設計対象を実体に即した形で取り扱い，設計過程をより広範囲に支援する手法について述べる．最後に，船舶の基本設計における主要目決定過程ならびに区画配置過程に適用して，提案した手法の有効性を検証する．

3･2　機能設計の過程とネットワークモデル

本節では，機能設計における問題点や設計プロセスに対する理解を示すとともに，設計対象ならびに設計処理過程のモデリング手法としてのネットワークモデルについて示す．

3･2･1　機能設計の問題点と従来の支援手法

緒言でも述べたように，機能設計は，多様な設計要求を満足するように設計対象物の要目や寸法，形状などを定める過程であり，一般には，設計者が「設計・評価・再設計」の過程を繰り返しながら設計対象を操作していくことにより設計解を得る．例えば，3･7節で事例として取り上げる船舶の基本設計は，要求される載貨重量や航海速力，さらに安定性などを満足するように，各種の要目を決定する問題であり，典型的な機能設計の問題である．このような設計過程は「デザインスパイラル」として象徴的に表現され[2]，その特質は多様な設計目標の間で総合化（ compromise ）をはかる点にある．一方，設計をはじめとするシンセシスに対する一手法として，従来，設計解を直接得ようとする数理計画法に基づいた最適化手法が用いられている[3]．機能設計に対しても，上記のような総合化の問題を最適化手法により取り扱おうとする試みが，船舶の基本設計に関する Lyon と Mistree による研究[4]をはじめとして行われている．しかし，そのような試みの多くは，以下の点で，最適化手法が機能設計とは必ずしも相入れないため，本質的な解決には至っていないように思われる．

(1) 設計解の評価の問題：設計解の評価規準のある部分は設計者の経験に依

存しているため，ある具体的な設計解が示されたときに設計者がそれを評価することはできるが，事前にそのような評価規準を明示することは困難である．

(2) 定式化の問題：数理計画法を適用するには設計問題自体の完全な定式化が必要であり，目的関数や制約条件を数式として表現しなければならない．一方，設計問題は本質的には不整構造（ ill-structured ）であり，上記の(1)をはじめとして，設計対象のすべてが完全に定式化できるとは限らない．

(3) 複数の設計目標間の総合化の問題：設計目標は一般には複数個存在し，いわゆる多目的最適化問題[5]となる．このような問題に対する数理計画法による解法もいくつかのものが提案されている[5]が，各設計目標間のバランスは，必ずしも明確に記述できるようなものではなく，最終的には設計者の判断に委ねられる．

これらの原因は，共通して設計解の評価に対する数学的な記述が困難である点にあり，機能設計においては，設計解の評価は最終的には設計者の判断によらざるを得ない．これに対して，MacCallum は，船舶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を行っている[6]．そのねらいは，機能設計における各種の機能を設計解に対してシミュレーション的に求められるようにして，設計者がフレキシブルに設計解を操作できるようにする点にあり，インタラクティブな設計解の操作を可能にすることにより設計を支援しようとするものであるが，船舶の例を引いて設計項目間のネットワーク上の関係を分析するに止どまっている．

本章で提案する手法は，以上のような点を踏まえて，設計項目間の依存関係をネットワークモデルとしてとらえ，オブジェクト指向プログラミングのもとで機能設計に対する強力な支援システムを構築しようとするものである．次項では，その基本的な考え方を示す．

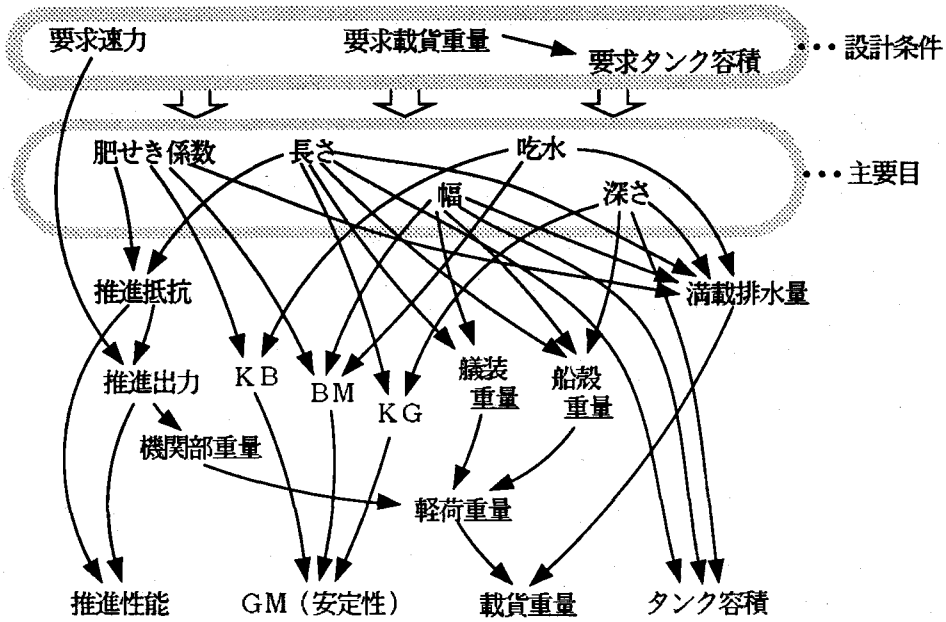


図3・1　設計におけるネットワークの一例

3・2・2　ネットワークモデルによる表現と設計プロセスの理解

上記の機能設計における問題点を明らかにするために，あらためて，船舶の基本設計の例を引いて，機能設計における各種の設計項目間の関係について考える．図3・1は，主要目の決定から機能の推定の部分に至る過程における各種の設計項目間の関連を示したものである（簡略化してある）．図のように，その過程では，まず，載貨重量や航海速力などの要求値が与えられ，それをもとにして船体の長さや幅などの各種の要目を設定する．このような設定は，一般には，過去の設計事例や経験によって定められるが，その要目が直ちに与えられた設計要求を満足するとは限らない．そこで，シミュレーション的にそのような機能や安全性などを推定し，その結果をもとに先に設定した各要目の値を修正し，そのような過程を繰り返すことにより，最終的にすべての要求を満足しバランスのとれた設計解を得るようにする．このような過程における問題は，設計者が柔軟に試行錯誤を繰り返す必要があるという点にある．

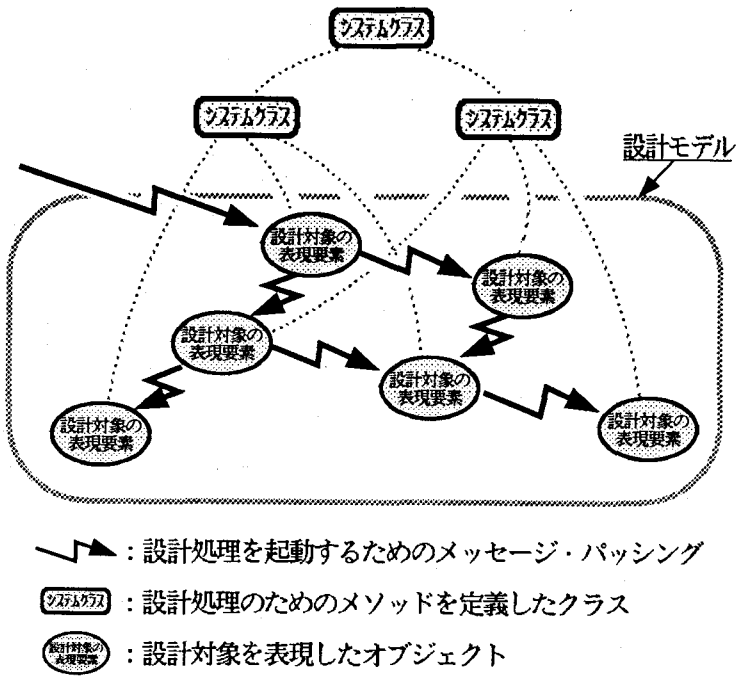


図3•2　ネットワークモデル

ネットワークモデル　　有効な設計支援のためには，上記のような過程に対して，シミュレーション的な設計操作が容易に実行でき，また，設計目標間の総合化をはかるための設計修正に対しても有効な指針を与える必要がある．本論文では，そのような設計支援を実現する基礎として各種の設計項目間の依存関係に着目した「ネットワークモデル」を提案する．すなわち，図3•1のような設計項目間の複雑な依存関係を処理するために，オブジェクト指向プログラミングに着目し，ネットワーク状の関係における '節点' を 'オブジェクト' や 'オブジェクトの変数' に対応させ，'アーク' に対応する設計処理の連鎖を 'メッセージ・パッシング' として処理することにより，各種の設計処理や設計知識の表現を実現するようにする（図3•2）．このようなモデルの導入により，

(1) 個々の設計項目をモジュール化してオブジェクトに記述することにより，全体の処理が自立的な個別処理の連鎖により行われるようになり，設計

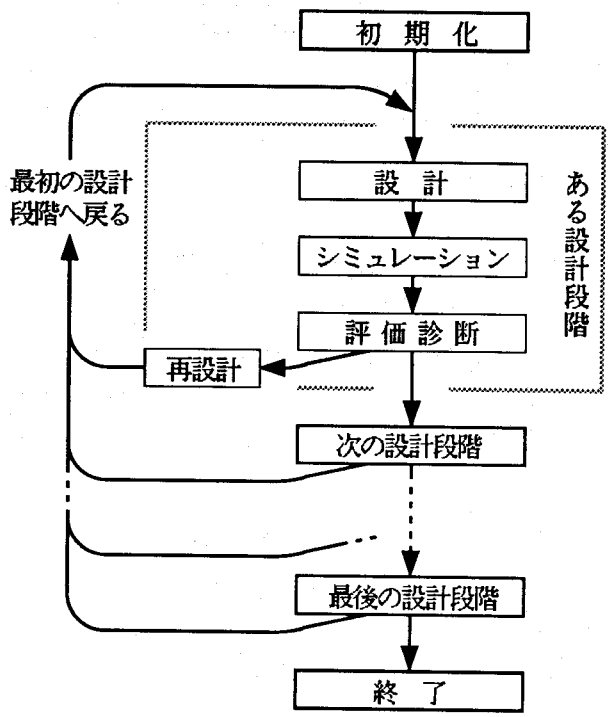


図3･3　設計プロセスのモデル化

支援システムの構築や設計対象の記述が容易になる．

(2) 機能設計における「設計・評価・再設計」の過程に伴う設計項目間の依存関係の処理をメッセージ・パッシングにより自立分散的に実現できる．

(3) 設計項目間の依存関係をシステムにおいて管理することが可能になり，設計処理の内容を明示的に設計者に把握させることが容易になり，また，再設計案を生成する過程を支援することができるようにもなる．

設計プロセスのモデル　　さらに，このようなネットワークモデルを実現する上で，上述した機能設計における「設計・評価・再設計」の繰返しによる設計の進行を，図3･3のように，以下の各フェーズの遷移として理解することにする．

(1) 設計：扱うべき設計案の生成，

(2) シミュレーション：設計案に対する機能シミュレーション，

(3) 評価診断：シミュレーション結果に基づいた設計案の評価，

(4) 再設計（修正）：評価結果に基づいた設計案の修正．

さらに，このような繰返しにおいては，設計対象の取扱いが徐々に詳細化されつつ，設計が進めていくものとして理解し，それをいくつかの設計段階に分けて考えることにする．例えば，3･7節で事例として取り上げる船舶の基本設計の場合には，その設計過程は後出の図3･22のようにいくつかの段階に分けて考えることができる．

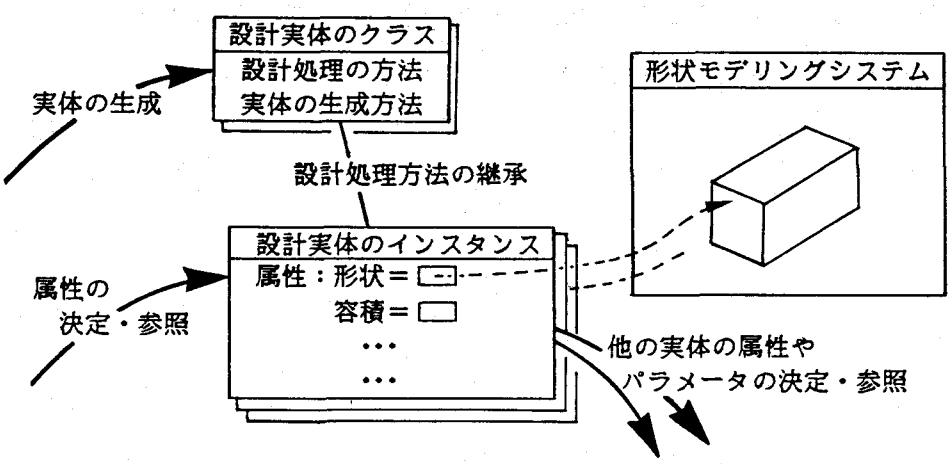
3･2･3 設計対象の表現方法

前項で示したような機能設計の過程では，設計対象を構成する様々の設計項目の内容が決定されたり修正されたりしながら操作されていく．このような設計対象は，「ネットワークモデル」においては '節点' として表現されるが，図3･1に示したような設計パラメータ間のネットワークを，さらに形状表現をも含んだ設計対象モデルへと展開するに当たり，以下に，そのようなモデルの構成を分類するとともに，オブジェクト指向による設計対象の表現方法の基本を示す．

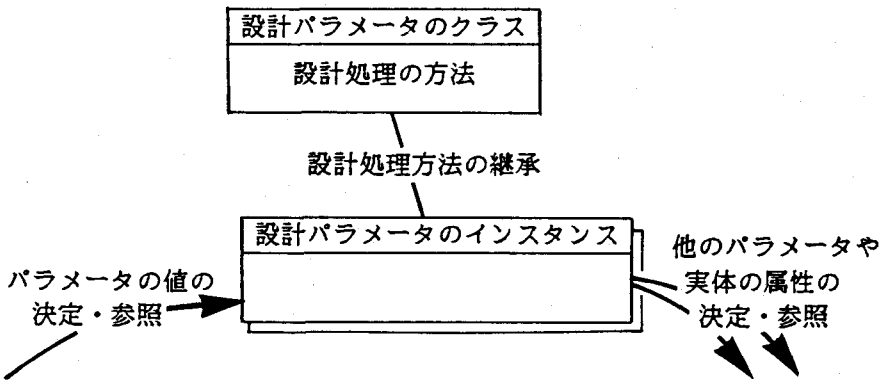
設計対象は，扱おうとする具体的な設計対象物に相当する設計実体に関する部分と，各種の要求機能や性能値などのパラメータに関する部分の二つに大別することができる．さらに，前者の設計実体は，そのような実体を生成する概念的な部分と，生成される実体自体とその属性に関する部分とに分類することができる．このような分類に対応して，設計処理を記述するにあたり，図3･2に示したネットワークモデルを構成するオブジェクトとして，以下の三種類のオブジェクトを用いる．

(1) 設計実体のクラス：後出(2)の設計実体のインスタンスを生成したり，その振舞，すなわち設計実体の属性の決定方法を保持する．

(2) 設計実体のインスタンス：実際の設計実体に対応するオブジェクトであり，(1)のクラスのインスタンスとして生成され，各種の属性値を保持する．



（a） 設計実体の表現



（b） 設計パラメータの表現

図3･4 オブジェクト指向による設計対象の表現

(3) 設計パラメータのインスタンス：各種の設計条件や評価項目などのような，設計実体の属性として保持することができない各種パラメータの決定方法を保持する．

図3･4は，このような表現における概念を示したものである．このような表現によって，設計実体はインスタンス（オブジェクト）として表現され，その生成・消去はインスタンスの生成・消去に対応するようになり，設計実体の属性はオブジェクトの変数として表現され，その操作は変数の操作に対応するようになる．また，設計パラメータについても，独立したオブジェクトの変数とし

て取り扱われるようになる．さらに，オブジェクト指向プログラミングの導入によって，各種の設計処理の方法についても，上記のオブジェクトに対するクラスにメソッドとして記述することにより，設計処理の過程を容易に記述することが可能になる．

なお，具体的なオブジェクトの記述やメソッドの内容，設計処理の方法については次節で後述する．また，上記(2)の設計実体のインスタンスにおける属性のうち，形状に関する属性は特に重要であり，形状もこのようなインスタンスの属性として扱うようにするが，その方法については3･6節で詳細に述べる．

3･3　設計支援システムの構成および知識表現とその処理

本節では，オブジェクト指向プログラミングによる設計支援システムの構成と，基本的な知識表現や処理の方法について示す．

3･3･1　システムの構成

本システムの構成を図3･5に示す．システムは，大きく分けて，汎用的な設計知識処理部と個々の設計問題に固有な知識ベースとから構成する．このようなシステムの分離により，一般のエキスパートシステム開発用シェルと同様の機能を持たせ，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り，機能設計に対する汎用ツールとすることができる．また，設計知識の追加や修正も容易となり，様々な設計問題に対応することができるようになる．

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ーフェースを取りつつ設計を行う部分であり，設計プロセスの進行を管理する部分，ユーザインターフェースを行う部分，知識ベースに対する処理を行う部分，設計対象の表現におけるオブジェクトに対するクラスを定義した部分から構成される．

また，個別問題に対する知識ベースは，前述の設計対象を表現する各種のオブジェクトの他に，形状処理を融合するための形状表現に関するオブジェクト，

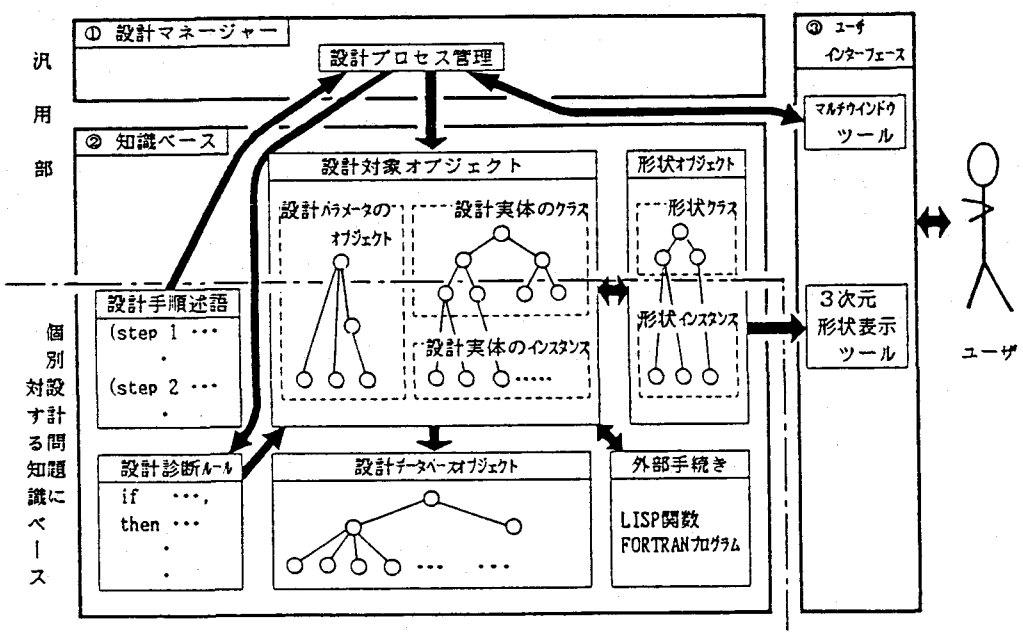


図3･5　システムの構成

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，設計候補の検索対象のデータを表現したオブジェクト，設計結果を評価診断するためのルール，数値計算や図形表示を行うための FORTRAN や C言語により記述されたプログラムや記号処理を伴う手続き的処理を行うための Lisp 関数からなる手続き型プログラムなどから構成される．以上のような多様な知識表現により，個別の設計問題における様々な知識に対応することができる．

なお，本システムは，エンジニアリングワークステーション Sun-3 上で構築され，Common Lisp[7] により記述されている．また，オブジェクト指向プログラミング環境は，Lisp 上のハッシュ表，構造体，連想リストなど[7][8]を用いて構築したものである．さらに，ユーザインターフェースの機能を記述するために SunView[9] を，3次元形状の表示のために SunCore[10] を用いた．

3・3・2　設計対象のオブジェクトによる表現

本項では，3・2節で前述したオブジェクト指向による設計対象の表現や処理について，具体的に述べる．

オブジェクトの構成と設計処理の継承　　本システムにおける設計対象の表現に関するオブジェクトは，3・2・3項の分類とは別に，システム構築の点から以下の二つに分けられる．

(1) システムの汎用部が用意するクラス（オブジェクト）：(2)の各オブジェクトに対するクラスであり，それらの内容に応じて数種類のものがあり，具体的な設計処理を行うための様々なメソッドを保持する．

(2) 個々の設計問題における設計対象を表現したオブジェクト：3・2・3項で示した３種類のオブジェクトであり，個別の設計問題に関する情報をそれぞれのオブジェクトの変数として保持する．

このような構成に対して，様々な設計処理は，(2)のオブジェクトが(1)のクラスに定義されたメソッドを継承することにより，後出の図3・8のようにして行われる．

設計処理を行うためのメソッドと変数　　設計処理を行うメソッドには，基本的なものとして，以下のものがあり，上記の(1)のクラスオブジェクトに定義される．

'decide'：指定された設計項目の内容を決定する．このとき，その内容が既に決定されていれば，直ちにその内容を返すが，そうでない場合には，内容の決定方法を記述した後述の変数 'procedure' を参照して，その内容をもとにして設計項目の決定処理を行い，その結果を保持するとともに，メソッドの起動元に返送する．

'delete'：指定された設計項目の内容を消去する．このとき，その内容に依存して決定された項目が存在すれば，その内容も消去すべく，再帰的に 'delete' を起動するメッセージが送信される．

'query'：指定された設計項目の内容を参照する．上記の 'decide' とは異なり，決定されていない場合には，何も行わない．

表3･1　設計処理メソッドの機能

オブジェクト / メソッド	設計実体のクラス	設計実体のインスタンス	設計パラメータのインスタンス
decide	自身の保持している実体の生成手続きに従って，設計実体のインスタンスを生成する．	指定された自身の属性の値を自身あるいはクラスが保持する決定手続きに従って決定し，保持する．	自身が保持している決定手続きに従って，対応するパラメータの値を決定し，保持する．
delete	自身の各インスタンスの属性値をメソッド 'delete' により消去した後，自身のすべてのインスタンスを消去する．	指定され自身の属性値を消去した上で，それに依存して決定された各種の値や実体のインスタンスをメソッド 'delete' により消去する．	自身のパラメータの値を消去した上で，それに依存して決定された各種の値や実体のインスタンスをメソッド 'delete' により消去する．
query	自身のインスタンスを参照する．	自身の指定された属性の値を参照する．	自身のパラメータの値を参照する．

'set-up'：ユーザによって指定された設計項目の内容を入力して設定する．このとき，その設計項目の変更前の内容に依存して決定されていた設計項目が存在する場合には，上記の 'delete' が作用する．

'display'：指定された設計項目の内容を参照し，それ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．

以上のメソッドの詳細な処理内容は，3･2･3項で前述した３種類のオブジェクトにより，表3･1のようにそれぞれに異なったものとなる．しかし，このようにメソッドの名称を統一することにより，設計処理を透明なものとすることができ，システムの構築も容易になる．これらのメソッドのうち，'decide' と 'delete' による処理については，3･3･3項でさらに詳細を示す．

また，各オブジェクトにおいて設計対象に関する情報を保持する変数には，設計項目の内容の決定方法を保持する 'procedure' をはじめとして，各パラメータや属性の値を保持するもの，設計の履歴を保持するものなどがある．なお，'procedure' の記述方法については以下に詳細を示す．

設計項目の内容の決定方法の記述　個々の設計実体における属性や設計パラメータの値を決定するための手続きの内容は，対応するインスタンスの変数 'procedure' に記述する．この 'procedure' には，他の設計項目の値やデータを引数として，算術計算や論理演算，条件式の記述，設計候補の検索，FORTRAN や C言語により記述された外部プログラムや Lisp 関数の呼出し，形状の定義やマスプロパティの算出などの形状属性に関する処理，知識ベースにおけるメッセージの送信，などの手続きを記述することができる．これらの中で比較的複雑な処理である検索と外部プログラムの呼出しについて，以下に説明する．

(1) 検索：機能設計では，設計要目の参考値や設計対象の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，船舶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，検索対象のデータをオブジェクトとして表現し，与えられた評価基準に基づいて検索を行う処理を融合することができる．なお，詳細については，本論文の第４章の中で示す．

(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 や C言語により記述されたプログラム，および，Lisp 関数との連携が必要である．外部プログラムの呼出しは外部プログラム名を指定することにより，他の 'procedure' と同様に記述でき，自由に呼び出すことができる．

また，形状属性に関する処理については，3･6節で後述する．一方，個々の設計実体のインスタンスを生成するための処理についても，設計実体のクラスの変数 'procedure' に記述する．

オブジェクトの記述例　図3･6は，上記のような 'procedure' を含めた設計実体のクラスとインスタンスの記述例である．これらは船舶の基本設計におけるものであり，(a)～(c)は設計実体のクラス，(d)はそれらのインスタンスの記述である．(a)は，船体の区画に共通するクラスであり，共通的な属性

```
(class (name tank-class)
  (super design-object)
  (iv (capacity nil
                (procedure (  integral 1                ))  )
      (oxg nil (procedure (- integral 'x / my capacity)) )
      (kg  nil (procedure (  integral 'z / my capacity)) )) )
```

（a） 船体の区画のクラス

```
(class (name cargo&tst-class)
  (super tank-class)
  (iv (parent-tank  nil)
      (pre-bulkhead nil (procedure 'dummy-bulkhead) )
      (aft-bulkhead nil (procedure 'dummy-bulkhead) )
      (geometry nil (procedure
         (let (( tank (geometry of my parent-tank)  )
               ( pre-bulkhead
                      (geometry of my pre-bulkhead) )
               ( aft-bulkhead
                      (geometry of my aft-bulkhead) ))
           (if (not (eq pre-bulkhead 'dummy))
               (setq tank (send tank 'get-lower-side-object
                                     pre-bulkhead) ))
           (if (not (eq aft-bulkhead 'dummy))
               (setq tank (send tank 'get-upper-side-object
                                     aft-bulkhead) ))
           tank )       ) ))  )
```

（b） 貨物倉とトップサイドタンクに共通のクラス

```
(class (name cargo-holds)
  (super cargo&tst-class)
  (cv (procedure
        (let (( n  number-of-bulkheads    ))
         (do (( i 1 (1+ i) )) ((> i n))
          (let (( inst (implode$ i      '-cargo-tank    ) )
                ( pre  (implode$ (1- i) '-cargo-bulkhead) )
                ( aft  (implode$ i      '-cargo-bulkhead) ))
            (send !self 'create-instance inst)
            (if (not (= i 1))
                (send inst 'put-value
                      'iv 'pre-bulkhead pre))
            (if (not (= i n))
                (send inst 'put-value
                      'iv 'aft-bulkhead aft)) )) ) ))
  (iv (parent-tank nil (procedure 'whole-cargo-tank) )) )
```

（c） 貨物倉のクラス

```
(instance (name 3-cargo-tank)
  (class cargo-holds)
  (iv (pre-bulkhead 2-cargo-bulkhead)
      (aft-bulkhead 3-cargo-bulkhead)
      (parent-tank  whole-cargo-tank)
      (geometry     object4903      )
      (capacity     7061.042        )
      (oxg          -28.52644       )
      (kg           8.319828        )) )
```

（d） ある貨物倉のインスタンス

図3･6 設計実体のクラスとインスタンスの記述例

```
(instance (name froude-number fn)
  (class dna)
  (iv (procedure ( r-vs / sqrt ( 9.806 * l ) ) )) )
```

（a） フルード数のインスタンス

```
(instance (name approx-block-coefficient a-cb)
  (class sna)
  (iv (procedure
        (( fn < 0.225 ) -->
            ( -11.134 * fn ^ 2 + 2.3290 * fn + 0.74719 )
         ( fn > 0.225 ) -->
            (  10.856 * fn ^ 2 - 7.5909 * fn + 1.8590  )) )
      (input-sentence "HOW MUCH IS BLOCK COEFFICIENT?" )  ) )
```

（b） 概略肥せき係数のインスタンス

```
(instance (name type-ship)
  (class snna)
  (iv (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 ) ) )) )
```

（c） 類似船のインスタンス

```
(instance (name prismatic-coefficient cp)
  (class dna)
  (iv (procedure ( $cp cp-curve ) )) )
```

（d） 柱形係数のインスタンス

図3･7 設計パラメータのインスタンスの記述例

である容積（capacity）や重心位置（oxg, kg）の決定方法をインスタンス変数（iv）の中に保持している．(b)は，(a)のサブクラスであり，各種の区画の中でも貨物倉とトップサイドタンクに共通する属性の決定方法を保持している．(c)は，貨物倉の実体に対するクラスの記述であり，(b)のサブクラスで，貨物倉の実体を生成するための手続きをクラス変数（cv）に保持している．(d)は，(c)のインスタンスとして設計過程で生成された実体を表現するインスタンスであり，(b)の記述に従って生成された形状オブジェクト（3･6節で後述）の名称や(a)の記述に従って計算されたマスプロパティの値などを保持している．一方，図3･7は設計パラメータの記述の船舶の基本設計における例であり，各インスタンスの変数 'procedure' にそれぞれの値の決定方法が記

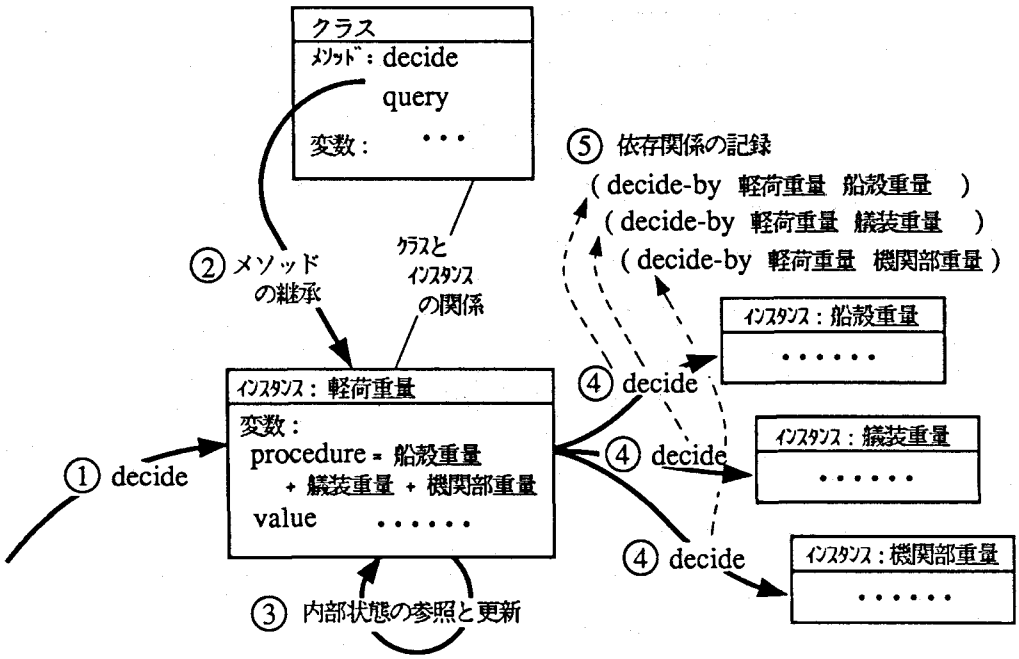


図3･8　オブジェクト間の継承と処理の連鎖

述されている．(a)はフルード数の記述で，' v_s / $\sqrt{9.806 * L}$ ' という計算式を保持している．(b)は概略肥せき係数の記述で，条件式が記述されている．また，(c)は類似船を検索するための記述，(d)は FORTRAN プログラムを呼び出して柱形係数を算出するための記述である．

3･3･3　設計項目間のネットワーク処理

次に，上述した設計対象の表現に基づいて行われる設計処理の中でも，メソッド 'decide' と 'delete' により行われる処理の内容を示す．

<u>設計項目の決定処理</u>　　各設計項目の内容の決定は，上記の 'procedure' を参照しながら，メソッド 'decide' によって行われる．図3･8は，その処理の過程を具体的に示したものであり，ある設計パラメータ '軽荷重量' の決定に関するものである．この処理は，まず，'軽荷重量' に対応するインスタンスに対して，決定処理を行うメソッド 'decide' を起動するためのメッセージを送信すること(①)により行われる．このメソッドの処理内容は，設計パラメー

タのインスタンスに共通する概念を定めたクラスから継承されており(②)．メソッド 'decide' が実行されると，まず，その値を保持する変数 'value' が参照され(③)，その値が既に決定されていれば，直ちにそれがメッセージの送信元に返却される．決定されていなかった場合には，変数 'procedure' の内容が参照され，'軽荷重量' の値を決定するために必要となる設計項目（ '船殻重量', '艤装重量', '機関部重量' ）に対して，同様のメソッド 'decide' を起動するメッセージが送信されて(④)，その戻り値をもとにして，'軽荷重量' の値が決定される．このとき，自動的に設計項目間の依存関係が自動的に述語として記録される(⑤)．このような関係は，次に示す再設計時における依存関係の処理や，3･4節と3･5節で示す設計支援のためなどに用いられる．なお，このような述語は非構造的な知識を表現するために用いる知識表現の方法であり（いわゆる述語論理の意味ではない)，二つのインスタンスの間の依存関係を述語として表現することにより，双方向からこのような記述に対してアクセスすることができるようになる．

設計項目間の依存関係の処理　設計における再設計の過程においては，修正しようとする設計項目の変更に伴って，その内容に依存して決定された他の設計項目の内容を再計算する必要がある．このような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などがある[11]．ここでは最後の方法を用いる．すなわち，メソッド 'delete' の処理により，図3･8のようにして設計処理の中で自動的に記録されている設計項目間の依存関係をもとにして，以下の手順で二次的修正を行うようにする．まず，一次的修正を行うために，ある設計項目に対してメソッド 'delete' を起動するためのメッセージが送信されると，その設計項目の内容を消去する．それとともに，メソッドによる処理の中から，設計項目の内容に依存して決定された他の設計項目の名称をあらかじめ記録されいる述語をもとにして参照して，二次的修正を行うために，それらに対応するオブジェクトに対して同様のメソッド

```
(rule21-1 : if (&ep dw < r-dw )
            then (&create-predicate ( too-small dw r-dw )) )

(rule21-2 : if (&ep dw > 1.05 * r-dw )
            then (&create-predicate
                        ( too-large dw ( 1.05 * r-dw ) )) )


(rule32-1 : if (&ep height-of-double-bottom <  b / 16.0 )
            then (&create-predicate
                  ( too-small height-of-double-bottom
                                          ( b / 16.0 ) )) )
```

図3･9　評価診断知識のルール表現

'delete' を起動するメッセージを送信する．さらに，このようなメッセージの送信を再帰的に行い，メソッド 'delete' の処理により，旧データを消去しつつ，必要なすべての二次的修正を自動的に行うようにする．なお，このようなメソッド 'delete' を起動するメッセージの送信は，メソッド 'decide' を起動したメッセージ送信の向きとは逆向きに行われる．

3･3･4　評価診断知識のルール表現とその処理

以上のような処理により操作される設計解は，設計過程において，設計条件と設計結果との比較，経験的知識，法規の適用などによる評価検討が加えられ，リファインされて満足解に近づけられる．本システムでは，これらの知識を「if ―― ，then ―― 」形式のルールとして表現した上で，求められた各設計項目の内容を各ルールの記述と照合して問題点があれば，それを意味する述語を作成するようにする．このようなルールの記述例を図3･9に示す．これも船舶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」などの知識が表現されている．このような記述に対して，ルールの記述中に各設計項目のオブジェクト名や属性名を記述するだけで，メソッド 'query' によりその値が参照され，ルールにより評価が行われて，診断結果の述語が英文に変換されて表示される．このよ

うな機能により，ユーザによる設計結果の判断に対して，助言を与えることができる．

3･3･5　設計プロセスの管理

本システムにおける設計プロセスは，前出の図3･3のようにして進行するものとするが，以下にそのための記述や機能について示す．

設計プロセス管理のための述語型知識　3･2･2項で述べた設計のプロセスの理解に対して，設計プロセスの進行を管理し，設計結果を表示して設計者の判断を方向付けるために，各設計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ようにする．これを，以下の形式の述語により行う．

(step 〈N〉 〈設計項目の名称〉)：〈N〉番目の段階で〈設計項目〉の内容を決定する．

(display 〈N〉 〈設計項目の名称〉)：〈N〉番目の段階で〈設計項目〉の内容を表示する．

(graphic-display 〈N〉 〈図形表示を行うための FORTRAN や C言語のプログラムを呼び出す記述〉)：〈N〉番目の段階で〈記述〉に従って設計結果を表現した図形を表示する．

これらの述語の記述例を図3･10に示す．これも船舶の基本設計に対するものの一部であり，後出の図3･22に対応する．このような表現により，まず，主要目が決定され，次に，基本的な性能の推定が行われ，さらに，主機関の選定が行われ，それぞれ結果が表示される．

設計プロセスの進行の管理　上記のような設計プロセスの記述に対して，図3･3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される．図3･3の各フェーズにおける具体的な処理を以下に示す．

(1) 初期化：ファイルとして記述された個々の設計問題に関する設計知識をロードするなどの前処理を行う．

(2) 設計とシミュレーション：上記の述語 'step' を参照し，メソッド 'decide' を用いて，現段階で決定すべき設計項目の内容を決定する．

```
(step 0 length)
(step 0 breadth)
(step 0 depth)
(step 0 full-load-draft)
(step 0 approx-block-coefficient)
(step 0 a-disp-f)

(step 1 approx-dead-weight)
(step 1 needed-vg-on-gren-full-load)
(step 1 a-vg-with-tst)
(step 1 approx-gm-on-full-load)
(step 1 approx-gm-on-ballast)
(step 1 p-value-on-full-load)
(step 1 p-value-on-ballast)

(step 2 (kind of main-engine))

(step 3 cp-curve)
(step 3 body-plan)

(step 4 requierd-ps)
(step 4 r-ps-with-sea-margin)

(step 5 (geometry of ship-body))
(step 5 (geometry of under-ship-body-surface))
```

図3･10　設計プロセス管理のための述語型知識

(3) 設計結果の表示と評価診断：上記の述語 'display','graphic-display' により現段階における設計状況を対比的に表示し，さらに，評価診断知識により設計結果を評価診断してその結果を表示する．

(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，(2) あるいは (5) のフェーズへ分岐する．また，すべての設計が終了した場合には，(6) のフェーズへ分岐する．

(5) 修正要目の入力と修正（再設計）：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' により修正を行う．また，必要に応じて，3･4節や3･5節で後述する機能によりこのような過程を支援するようにする．

(6) 終了：設計結果が満足と判断されたとき，設計を終了する．

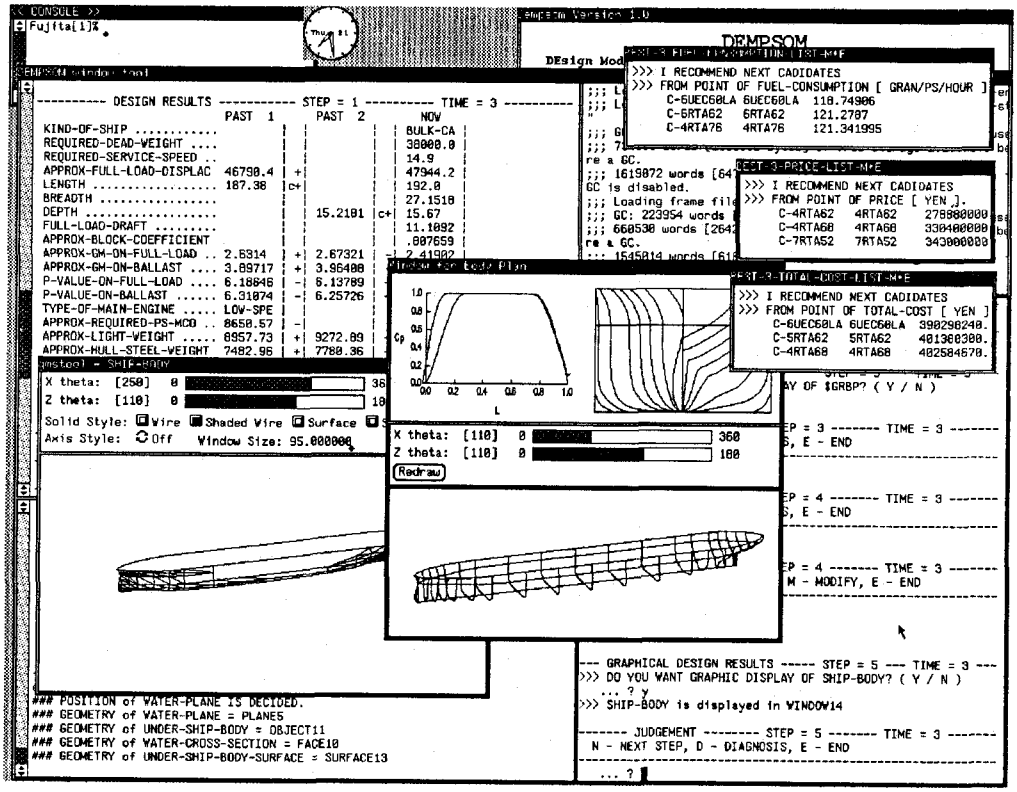


図3･11　システムの表示画面の一例

3･3･6　ユーザインターフェース

本システムに限らず設計支援システムには，試行錯誤に代表されるような設計者との相互作用が不可欠であるため，ユーザインターフェースの機能が重要となる．特に本手法の場合，設計解に対する総合的な判断が設計者に要求されるため，設計の進行状況や種々の修正案を多角的に表示することが重要である．以下に，本システムにおけるいくつかのインターフェース機能について述べる．前項で示した設計プロセスにおける設計のフェーズでは，メソッド 'decide' の処理の中から，メソッド 'display' が自動的に起動されることにより，設計の進行状況が順次表示される．また，設計結果の表示のフェーズでは，述語 'graphic-display' により設計結果を表した図形が表示され，述語 'display' により各設計項目の内容が，過去の履歴とともに対比的に表示される．さらに，

ワークステーション上のウインドウシステムの機能を利用して，様々な出力をユーザとの対話・設計進行状況のトレース・設計結果の表示のそれぞれのウインドウに分けて出力する機能や，設計結果を表した各種の図や表をそれぞれ別のウインドウに表示する機能などをシステムに付加する．図3･11は後述の事例における表示画面の一例である．

以上が，本システムにおける基本的な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．以下では，設計案の生成過程に対する支援機能や形状処理との融合について示す．

3･4　感度解析による設計支援機能

3･3節で示したシステムにより，機能設計における設計対象の様々な操作の過程を支援することができる．しかし，その中にあって，設計解に対してどのような修正を加えていくかは，設計シンセシスの本質に係わる部分であり，何らかの支援が必要である．本節では，そのような過程に対する支援機能として，数式微分を用いた感度解析による機能について示す．

3･4･1　感度解析による支援のねらい

設計解の修正過程は，修正すべき設計項目を選択し，その項目の修正値を設定しては，再設計を行う過程であり，効率的な設計を行うためには，前述のように有効な修正を行っていくことが重要である．しかし，このような操作においては，様々な設計項目が複雑に入り組んで関係しあっているため，設計項目間の依存関係や数量的な関係を設計者の能力のみで把握することは困難である．そこで，3･2節で示したネットワークモデルに基づいて，そのような設計項目間のネットワークや数量的な依存関係を視覚的に提示して，設計者を支援するようにする．すなわち，

(1) 修正すべき設計項目の選択に対して，設計項目間の依存関係を表現した

ネットワークをビジブルに表示して，設計者を支援する．

(2) 修正しようとする設計項目の修正値の設定に対して，数式微分を用いた感度解析により適切な設計項目の値を提示し，それによる設計解の変化を予測することにより，設計者を支援する．

以上により，設計過程を有効に支援することができる．このような支援の利点は，設計項目間のつながりや量的な依存関係を明示的に表現できる点にあり，設計者に対する有効な支援となる．以下に，その具体的な機能の内容を示す．

3･4･2　設計項目間の依存関係の表示

まず，設計項目間の依存関係を表示する機能について示す．表示すべき関係は，各設計項目に対応する節点と依存関係に対応して依存される方から依存する方への向きを持つアローとから構成される有向グラフであり，それを構成する個々の関係は，3･3･3項で述べたように，

（ decide-by　設計項目Ａ　設計項目Ｂ ）

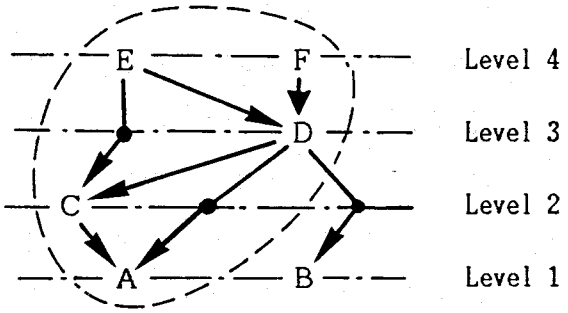
という形式の述語として，設計の過程で自動的に記録されている．なお，この例は「設計項目Ａの値が設計項目Ｂの値に依存して決定された」ことを意味している．しかし，このような依存関係は設計の状況に応じて数多くの関係が存在するため，すべての依存関係に関するネットワークを表示することは適切ではなく，再設計の目的に応じて，注目すべき設計項目を指定して，それに関係するネットワークの部分を抽出して，必要な情報を表示する必要がある．そのために，以下の手順で，それらの依存関係を整理した上で，自動的に視覚的な表示を行うようにする．

(1) 依存関係の抽出：現在の設計解における依存関係の中から，注目したい設計項目とそれに関連する設計項目間の依存関係を抽出する．

(2) 依存関係のレベル分け：抽出された依存関係に含まれる設計項目を，その関係に従って階層に分ける．

(3) グラフ表示のための前処理：設計項目間の依存関係やレベル分けのみで

```
( decide-by  A  C )
( decide-by  A  D )
( decide-by  B  D )
( decide-by  C  D )
( decide-by  C  E )
( decide-by  D  E )
( decide-by  D  F )
```



（a） 依存関係を表現した述語　　（b） (a)に対応するネットワーク

```
( decide-by  A  C )
( decide-by  A  D )
( decide-by  C  D )
( decide-by  C  E )
( decide-by  D  E )
( decide-by  D  F )
```

（c） 抽出された述語

```
( level  A  1 )
( level  C  2 )
( level  D  3 )
( level  E  4 )
( level  F  4 )
```

（d） レベル分けを表現した述語

図3･12　ネットワーク表示処理

はグラフ表示を行うことは困難であり，これらのデータをわかりやすく表示できるように加工する．

(4) グラフ表示：加工されたデータをもとにしてグラフの表示を行う．

以上の処理過程を図3･12のような簡単な例をもとにして以下に示す．いま，仮に，(a)のような依存関係があったとする．これには，(b)のようなネットワークが対応する．設計項目Aに注目した場合，A に関連する関係のみが再帰的なパターンマッチングによって取り出される．つまり，(a)の述語の中から A とは関係がなく B にのみ関連する述語 '(decide-by B D)' は除かれ，(c)の述語が抽出され，(b)の破線で囲まれた部分のネットワークに焦点が当てられる．続いて，抽出された関係に対して，パターンマッチングにより「その設計項目に依存して決定された設計項目が存在するかどうか」の判断を繰り返すことによりレベル分けを行うと，(d)の述語が生成されて，(b)に示したような各要素のレベルが把握できる．続いて，グラフ表示のための前処理が行

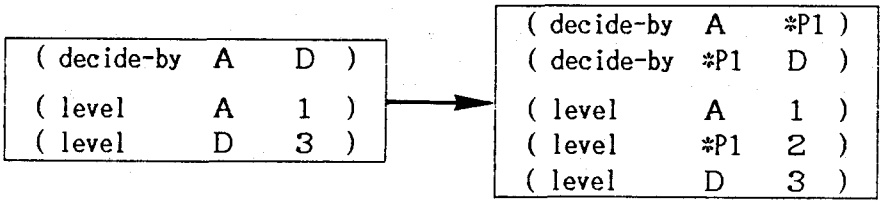


図3･13　ネットワーク表示のためのデータ加工

われる．グラフを表示するためには，各設計項目に対応する節点を画面上でどのように配置し，依存関係に対応するアローをどのように引くかを決定する必要がある．ここでは，あるレベルの節点は画面上で横一列に配置するものとし，また，離れたレベル間の関係については，両節点の間の各レベルに中継点（図3･12(b)中の黒丸）を設けて，それらの上を通ってアローを引くことにする．このような中継点の設定については，例えば，図3･12(b)における A と D のような関係を図3･13のように置き換えてアローを引くための中継点 '*P1' を設けることにより対処する．そのほか，各レベルに含まれる節点を横一列でどのような順番で並べるかを決定すれば，各節点やアローの位置が定まり，必要な依存関係を表示することができる．なお，このような機能の適用例については，3･4･4項で後述する．

3･4･3　数式微分を用いた設計案の生成支援

設計案の生成過程において，前項で示したようにして設計項目間の依存関係が明らかになり，修正すべき設計項目が明らかになると，次に，その値をどのように設定するかを決定しなければならない．このとき，「そのような設計項目の設定が他の項目にどのように影響するか」あるいは，「ある評価項目をある値にするためには，その設計項目をどのような値に設定すれば良いか」がわかれば，設計案の生成を支援することができる．

ある設計項目の値の変更による他の設計項目の値の変化は，それぞれの設計手続きや現在の値に依存する．つまり，値を変更しようとする設計項目を x，ある評価項目を z，z の設計手続き中に含まれる他の設計項目を y_1，y_2，$\cdots$，y_m としたとき，

$$\frac{dz}{dx} = \sum_{i=1}^{m} \frac{\partial z}{\partial y_i} \frac{dy_i}{dx} \quad \cdots\cdots(3\cdot1)$$

の関係を用いると，x を x_0 から x_n に変更したときの z の変化量は，$dz/dx(x_n-x_0)$ として感度解析的に得られる．一方，z を z_0 からの z_n とするような x_n も，この関係より，$x_0+(z_n-z_0)/(dz/dx)$ により求めることができる．以上の処理において必要となる各設計項目の値や設計手続きは，3･3･2項で示したように，設計対象を表現したオブジェクトの変数 'procedure' などに保持されており，自由に参照できる．また，式(3･1)中で必要となる各設計項目の偏微分値も，各設計項目の設計手続き（設計式）を参照し，それに含まれる他の設計項目を抽出して，それぞれによる偏微分式を数式微分により求め，それを評価することにより得られる．なお，数式微分はパターンマッチングを用いた Lisp 関数によるプログラムにより行うことができる[12]．一方，このような微分処理の手順は，前項で示したレベル分けに従って定まる．以上のようにして得られる新しい設計解の予測値も，前項で示したネットワーク上で表示して，設計案の生成を支援することにする．

3･4･4 感度解析による設計支援の一例

本支援機能の適用例として，後述の船舶の基本設計過程における設計案の生成支援に適用した例を示す．図3･14は，あるばら積み貨物船の基本設計において，基本性能の推定値をもとに主要目を決定している部分の実行例で，ある設計結果を評価診断し，それをもとに新しい設計案を生成して，再設計を行ったものである．この基本設計の初期段階は，設計手続きが比較的簡単な算術式で

記述されており，本手法を適用することができる．以下，図中の番号に従って説明する．

(1)：設計条件に対して得られた船の要目（初期値）が表示される．設計条件は載貨重量が 38000 t，航海速力が 14.9 knot であり，それに対して決定された主要目は長さが 187.38 m，幅が 27.15 m，… であり，続いて各評価項目が表示されている．これに対して評価診断を行うと，「載貨重量が不足する」ことなどが表示される．

(2)：(1)の結果に対して，ユーザが載貨重量を修正するために，これに関係する設計項目間のネットワークの表示を求め，そのネットワークが表示される．

(3)：(1)の診断結果や(2)のネットワークの表示をもとに，設計者が修正すべき設計項目として船の長さを指定する．

(4)：(3)に対して，長さに依存して決定された設計項目と，その値を 1 m 増加させた場合における他のパラメータの増減量が表示される．

(5),(6)：続いて，載貨重量を設計条件を満足する 38600 t にするために必要な長さの設定値が計算され，長さをそのように設定したときの各設計項目の値がネットワーク上に表示される．

(7)：長さを(6)に従って 191.59 m とすると，再設計が行われ，新しい載貨重量の値が決定され，その結果が前回の結果と対比されて表示される．表中には，長さを増加させたこと (C+)，載貨重量が増加したこと (+) などが示されている．また，これに対して，(1)と同様に評価診断を行うと，上記の載貨重量に関する問題点が解消されたことが確認できる．

以上のように，ネットワークのビジブルな表示や数式微分を利用した適切な設計案の提示機能により，設計解の生成過程を有効に支援することができる．また，これにより設計解の背後にある設計問題の構造を把握することもできるようになる．

(1)：与えられた設計条件に対する初期設計解とその評価診断の結果．

```
---------- DESIGN RESULTS ---------- STEP = 1 ---------- TIME = 1 ----------
 KIND-OF-SHIP ............. BULK-C |
 REQUIRED-DEAD-WEIGHT .... 38000. |
 REQUIRED-SERVICE-SPEED .. 14.9   |
 APPROX-FULL-LOAD-DISPLAC  46790. |
 LENGTH ................... 187.38 |
 BREADTH .................. 27.152 |
 DEPTH .................... 15.218 |
 FULL-LOAD-DRAFT ......... 11.109 |
 APPROX-BLOCK-COEFFICIENT  .80766 |
 APPROX-GM-ON-FULL-LOAD .. 2.6314 |
 APPROX-GM-ON-BALLAST .... 3.8972 |
 P-VALUE-ON-FULL-LOAD .... 6.1865 |
 P-VALUE-ON-BALLAST ...... 6.3107 |
 TYPE-OF-MAIN-ENGINE ..... LOW-SP |
 APPROX-REQUIRED-PS-MCO .. 8650.6 |
 APPROX-LIGHT-WEIGHT ..... 8957.7 |
 APPROX-HULL-STEEL-WEIGHT  7483.0 |
 APPROX-OUTFIT-WEIGHT .... 864.91 |
 APPROX-MACHINERY-PART-WE  609.87 |
 APPROX-DEAD-WEIGHT ...... 37833. |
 NEEDED-VG-ON-GREN-FULL-L  53200. |
 APPROX-VG-WITH-TOP-SIDE-  50713. |
-------------------------------------------------------------------------------

---------------------------------- DIAGNOSIS ----------------------------------
 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.
-------------------------------------------------------------------------------
```

(2)：載貨重量が依存する設計項目．

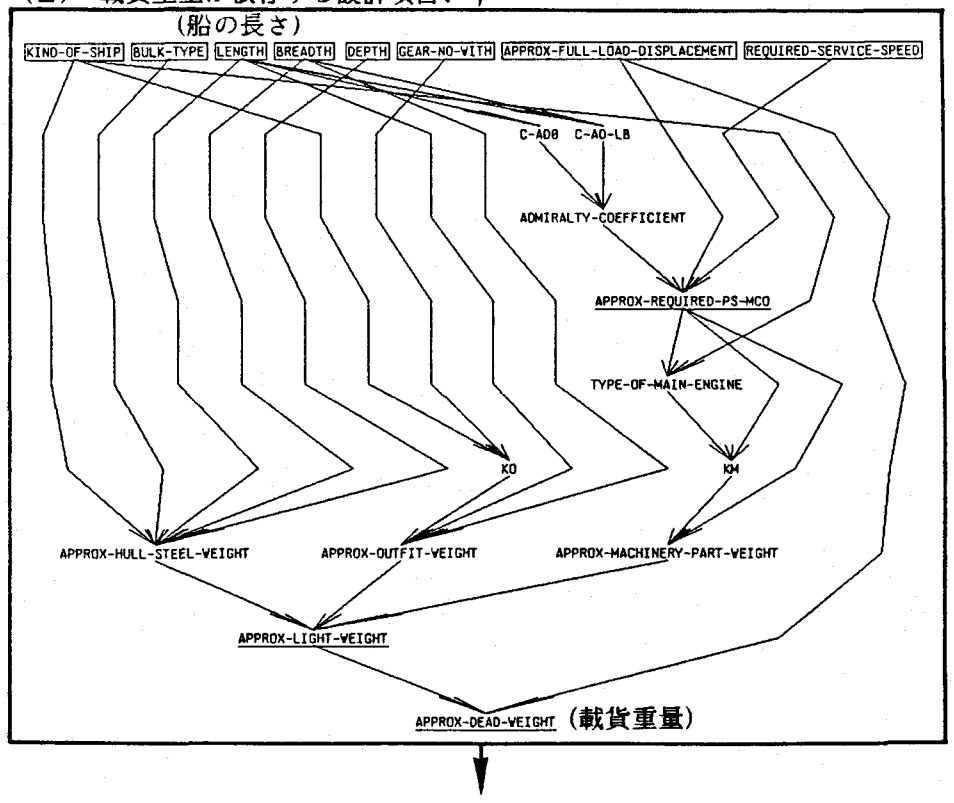
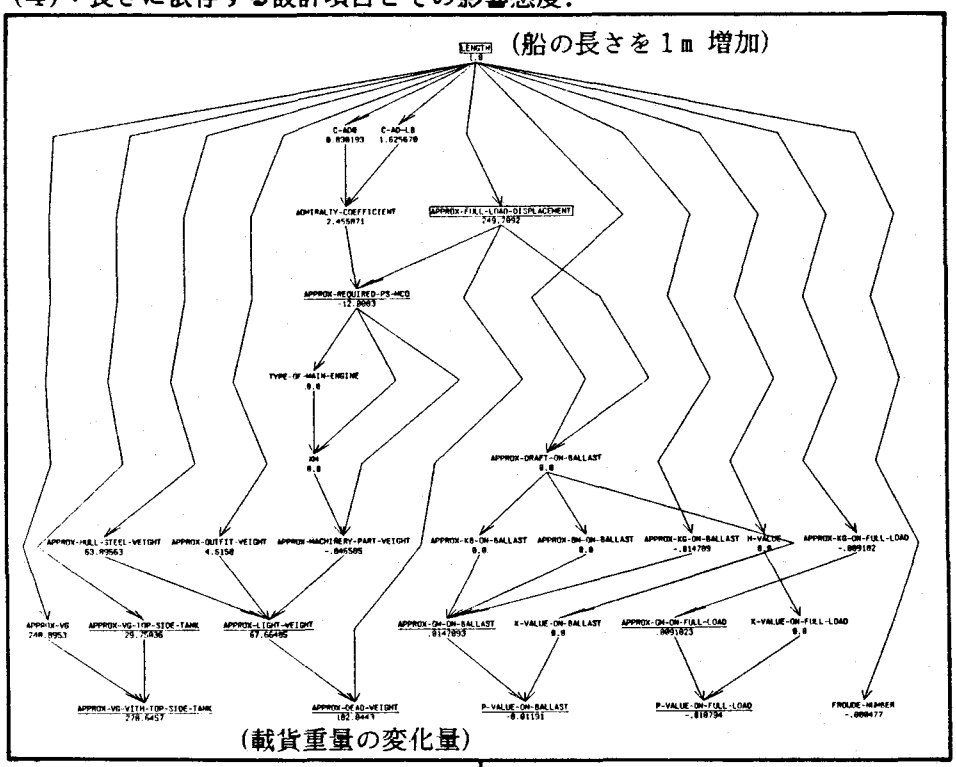


図3･14　感度解析を融合した設計支援の実行例（1）

(3)：修正する設計項目の指定.

```
------------------- SELECT DESIGN VARIABLES TO BE CORRECT -------------------
>>> PLEASE INPUT VARIABLE NAME TO BE CORRECTED ( E - END )
   ... ? LENGTH
>>> SELECT VARIABLE DECIDED BY OTHER PRINCIPAL PARTICULARS.
>>> ( 1 - APPROX-FULL-LOAD-DISPLACEMENT, 2 - FULL-LOAD-DRAFT,
      3 - APPROX-BLOCK-COEFFICIENT, 4 - BREADTH, )
   ... ? 1
```

(4)：長さに依存する設計項目とその影響感度.

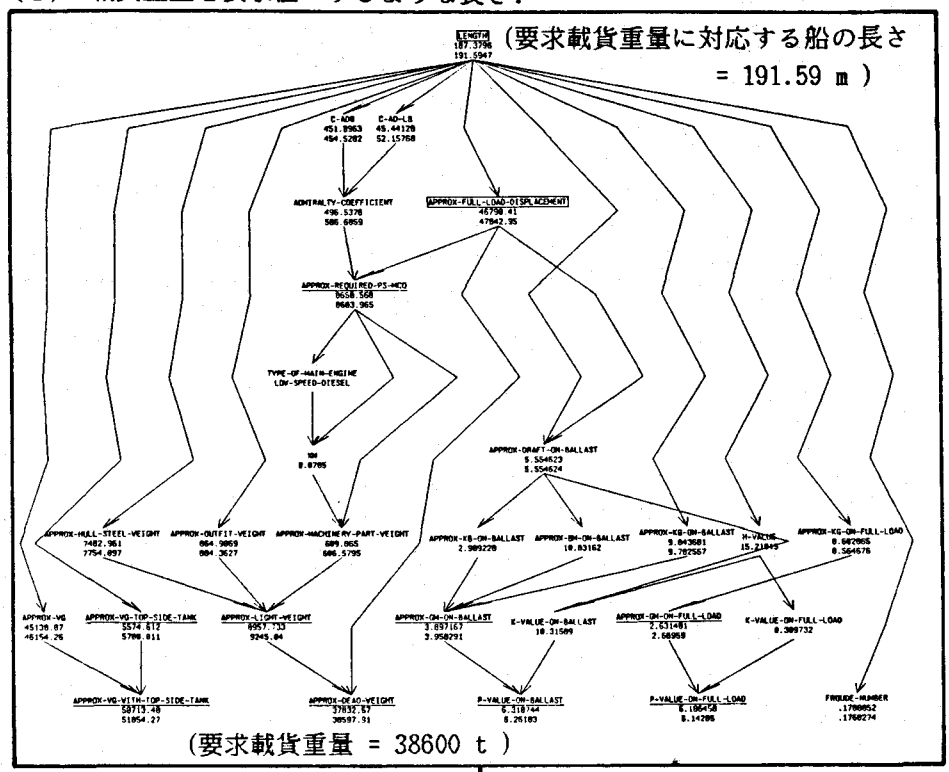


(5)：載貨重量の要求値の設定.

```
>>> WHICH PARAMETER DO YOU WANT TO SET UP TO THE SUITABLE VALUE?
>>> ( 1 - APPROX-VG-WITH-TOP-SIDE-TANK, 2 - APPROX-DEAD-WEIGHT,
      3 - P-VALUE-ON-BALLAST, 4 - P-VALUE-ON-FULL-LOAD,
      5 - APPROX-VG-TOP-SIDE-TANK, 6 - APPROX-LIGHT-WEIGHT,
      7 - APPROX-GM-ON-BALLAST, 8 - APPROX-GM-ON-FULL-LOAD,
      9 - APPROX-REQUIRED-PS-MCO, )
   ... ? 2
>>> HOW VALUE DO YOU WANT TO SET UP APPROX-DEAD-WEIGHT?
>>>     IT'S NOW VALUE IS 37832.67
   ... ? 38600
```

図3･14　感度解析を融合した設計支援の実行例（2）

(6)：載貨重量を要求値にするような長さ.



(7)：再設計の結果と得られた設計解に対する評価診断の結果.

```
---------- DESIGN RESULTS ----------- STEP = 1 ---------- TIME = 2 ----------
                          PAST 1        NOEW
 KIND-OF-SHIP .............          |   | BULK-C |
 REQUIRED-DEAD-WEIGHT ....          |   | 38000. |
 REQUIRED-SERVICE-SPEED ..          |   | 14.9   |
 APPROX-FULL-LOAD-DISPLAC  46790.   | +| 47843. |
 LENGTH ................... 187.38 |C+| 191.59 |
 BREADTH .................          |   | 27.152 |
 DEPTH ....................         |   | 15.218 |
 FULL-LOAD-DRAFT .........          |   | 11.109 |
 APPROX-BLOCK-COEFFICIENT           |   | .80766 |
 APPROX-GM-ON-FULL-LOAD .. 2.6314   | +| 2.6696 |
 APPROX-GM-ON-BALLAST .... 3.8972   | +| 3.9583 |
 P-VALUE-ON-FULL-LOAD .... 6.1865   | -| 6.142  |
 P-VALUE-ON-BALLAST ...... 6.3107   | -| 6.2618 |
 TYPE-OF-MAIN-ENGINE .....          |   | LOW-SP |
 APPROX-REQUIRED-PS-MCO .. 8650.6   | -| 8604.0 |
 APPROX-LIGHT-WEIGHT ..... 8957.7   | +| 9245.0 |
 APPROX-HULL-STEEL-WEIGHT  7483.0   | +| 7754.1 |
 APPROX-OUTFIT-WEIGHT .... 864.91   | +| 884.36 |
 APPROX-MACHINERY-PART-WE  609.87   | -| 606.58 |
 APPROX-DEAD-WEIGHT ...... 37833.   | +| 38598. |
 NEEDED-VG-ON-GREN-FULL-L           |   | 53200. |
 APPROX-VG-WITH-TOP-SIDE-  50713.   | +| 51854. |
-------------------------------------------------------------------------------

---------------------------------- DIAGNOSIS ----------------------------------
 VG-WITH-TOP-SIDE-TANK WILL BE TOO SMALL AS COMPARED WITH
    NEEDED-VG-ON-GREN-FULL-LOAD.
-------------------------------------------------------------------------------
```

図3･14　感度解析を融合した設計支援の実行例（3）

3・5　最適化手法との融合による設計支援機能

前項で示した機能により設計案の修正過程を有効に支援することができるが，数式微分による提示機能は修正しようとする設計項目が一つの場合に限られるため，その適用の範囲は限られたものとなる．本節では，より総合的な支援機能として最適化手法を融合した手法を提案し，その内容について示す．

3・5・1　最適化手法との融合のねらい

最適化手法は設計シンセシスにおける有効な手段の一つではあるが，3・2・1項でも述べたように，その適用には設計目標間の総合化などの点で，様々な困難が伴う．そこで，本機能においては，それらの点を踏まえた上で，数理計画法を適用するための定式化を対話的かつ柔軟に行い，これによって有効に最適化手法を適用して合理的な設計支援を実行する．すなわち，機能設計における設計目標間の総合化を支援するために，次のような立場から最適化手法を融合する．

(1) 機能設計は，あくまでも「設計・評価・再設計」の過程を繰り返しながら，設計解をリファインしていくことにより行われるものとして，そのリファインの操作を支援する立場から最適化手法を融合する．

(2) (1)の立場から，過去の設計事例における実績を積極的に参照したり，それを用いたリファインを重視し，設計者の判断に基づいて設計解を逐次的に改善していくようにする．

(3) 数理計画法を適用するための定式化における制約条件や目的関数は，設計を行う以前に定まっているものではなく，得られた設計解の問題点に対応して，その都度，上記の繰返しの中で徐々に明らかにされていくものであると考え，最適化手法を適用する過程で，設計者の判断に従って定式化に柔軟な変更が加えられるようにする．

このような支援手法の考え方は，既存の設計事例は一般に目標間のバランスの点では優れており，設計者が陽に意識しない様々な要求をあらかじめ満足して

いること，設計者にとって具体的な設計解からその解のもつ問題点を把握することは容易であること，さらに，複数の目標に対する総合的な判断は自動的なアルゴリズムに表現することは困難であり，むしろ人間が得意とする処理の一つであって人間を介入されたほうが望ましいことなどの点に基づくものである．

3･5･2　最適化手法との融合による支援の過程

上記の基本概念に基づいて設計支援手法を実現し，設計解を逐次的に改善していくために，まず，設計過程を以下の二つの段階に分けて考える．

(1) 満足化の過程：設計条件などの必要最低限の機能を満たす設計可能解（満足解）を得る過程．

(2) 最適化の過程：(1)で得られた設計解に対してさらに修正を加え，目標間のバランスをはかりながら，より高い機能を有する設計解を得る過程．

図3･15は，本機能の概念とその適用による支援過程を具体的に示した図である．図中の ▒▒ 部は，既知の明確な設計条件を満足する領域（許容領域）であり，少なくとも，この中の解を得る必要がある．これが上記の満足化の過程に対応する．また，二つ（複数）の等高線は競合する設計目標（Obj-1，Obj-2）であり，許容領域内において総合的な観点から望ましい設計解を得る必要があり，これが最適化の過程に対応する．

さて，Ｉ点を設計要求をもとにして得られる初期設計解とする．このような解は一般に過去の設計事例を参考にするなどして得られるが，図のようにあらかじめ定められた許容領域内に初期解を得ることは困難である．そこで，まず，許容領域を制約条件として数理計画法を適用し，条件を満足する設計解（満足解 S_1点）を求めることにする．このとき，前項で示した考え方に従って，初期解の持つ優れた面を満足解に可能な限り反映させるように，初期解と満足解との距離を最小化するようにする．しかし，以上の過程において，すべての設計条件をあらかじめ定式化することは容易ではないため，場合によっては，得られえた解に不都合を生じる場合もある．そのような場合には，設計者の判断に基づいてインタラクティブに設計解に対する側面制約を加えるなどして，再

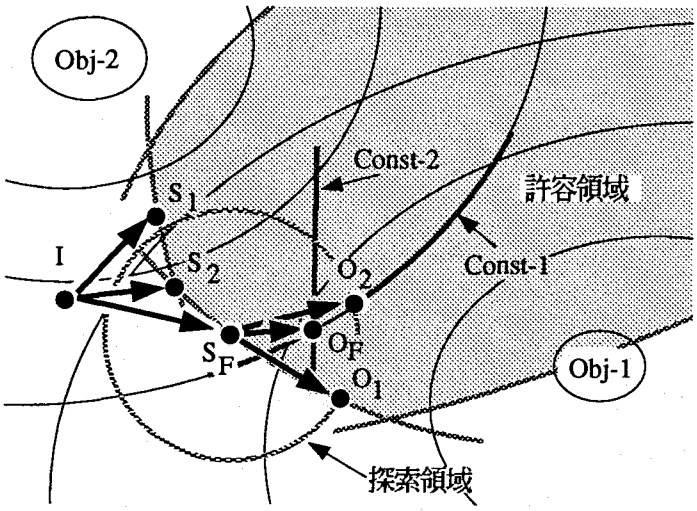


図3･15　最適化手法との融合の概念

び最適化手法を適用し新たな解（S_2点）を得るようにする．このような過程を数回繰り返すことにより適切な満足解（S_F点）が明らかとなる．

続いて，さらに解を改善するために最適化を行う．つまり，ある一つの設計目標に限った最適化計算を既に得られている満足解（S_F点）の周辺で行い，さらに，結果をみて必要があれば，前回の定式化に変更を加えたり，設計目標を入れ換えたりして，繰り返し数理計画法を適用することにより，最終的な設計解を得るようにする．ここで，既存解（Ｉ点およびS_F点）の優位性に留意するために，探索領域を限定するようにする．すなわち，図3･15におけるS_F点の周辺（図中の S_F点 を中心とする円の内部）に探索領域を限って，Obj-1に関する最適化を行うようにする．これによって得られた解（O_1点）も，対立する設計目標 Obj-2 に関する評価が悪化しそれが許容できなかったり，満足化の場合と同様，他の評価の点で問題となることがある．この場合も，設計者の判断により，新たな制約条件（Const-1, Const-2）を追加したり，探索領域の大きさを変更したりして，再び数理計画法を適用し，さらに，設計解を改善していくようにする（O_2点，O_3点，･･･）．このようにして，いくつかの具体的な解が得られるが，これらの結果を比較すれば設計者にとって設計目標間

のトレードオフや改善の限界を総合的に判断することは容易であり，目標間の総合化をはかった上で，最終的な解（O_F点）を得ることができるようになる．なお，以上の取扱いは一種の制約指向に相当する考え方に基づくものでもある．

以下では，このような支援を行うための具体的な方法について示す．

3･5･3 最適化手法との融合のための定式化

前項で示したような支援を行うためには，設計対象を数学的な形に定式化し，それに適した数理計画法を適用する必要がある．本手法においては，上述したような支援方法に従って，以下のような定式化を行う．

設計変数：設計者が指定する設計項目の値．

制約条件：各設計実体の属性や設計パラメータの決定手続きに対応する数学的な関係式（等式制約条件）や，設計解の評価診断知識から変換される条件（不等式制約条件）．設計者が恣意的に追加した制約条件．最適化の場合には，さらに探索領域に対応する後出の式(3･3)の不等式制約条件が加わる．

目的関数：満足化の場合には初期設計解と満足解との距離に対応する後出の式(3･2)，最適化の場合には設計者が指定する設計目標．

また，具体的な数理計画法としては，本機能における数理計画問題の探索の範囲が比較的狭く，また，Lisp 関数による数式微分を用いることにより対象のモデルを3･4･3項のような方法により容易に線形化することができるなどの理由から，制約条件と目的関数の逐次的な線形化近似により，繰り返し線形計画法（ＬＰ）を適用して最終的な最適解を求める逐次線形計画法（ＳＬＰ）[13]を用いることにする．以下に，満足化・最適化のそれぞれに対する具体的な目的関数や制約条件について示す．

<u>満足化における目的関数</u>　満足化の過程では，まず，初期解に近い許容領域内の解を求める．そのために，以下の初期解と新たな設計解との距離を最小化する目的関数を便宜的に導入する．

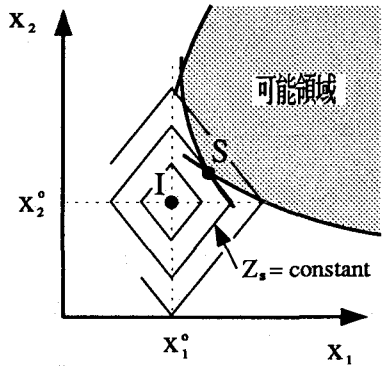


図3･16　満足化における目的関数

$$Z_s = \sum_{i=1}^{n} \frac{1}{w_i} \left| \frac{x_i - x_i^0}{x_i^0} \right| \rightarrow \min \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3\cdot2)$$

ここで，x_i は修正しようとする設計変数（ $i = 1, 2, \cdots, n$ ）であり，x_i^0 は x_i の初期設計解における値，w_i は x_i の修正幅に関する重み付けの係数である．設計変数の数が２個の場合における目的関数と制約条件の関係は図3･16のようになり，w_i により定まる距離空間において許容領域の中で最もＩ点に近いＳ点が解として得られる．なお，この w_i のもつ性質については後述する．

<u>最適化における探索領域</u>　　最適化の過程では，修正しようとする解の周辺に限って，ある一つの設計目標を最適とするような解を求める．このとき，設計変数の修正幅に制約を加えるために，以下の制約条件を便宜的に付加する．

$$\frac{1}{w_i} \left| \frac{x_i - x_i^*}{x_i^*} \right| \leq \alpha \quad (i = 1, 2, \cdots, n) \quad \cdots\cdots\cdots (3\cdot3)$$

ここで，x_i^* は x_i の満足解において得た値，α は各設計変数の変更幅の上下限を規定する定数である．なお，α の意味については後述する．

設計者との対話による定式化の変更　上記の定式化に加えて，総合化における設計者の意図を反映するために，以下のようにして，定式化を調整できるようにする．

(1) 制約条件の付加：3･5･2項で述べたように，定式化において必要な制約条件が欠如しているために，設計者が意図した解が求められない場合があり，そのような場合には解のもつ問題点に対応して制約条件を追加する必要がある．また，逆に過剰な制約条件を追加したために，解が得られなくなったりする場合もあり，それを削除することも必要である．そこで，設計者が設計解をみては，自由に制約条件を追加したり，削除したりできるようにする．

(2) 満足化における距離空間の変更：満足化における距離空間は，式(3･2)において w_i により規定される．この値の既定値は 1.0 とするが，設計者はこの値を小さくすることにより，対応する設計変数の値の変更を抑制することができる．

(3) 最適化における探索領域の変更：最適化における探索領域は，式(3･3)において α により設計変数に対して共通的に決定され，その値が大きいほど広くなる．また，個別の設計変数に対しては w_i により調整することができ，満足化の場合と連動して，その値を小さくするほど対応する変数の値の修正幅の上下限は小さくなる．

オブジェクトによる表現　以上の定式化を設計支援システムにおいて取り扱うにあたっては，上記の目的関数や制約条件なども，3･3･2項で示した設計対象の表現と同様，オブジェクトとして表現する．すなわち，目的関数や制約条件に対するクラスを定義して，ＳＬＰを適用するための線形化などの処理を行うメソッドを定義した上で，具体的な目的関数や制約条件はそれぞれ独立にそれらのインスタンスとして取り扱うようにする．このほか，設計対象のオブジェクトのクラスに対しても本機能を実現するためのメソッドを追加し，これらのメソッドの連携によって，最適化手法を融合するための各種の処理が行われるようにする．

3・5・4　最適化手法の適用手順

以上のような方法を具体的に適用して，再設計を支援する手順を以下に示す．まず，得られた設計解を評価診断した結果に基づいて，設計解に問題点がある場合には満足化のための定式化を行い，そうでない場合には最適化のための定式化を行う．前者の場合には，複数の問題点が存在するときにはそれらを一度に解消するか，あるいは逐次的に解消するかを設定できる．後者の場合には，どの設計目標に関する最適化を行うかを設定する．設計変数は，3・4・2項で示した機能により設計項目間の依存関係を把握した上で，対話的に選定できるようにする．続いて，選定された設計変数に対応して，上記の制約条件や目的関数が自動的に設定される．さらに，このような定式化に対し，必要に応じて3・5・3項で述べたような方法により変更を加えるようにする．以上のようにしてインタラクティブに定式化が定まると，次に，ＳＬＰによってその解を求める．なお，そのためには，各種の不等式制約条件を等式制約条件に変換したり，各関数を初期解の近傍で線形化したり，さらに，必要に応じて線形化を繰り返したりする必要がある．これらの処理はすべて上記のようなオブジェクトによる表現に基づいて自動的に行い，線形化は3・4・3項で示した方法により数式微分に基づいて行う．また，ＬＰ問題の求解は Lisp による処理の中から FORTRAN により記述されたプログラムを自動的に呼び出して行う．

3・5・5　最適化手法と融合による設計支援の一例

本支援機能の適用例として，前節と同様，後述の船舶の基本設計過程における設計案の生成支援に用いた例を示す．船舶の基本設計においては，載貨重量や貨物倉容積，安定性などが満足すべき必要機能であり，軽荷重量や所要推進動力などが最適化すべき項目に相当する．図3・17は満足化過程に対する実行例であり，ある設計結果を評価診断し，それをもとに新しい設計案を生成して満足化を行ったものであり，以下に図中の番号に従って説明する．

(1)：初期設計解に対する評価診断の結果であり，載貨重量（dead weight）と貨物倉容積（vg）が不足することが指摘されている．

(1)：初期設計解に対する評価診断の結果.

```
------------------------------ DIAGNOSIS ------------------------------
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.
-----------------------------------------------------------------------
```

(2)：満足化のための定式化.

```
-------------------- SUPPORT BY OPTIMIZATION TECHNIQUE --------------------
>>> BEFORE OPTIMIZATION, YOU MUST MODIFY THE DESIGN MODEL TO SATISFY
>>> FOLLOWING CONSTRAINTS ACCORDING TO DESIGN DIAGNOSIS.
  RULE41-2 : ( NEEDED-VG-ON-GREN-FULL-LOAD < APPROX-VG-WITH-TOP-SIDE-TANK )
  RULE21-3 : ( REQUIRED-DEAD-WEIGHT < APPROX-DEAD-WEIGHT )
>>> IN WHICH WAY DO YOU WANT TO SATISFIZE CONSTRAINTS?
>>> ( 1 - ONCE, 2 - SEQUENCIALY, )
   ... ? 1
```

(3)：(2)に対する満足化の結果.

```
------------------------ SATISFICATION RESULTS --------------------------
>>> I GET FOLLOWING NEW MODEL BY OPTIMIZATION TECHNIQUE.
>>> THIS SATISFIZES DIAGNOSIS RULES.
      BREADTH ................. 30.28803 --> 31.92521 : + :
      DEPTH ................... 17.85626 --> 18.54042 : + :
>>> EVALUATIVES ARE AS FOLLOWS ON THIS RESULTS.
      APPROX-VG-WITH-TOP-SIDE . 76754.88 --> 84003.62 : + : WITHIN
      APPROX-DEAD-WEIGHT ...... 59718.94 --> 62998.81 : + : WITHIN
      P-VALUE-ON-BALLAST ...... 6.847605 --> 6.391001 : - : WITHIN
      P-VALUE-ON-FULL-LOAD .... 6.545976 --> 6.4125   : - : WITHIN
      APPROX-VG-TOP-SIDE-TANK . 8437.178 --> 9233.985 : + : WITHIN
      APPROX-LIGHT-WEIGHT ..... 12624.57 --> 13255.14 : + : MINIMUM
      APPROX-GM-ON-BALLAST .... 3.799246 --> 4.650011 : + : WITHIN
      APPROX-GM-ON-FULL-LOAD .. 2.738722 --> 3.000869 : + : WITHIN
      APPROX-REQUIRED-PS-MCO .. 11087.01 --> 11839.36 : + : MINIMUM
>>> THE FOLLOWING CONSTRAINTS ARE ACTIVE ON THIS RESULTS.
   R : ( APPROX-DEAD-WEIGHT < 1.05 * REQUIRED-DEAD-WEIGHT )
   R : ( NEEDED-VG-ON-GREN-FULL-LOAD < APPROX-VG-WITH-TOP-SIDE-TANK )
```

図3･17　満足化過程に対する支援の実行例（1）

(2)：設計者が，修正すべき設計変数として，長さ・幅・深さを選定したことに基づいて，満足化のための定式化が自動的に行われる.

(3)：(2)の定式化に対して満足化が行われ，得られた解が提示される．このとき，各設計項目がどのように変化するかが，'-->' で示されるとともに，その解においてアクティブになっている制約条件が示される.

(4)：この解に対して，設計者が「幅が長過ぎる」と判断して，'BREADTH ≦

(4)：(3)の結果をもとにした定式化の変更（制約条件の追加）.

```
------------------- MODIFICATION OF CONSTRAINT & WEIGHTS ------------------
>>> PLEASE SELECT MODE ON MODIFICATION.
>>> ( 1 - CONSTRAINT, 2 - WEIGHT, 3 - RANGE, 4 - LIST-UP, 5 - END, )
   ... ? 1
>>> PLEASE SELECT MODE ON USER DEFINED CONSTRAINTS.
>>> ( 1 - APPEND, 2 - EXIT, 3 - EXIT-MODIFY, )
   ... ? 1
>>> PLEASE INPUT CONSTARINT IN LIST FORM.
   ... ? ( BREADTH < 31.0 )
>>> PLEASE SELECT MODE ON USER DEFINED CONSTRAINTS.
>>> ( 1 - APPEND, 2 - DELETE, 3 - EXIT, 4 - EXIT-MODIFY, )
   ... ? 4
```

(5)：(4)による定式化の変更のもとでの満足化の結果.

```
------------------------ SATISFICATION RESULTS --------------------------
>>> I GET FOLLOWING NEW MODEL BY OPTIMIZATION TECHNIQUE.
>>> THIS SATISFIZES DIAGNOSIS RULES.
      LENGTH ................. 216.6722 --> 224.2095 : + :
      BREADTH ................ 30.28803 --> 30.99969 : + :
      DEPTH .................. 17.85626 --> 18.45198 : + :
>>> EVALUATIVES ARE AS FOLLOWS ON THIS RESULTS.
      APPROX-VG-WITH-TOP-SIDE . 76754.88 --> 84003.17 : + : WITHIN
      APPROX-DEAD-WEIGHT ..... 59718.94 --> 62998.79 : + : WITHIN
      P-VALUE-ON-BALLAST ..... 6.847605 --> 6.758183 : - : WITHIN
      P-VALUE-ON-FULL-LOAD .... 6.545976 --> 6.663277 : + : WITHIN
      APPROX-VG-TOP-SIDE-TANK . 8437.178 --> 9233.937 : + : WITHIN
      APPROX-LIGHT-WEIGHT ..... 12624.57 --> 13620.25 : + : MINIMUM
      APPROX-GM-ON-BALLAST .... 3.799246 --> 4.084735 : + : WITHIN
      APPROX-GM-ON-FULL-LOAD .. 2.738722 --> 2.753791 : + : WITHIN
      APPROX-REQUIRED-PS-MCO .. 11087.01 --> 11353.67 : + : MINIMUM
>>> THE FOLLOWING CONSTRAINTS ARE ACTIVE ON THIS RESULTS.
   R : ( APPROX-DEAD-WEIGHT < 1.05 * REQUIRED-DEAD-WEIGHT )
   R : ( NEEDED-VG-ON-GREN-FULL-LOAD < APPROX-VG-WITH-TOP-SIDE-TANK )
   U : ( BREADTH < 31.0 )
```

(6)：再設計により得られた設計解に対する評価診断の結果.

```
------------------------------- DIAGNOSIS --------------------------------
  DESIGN RESULT HAS NO PROBLEM.
--------------------------------------------------------------------------
```

図3･17　満足化過程に対する支援の実行例（2）

31.0' なる制約条件を付加して，定式化に変更を加えている.

(5)：(4)の定式化に対して再び満足化が行われ，得られた解が提示される.

(6)：得られた結果に対して再設計を行い，その結果，満足化がはかられたことが確認できる

以上のようにして，満足化の過程を支援することができる.

表3･2　最適化過程に対する支援の実行結果

		Initial	Plan 1	Plan 2	Plan 3
設計変数	長さ	224.210	213.530	216.467	222.591
	幅	30.9997	30.9997	30.8734	30.9766
	深さ	18.4520	19.3744	19.1899	18.5998
設計目標	軽荷重量	13620.3	12896.6	13074.6	13505.8
その他の設計目標	載貨重量	62998.8	60072.9	60597.3	62503.8
	バラスト時のＰ値	6.75818	7.66638	7.54829	6.89974
	満載時のＰ値	6.66328	7.76851	7.58005	6.82483
	バラスト時のＧＭ	4.08474	3.35332	3.41621	3.95268
	満載時のＧＭ	2.75379	2.14707	2.22861	2.64952
	所要推進動力	11353.7	11439.9	11353.7	11353.7
探索領域の定義：α			0.10	0.04	0.04
付加された制約条件		B ≦ 31.0	B ≦ 31.0	B ≦ 31.0 PS ≦ 11354	B ≦ 31.0 PS ≦ 11354 D ≦ 18.6

だたし，'B' は幅を，'PS' は所要推進動力を，'D' は深さを表す．

表3･2は，続いて行った軽荷重量に関する最適化（最小化）によって生成された設計解を対比したものである．'Initial' は上記の満足化によって得られた解であり，'Plan 1' はこれに対して直ちに最適化を行った結果である．両者を比較すると，軽荷重量に競合する設計目標である所要推進動力が増加していることがわかる．そこで，探索領域を縮小し，所要推進動力に対する制約条件を追加して，再び最適化を行うことにする．その結果が 'Plan 2' である．しかし，その解においても船体の深さが大き過ぎるという問題点が残ったため，さらに制約条件を追加して最適化を繰り返す（Plan 3）．このような過程を通じて，設計者は徐々に望ましい設計解を得ることができる．

以上のように，最適化手法を融合した適切な設計解の提示機能により，設計解の生成過程に対して，インタラクティブかつ段階的に柔軟な支援を行うことができる．また，本機能は，知識ベースに表現された設計対象のオブジェクト

表現に基づいて自動的に定式化を行うため，設計者が煩わされることがないという特徴も有する．

3･6　形状処理との融合

3･6･1　設計シンセシスにおける形状処理

設計支援において設計対象の形状の取扱いは重要な要素であり，本システムが対象とする機能設計においても，設計対象のもつ機能を評価していく上で形状処理が不可欠となる問題も少なくない．設計における形状処理については，古くから研究がなされており，基本的な形状モデリングの手法として，CSG (constructive solid geometry) と B-Reps (boundary representation) の二つが定着し，また，自由曲面のモデリングについても多くの手法が提案されている[14]．しかし，従来の研究や試みの多くは，形状そのものの処理に重点が置かれており，設計本来のシンセシス過程を支援しようとはしていない．これに対して，最近では，知識情報処理の技術をもとに形状処理を行おうとする試みとして，フィーチュアモデル (feature model) を形状モデルの中心に据えた研究[15]も行われている．しかし，これらの研究の多くも，詳細設計や生産設計などの過程に対する支援を念頭においたものであり，機能設計に対するものは少ない．本システムでは，機能設計における設計処理を有効に支援しようとする立場から，前述したネットワークモデルに対して形状処理を融合する．すなわち，既に3･2節や3･3節で示した設計対象の表現において，形状を設計実体の属性のひとつとして取り扱うことにより，形状処理を伴った機能設計の過程を支援するようにする．

3･6･2　設計処理と形状処理の連携

図3･18は融合の基本的な形態を示したものであり，オブジェクト指向プログラミングのもとで構築した設計支援システムに対して，同じ環境の上でオブジェ

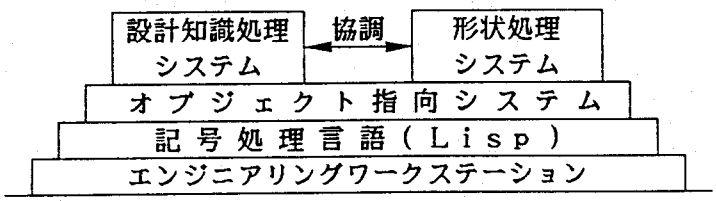


図3･18　形状処理との融合の基本概念

クト指向プログラミングによる形状処理システムを連携させる．具体的には，このような構成のもとで，前出の図3･4(a)に示したようにして，設計実体のインスタンスの属性のひとつとしてその形状を保持させ，図3･6に示したような各種の属性の決定手続きの記述に従って，形状に付随するマスプロパティの算出などの処理を実行するようにする．また，形状のモデリング手法としては，B-Reps を用いることにする．これは事例として取り上げる船舶の基本設計における形状の取扱いが B-Reps のほうが容易であるためであり，取り扱う対象によっては，CSG による方法が適している場合や，両者を連携させたほうが良い場合もある．なお，形状モデリングをオブジェクト指向のもとで実現しようとする試み[16]は少なくないが，その多くは機能設計を対象としていない．

3･6･3　オブジェクト指向による形状のモデリング

B-Reps による形状の表現[14]は，一般に，対象の形状を構成する立体・面・稜線・頂点などの間の連結関係をポインタを用いて定義することにより行われる．このような表現をオブジェクト指向のもとで実現する場合のデータ構造を図3･19に示す．まず，B-Reps における上記の形状表現要素のそれぞれをオブジェクト（インスタンス）として表現し，要素間の位相的な接続関係をオブジェクト間の関係 'composed-of' として表現することにより，B-Reps のデータ構造を表現する．なお，このようなオブジェクト間の「関係」は，個々のオブジェクトの内部状態ではなく，オブジェクト相互間の帰属関係や支配関係を記述するために導入した表現の方法である．さらに，このような表現のもとで，オブ

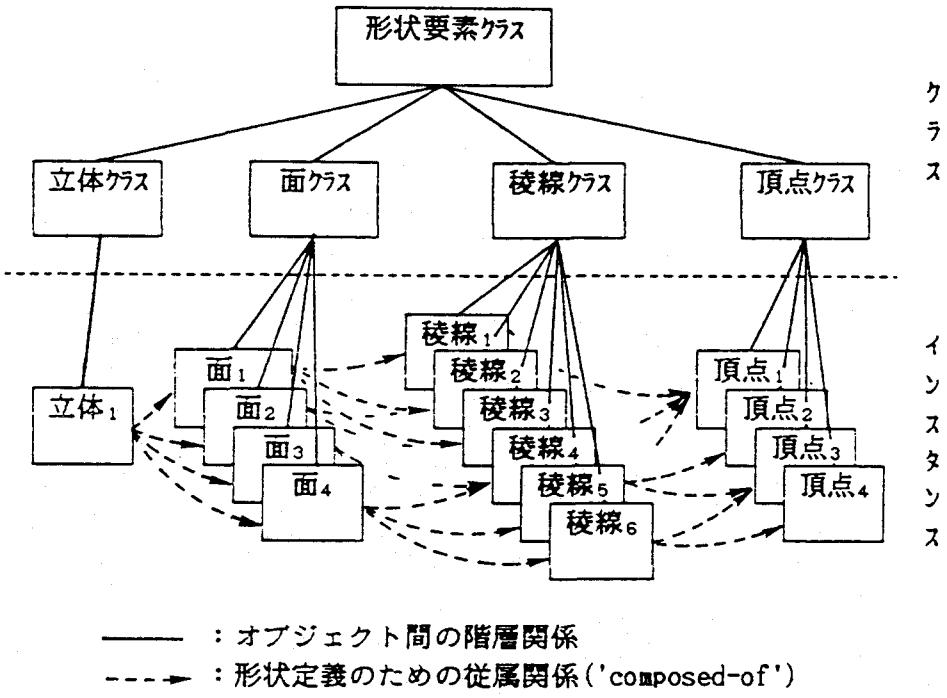


図3・19　形状表現のためのオブジェクト

ジェクト指向におけるデータと手続きを一体化する性質を生かして，容積や重心をはじめとするマスプロパティの算出や，形状の分割などの各種形状処理をそれらのクラスにメソッドとして記述するようにする．これによって，各種の形状処理がオブジェクト間のメッセージ・パッシングによって実現されるようになり，処理の記述も容易になる．

例えば，マスプロパティの計算は，対象の形状を一様な多面体として表現できるものに限定した場合，立体に対する各種の体積分は，ガウスの定理により立体の境界面を構成する各平面における面積分の和に変換することができ，さらに，そのような面積分は，グリーンの定理により各面の境界線を構成する線分における線積分の和へと変換することができる．これに基づいて，各種の積分処理を各オブジェクトのクラスにメソッドとして定義しておけば，図3・19に示したような形状要素のつながりをたどって，メソッドを起動するメッセージを送信していくことにより，各種のマスプロパティを求めることができる．

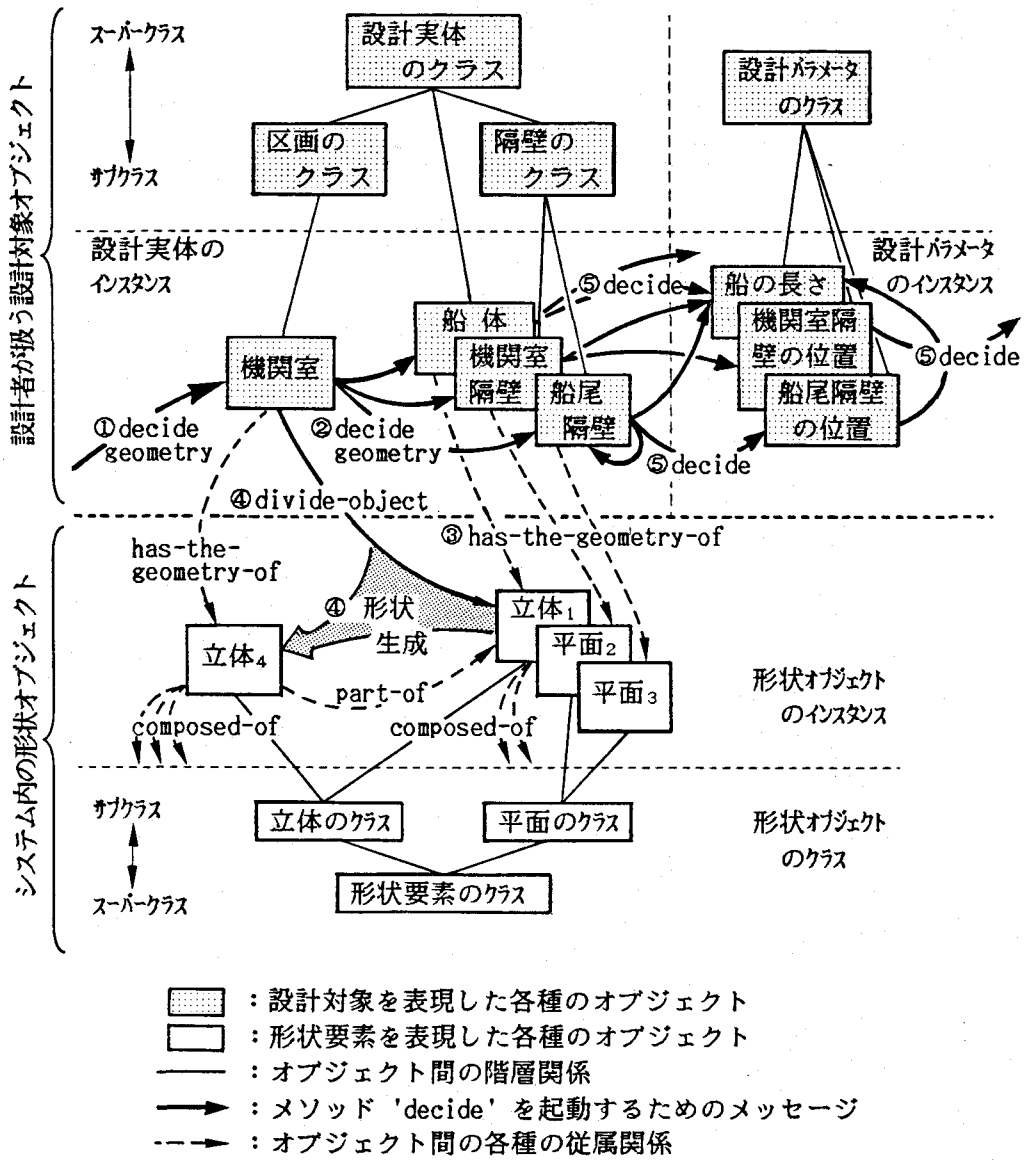


図3・20　形状操作処理の一例

3・6・4　形状操作のための設計知識の表現と設計処理の一例

次に，上記のような形状の表現に基づいた設計対象の具体的な取り扱いについて示す．図3・20はそのような処理過程の一例を示したものであり，後出の事例における機関室の形状の決定に係わる処理を示している．図の上半部は3・3

節で示した設計対象を表現するオブジェクトであり，前出の図3･6のようにして，形状の定義方法やマスプロパティの算出方法などをオブジェクト自身の変数 'procedure' に保持したり，クラスオブジェクトから継承したりしている．一方，下半部は図3･19に示したようなシステム内部で形状を取り扱うためのオブジェクトを示している．なお，両者のうち，上半部のオブジェクトが設計者の扱うオブジェクトである．これに対して，機関室の形状の決定は以下のようにして行われる．

まず，'機関室' のインスタンスに対してその形状の決定を求めるメッセージを送信する(①)と，そのインスタンスの変数 'procedure' に従って，そのために必要となる '船体' や '機関室隔壁' などの設計実体の要素を用いて(②)，形状の決定が行われる．その過程において，それらの設計実体の形状が定義されると，各設計実体に関するインスタンスは，関係 'has-the-geometry-of' により，システム内部でそれぞれに形状を表現したインスタンス '立体$_1$'，'平面$_2$'，'平面$_3$' と連結される(③)．そして，機関室の形状を決定するための実際の処理は，そのような形状インスタンスの間で行われ(④)，機関室の形状は '立体$_4$' として決定される．そのほか，以上のような形状操作に必要となる他の関連したパラメータや実体のインスタンスに対しても，同様のメッセージが送信され必要な項目が決定される(⑤)．

以上のような各種オブジェクトの協調した処理によりそれぞれの形状が決定されると，各設計実体のマスプロパティが計算できるようになり，その結果を用いることにより各種の機能を評価することができる．なお，後述の3･7･3項では船舶の区画配置設計を取り上げて，このような設計処理と形状処理の融合による処理の具体例を示す．

3･7 事 例 ── 船舶の基本設計への適用 ──

最後に，本システムを船舶の基本設計に適用した事例を示す．

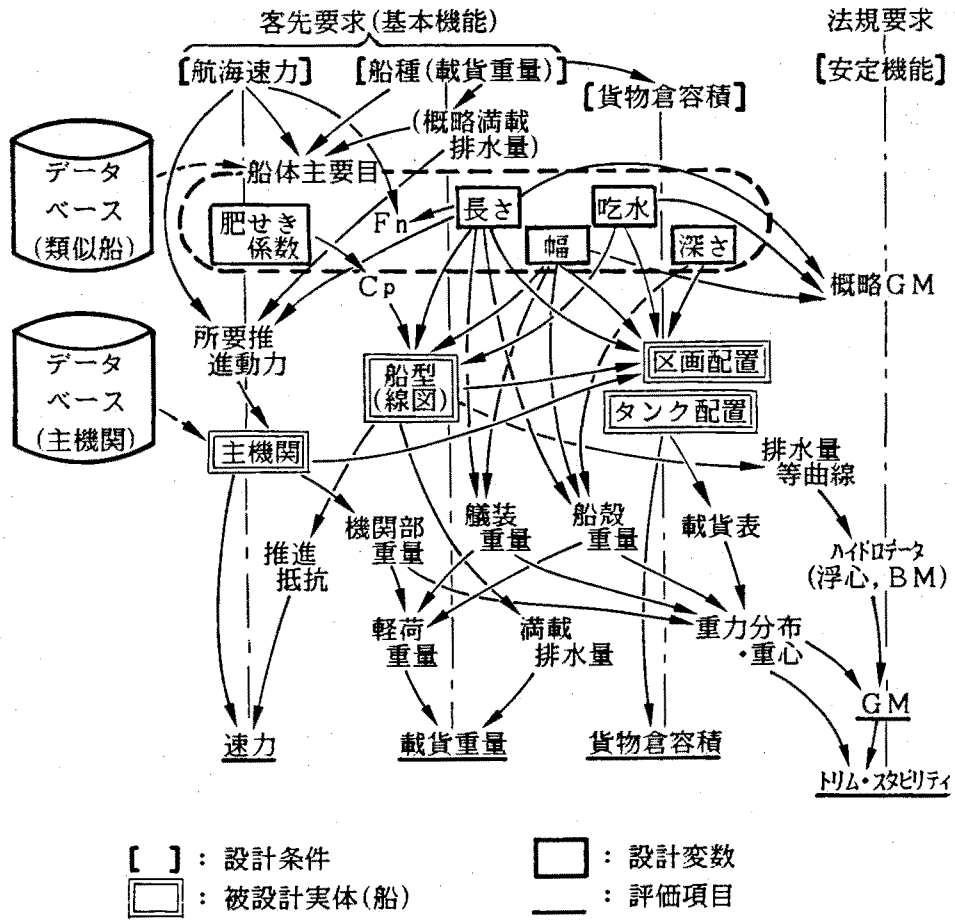


図3・21　船舶の基本設計過程

3・7・1　船舶の基本設計過程

事例として取り上げる船の基本設計[17)~19)]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），さらには，船体の形状や各種区画の配置を定めるプロセスである．このような設計の特質は，数理計画法による最適化とは異なり，多くの設計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点にある．すなわち，このプロセスでは，図3・21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っては，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

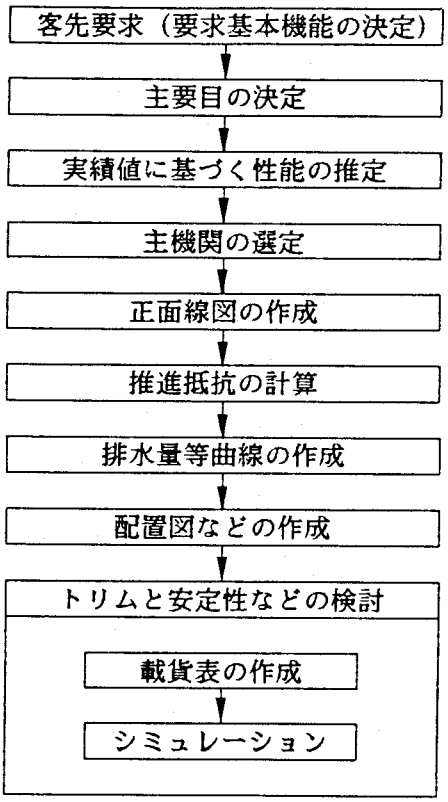


図3･22　船舶の基本設計の手順

繰り返すことによって設計を行う．しかし，これらの処理は複雑に連鎖し，前出図3･1のようなネットワークが設計全体に対して形成されるため，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操作することは困難である．また，設計対象に関する知識も膨大な量になるため，従来の枠組で柔軟な設計支援システムを構築することは困難である．このような点で，本システムの手法による支援に期待が持たれる．以下に，本システムにより，あるばら積み貨物船の基本設計を行った事例を示す．

なお，図3･21に示した設計過程は便宜的に図3･22のように書き下だすことができ，それに対応して設計プロセスが前出の図3･10のように記述される．また，この事例においてシステムに記述した知識の総量は，全体で，設計対象を表現したオブジェクトが約 500 個，評価診断ルールが約 100 個であった．

```
---------- DESIGN RESULTS ----------- STEP = 1 ---------- TIME = 3 ----------
                          PAST  1          PAST  2          NOW
 KIND-OF-SHIP .....            |   |            |   | BULK-CA |
 REQUIRED-DEAD-WEIGHT ....     |   |            |   | 38000.0 |
 REQUIRED-SERVICE-SPEED ..     |   |            |   | 14.9    |
 APPROX-FULL-LOAD-DISPLAC  46790.4 | +|            |   | 47944.2 |
 LENGTH ..................  187.38 |C+|            |   | 192.0   |
 BREADTH .................     |   |            |   | 27.1518 |
 DEPTH ...................     |   | 15.2181    |C+| 15.67   |
 FULL-LOAD-DRAFT .........     |   |            |   | 11.1092 |
 APPROX-BLOCK-COEFFICIENT      |   |            |   | .807659 |
 APPROX-GM-ON-FULL-LOAD .. 2.6314  | +| 2.67321    | -| 2.41902 |
 APPROX-GM-ON-BALLAST .... 3.89717 | +| 3.96408    | -| 3.67379 |
 P-VALUE-ON-FULL-LOAD .... 6.18646 | -| 6.13789    | +| 6.56094 |
 P-VALUE-ON-BALLAST ..... 6.31074 | -| 6.25726    | +| 6.6044  |
 TYPE-OF-MAIN-ENGINE ..... LOW-SPE |   |            |   | LOW-SPE |
 APPROX-REQUIRED-PS-MCO .. 8650.57 | -|            |   | 8599.9  |
 APPROX-LIGHT-WEIGHT ..... 8957.73 | +| 9272.89    | +| 9355.86 |
 APPROX-HULL-STEEL-WEIGHT  7482.96 | +| 7780.36    | +| 7863.33 |
 APPROX-OUTFIT-WEIGHT .... 864.907 | +|            |   | 886.234 |
 APPROX-MACHINERY-PART-WE  609.865 | -|            |   | 606.293 |
 APPROX-DEAD-WEIGHT ..... 37832.7 | +| 38671.3    | -| 38588.3 |
 NEEDED-VG-ON-GREN-FULL-L      |   |            |   | 53200.0 |
 APPROX-VG-WITH-TOP-SIDE-  50713.5 | +| 51964.0    | +| 53506.9 |
-------------------------------------------------------------------------------
```

図3･23　主要目の決定過程の実行例

3･7･2　主要目決定過程への適用

船舶の基本設計における主要目の決定過程は，主要目を設定しては実績値に基づいて基本的な性能を推定しつつ修正を繰り返すことにより，各種の機能を満足するような要目を決定する過程である．3･4節や3･5節で示した支援機能の実行例はこのような過程に対するものであり，それぞれ，設計を対話的に進めながら，設計結果に対する診断の表示をもとに設計案の修正を行って，各主要目を決定することができる．図3･23はある設計例における設計の履歴を示した表示画面であり，各設計項目の値が対比的に示されている．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 '-' で示されている．このような過程に続いて，主機関が選定され（第４章，4･6･1項に示す事例を参照のこと），さらに，船体の形状が図3･24に示すようなワイヤフレームとして決定され，その形状を操作して次項で示す区画配置などが行われていく．

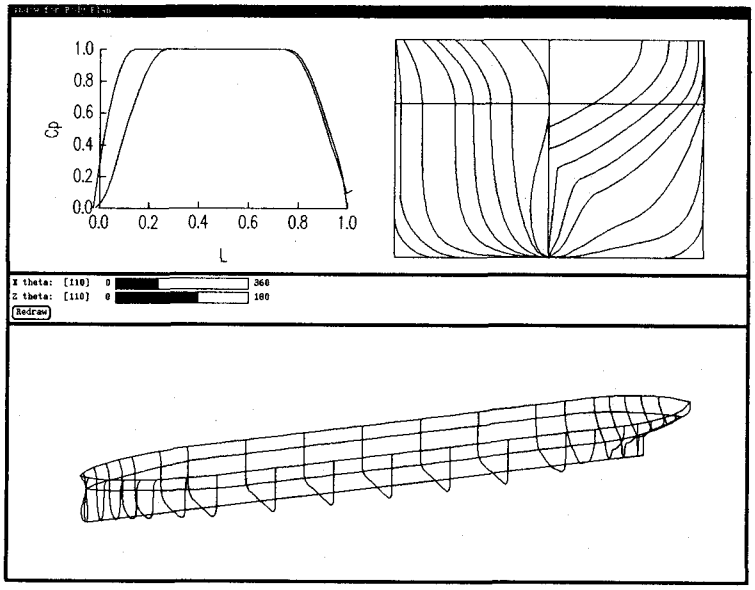
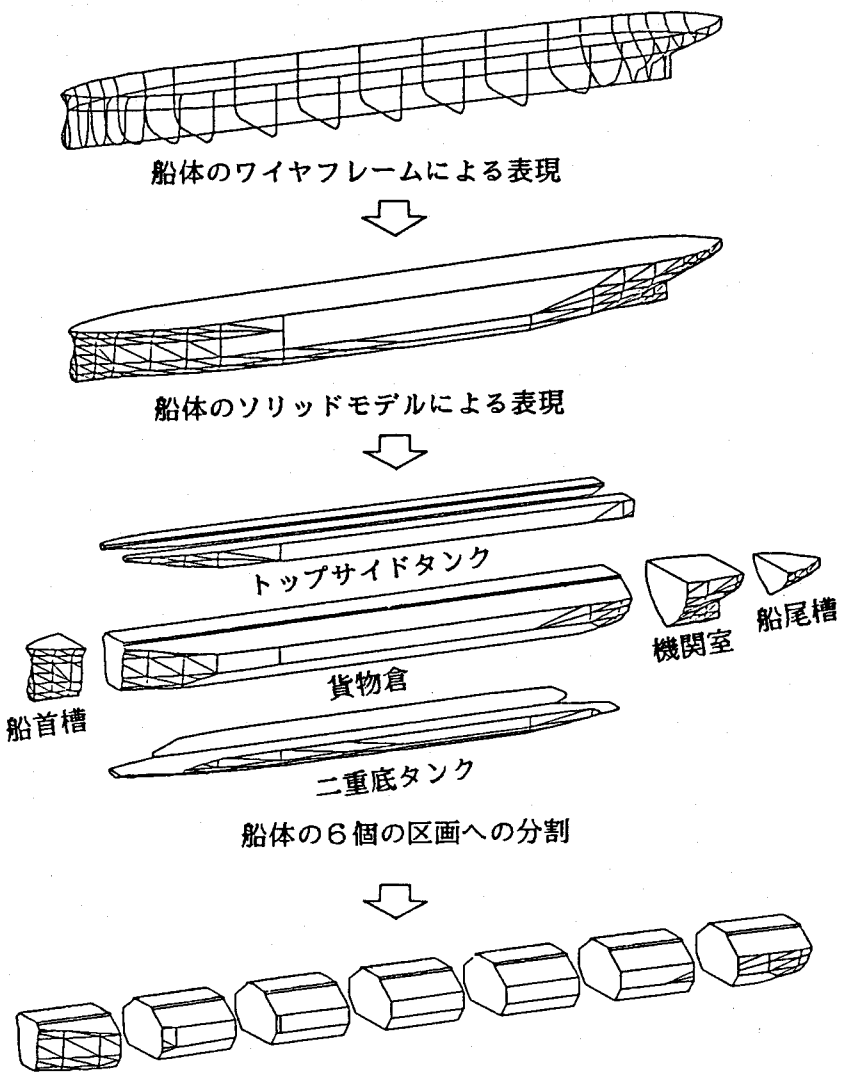


図3･24　船体形状のワイヤフレーム表現

3･7･3　区画配置過程への適用

船舶の区画配置は，船体内部を各種の区画に分割・配置して，容積や様々な載貨状態における安定性やトリムなどを検討し，必要に応じて配置を修正していく過程であり，3･6節で示した形状処理機能との融合のもとに本手法を導入することにより，船体や各種の区画を取り扱って有効に設計過程を支援することができる．

図3･25は，この過程における一連の処理を示したものである．まず，図中(a)のように，図3･24に示したワイヤフレームによる表現を B-Reps によるソリッド表現に変換した上で，船体を計６個の部分に分割し，さらに，中央部の貨物倉の部分を計７個の部分に分割する．続いて，図中(b)のようにして，分割された各部分の様々なマスプロパティが計算され，各載貨状態に対する安定性の検討が行われる．その結果，それぞれの載貨状態においてトリムが大き過ぎることなどが評価診断知識により指摘され，それに対して設計者は機関室の位置をやや前方に移動させようとしている．図中(c)はその変更を示したものであり，その変更に対して再びマスプロパティを計算し安定性の検討を行って



（ａ） 船体形状の分割処理

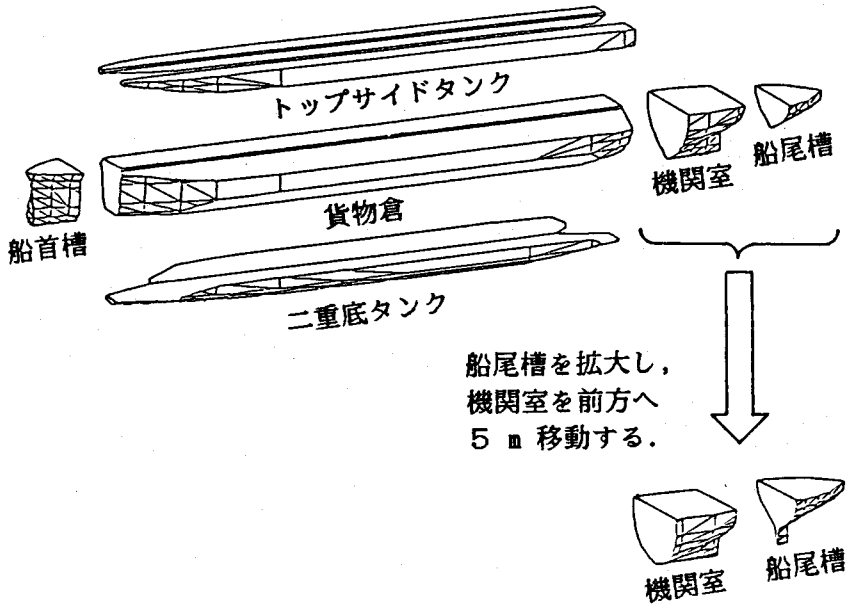
図3･25　区画配置設計過程の実行例（１）

いる過程が図中(ｄ)である．機関室の位置の変更によって，上記の問題点のうちの一つが解消されている．このような処理を繰り返すことによって，形状操作を伴う区画配置設計の過程を支援することができる．また，このような機関

```
### LOADING-TABLE-ON-GREN-FULL-LOAD

ITEM          USE          CAPACITY    WEIGHT      OXG          KG
FPT           BALLAST      1394.0967   0.0         -88.23918    9.123522
7-LEFT-TOPS   TS-BALLAST   475.60056   339.7147    59.177494    14.061586
6-LEFT-TOPS   TS-BALLAST   475.84845   339.89175   37.25644     14.060196
5-LEFT-TOPS   TS-BALLAST   475.84866   339.8919    15.328717    14.06021
4-LEFT-TOPS   TS-BALLAST   475.84866   339.8919    -6.5990014   14.060202
3-LEFT-TOPS   TS-BALLAST   475.84866   339.8919    -28.526714   14.060202
3-FUEL-TANK   C-OIL        332.8224    317.17975   -28.526714   0.84902334
2-FUEL-TANK   C-OIL        332.8222    317.17957   -28.526714   0.84902334
1-FUEL-TANK   C-OIL        332.82217   317.17953   -50.454437   0.8490234
APT           BALLAST      1008.5471   0.0         92.987434    13.030289
A-OIL         A-OIL        181.5398    163.38582   79.149284    0.84902334
WATER         WATER        417.4708    417.4708    88.150566    14.223083
LIGHT-WEIGH   LIGHT-WEIGH              8029.597    10.721378    9.763677
CONSTANT      CONSTANT                 218.60301   79.149284    17.358

     ....................................

---------- DIAGNOSIS ----------
①TRIM-ON-NORMAL-BALLAST IS TOO LARGE AS COMPARED WITH ( 0.015  * LENGTH ) .
②TRIM-ON-GREN-FULL-LOAD IS TOO LARGE AS COMPARED WITH 0.3     .
 ( GM-ON-GREN-FULL-LOAD / BREADTH ) IS TOO LARGE AS COMPARED WITH 0.1     .
---------- JUDGEMENT ----------
 N - NEXT STEP, S - NETWORK, M - MODIFY, E - END
-------------------------------------------------------------------------------
   ... ? m

---------- SELECT DESIGN VARIABLES TO BE CORRECT ----------
>>> Please input variable wanted to correct ! ( E - END )
   ... ? ENGINE-ROOM-BULKHEAD
>>> Please input ENGINE-ROOM-BULKHEAD's variable want to correct ! ( E - EXIT )
   ... ? x
>>> Please input variable wanted to correct ! ( E - END )
   ... ? AFT-PEAK-BULKHEAD
>>> Please input AFT-PEAK-BULKHEAD's variable want to correct ! ( E - EXIT )
   ... ? x
>>> Please input variable wanted to correct ! ( E - END )
   ... ? e
---------- DELETE & SET-UP DESIGN VARIABLES ----------
>>> input X of ENGINE-ROOM-BULKHEAD.
>>>   PRE-VALUE  now: -70.148
   ... ? -65.948
>>> input X of AFT-PEAK-BULKHEAD.
>>>   PRE-VALUE  now: -88.150566
   ... ? -83.950566
### VG-WITH-TOP-SIDE-TANK IS DELETED.
### VG IS DELETED.

     ....................................
```

（b） マスプロパティの算出と安定性に関する検討

図3･25 区画配置設計過程の実行例（2）



（c） 機関室の位置の移動による形状操作

図3･25 区画配置設計過程の実行例（3）

室の移動のほか，例えば，貨物倉の数を変更するような設計変更に対しても，図中(e)のように柔軟に対応することができる．

以上のように，本システムを適用することにより，船舶の基本設計の過程を有効に支援することができる．また，このような個別の問題における設計知識の記述も，オブジェクト指向に基づいた表現によって容易に行うことができる．

3･8 結 言

本章では，機能設計を対象に取り上げ，設計処理の基本として設計項目間の依存関係に着目したネットワークモデルを提案し，設計対象の表現方法ならびに設計処理の実現方法としてオブジェクト指向の概念を導入した設計支援システムについて示した．本システムは，

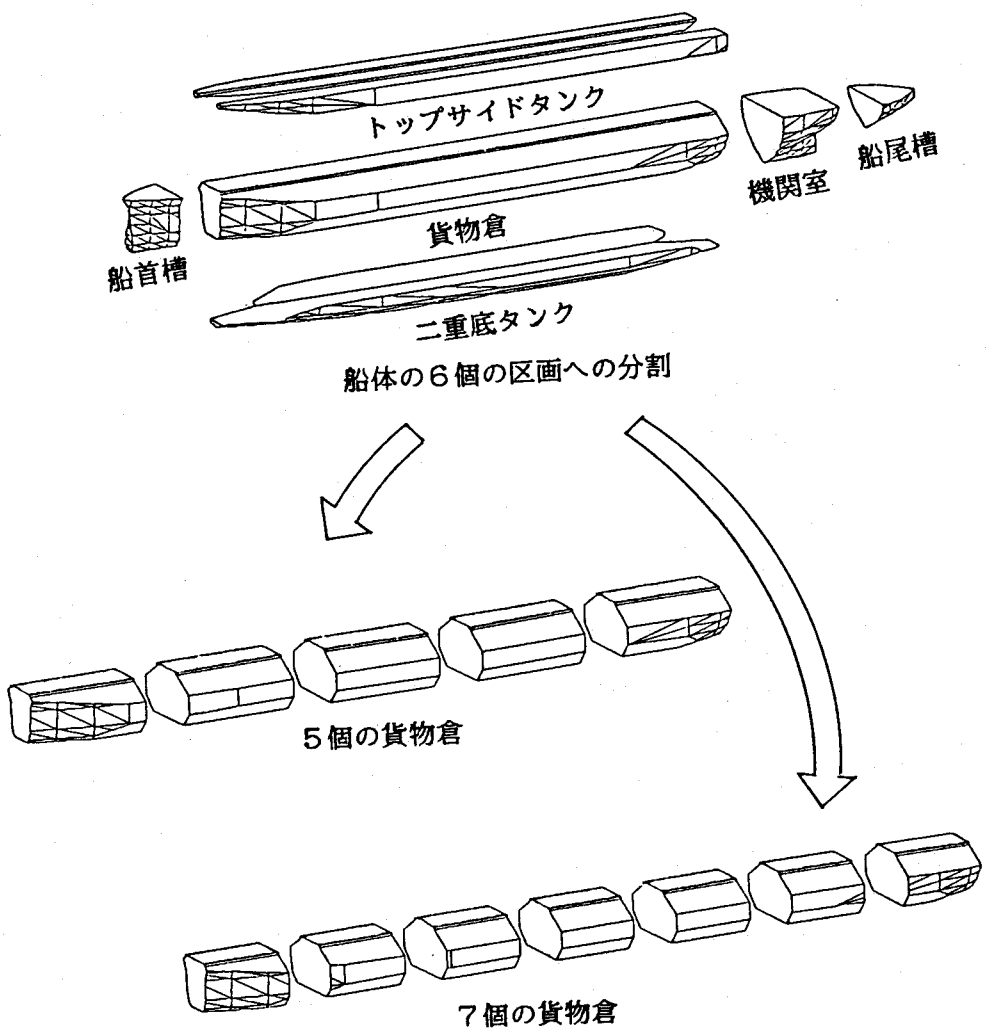
```
!!! LOADING-TABLE-ON-GREN-FULL-LOAD

ITEM          USE          CAPACITY    WEIGHT      OXG          KG
FPT           BALLAST      1394.0967   0.0         -88.23918    9.123522
7-LEFT-TOPS   TS-BALLAST   462.93408   330.6672    55.28616     14.059728
6-LEFT-TOPS   TS-BALLAST   462.82852   330.5918    33.956425    14.060201
5-LEFT-TOPS   TS-BALLAST   462.82834   330.59167   12.628714    14.060181
4-LEFT-TOPS   TS-BALLAST   462.82816   330.59155   -8.698994    14.060183
                           462.76[illegible]                    -30.025446

3-FUEL-TANK   C-OIL                    317.1796                 0.84902340
2-FUEL-TANK   C-OIL        332.82233   317.1797    -30.02671    0.8490233
1-FUEL-TANK   C-OIL        332.80704   317.1651    -51.353954   0.84906167
APT           BALLAST      1687.4265   0.0         90.15862     12.604335
A-OIL         A-OIL        181.5398    163.38582   79.149284    0.84902334
WATER         WATER        417.4708    417.4708    88.150566    14.223083
LIGHT-WEIGH   LIGHT-WEIGH              7902.4653   10.721378    9.763677
CONSTANT      CONSTANT                 218.60301   79.149284    17.358

    ....................................

---------- DESIGN RESULTS ----------- STEP = 9 ---------- TIME = 4 ----------
                            PAST  1      PAST  2      PAST  3        NOW
 DISP-ON-GREN-FULL-LOAD ..          |  |          |  | 48881.1 | -| 47817.6 |
 DRAFT-ON-GREN-FULL-LOAD .          |  |          |  | 11.765  | -| 11.5411 |
 GM-ON-GREN-FULL-LOAD ....          |  |          |  | 2.71805 | -| 2.71204 |
 TRIM-ON-GREN-FULL-LOAD ..          |  |          |  | .997876 | -| .004751 |
 DRAFT-F-ON-GREN-FULL-LOA           |  |          |  | 11.2381 | +| 11.5385 |
 DRAFT-A-ON-GREN-FULL-LOA           |  |          |  | 12.236  | -| 11.5433 |
 GZ-MAX-ON-FULL-LOAD-COND           |  |          |  | 1.70171 | -| 1.6449  |
---------------------------------- JUDGEMENT ----------------------------------
  N - NEXT STEP, D - DIAGNOSIS, E - END
-------------------------------------------------------------------------------
   ... ? d

---------------------------------- DIAGNOSIS ----------------------------------
①TRIM-ON-NORMAL-BALLAST IS TOO LARGE AS COMPARED WITH ( 0.015  * LENGTH ) .
②
---------------------------------- JUDGEMENT ----------------------------------
  N - NEXT STEP, S - NETWORK, M - MODIFY, E - END
-------------------------------------------------------------------------------
   ... ? m
```

（d） マスプロパティと安定性の再計算とそれに対する検討

図3･25 区画配置設計過程の実行例（4）

(1) 設計対象に関する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，知識の記述・追加・修正が容易となり，複雑な設計対象における大量の対象知識を比較的容易に記述してシステムを構築することができる．

(2) 設計処理をネットワークモデルに基づいて各種の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，設計対象におけるネットワー



（e） 貨物倉の数の変更による形状操作

図3･25 区画配置設計過程の実行例（5）

ク状の関係を自動的に操作でき，効率的な設計解の修正が可能になる．

(3) ネットワークモデルに基づいて感度解析や最適化手法を融合することにより，有効な設計支援を行うことができる．

(4) 設計対象のオブジェクトによる表現に対して形状処理を融合して，機能設計における形状の取扱いを有効に支援することができる．

(5) 設計解の評価診断知識や設計プロセスの記述により，統合的な設計支援環境を構築することができる．

(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したことにより，設計支援システムを種々の設計問題に対して比較的容易に適用することができる．

以上の諸機能によって，個別の設計問題に固有な設計知識の記述が容易になり，機能設計の過程を柔軟に支援することができる．また，事例として船舶の基本設計へ適用することにより，本システムおよび手法の有効性を検証した．

文　献

(1) 本論文，第２章文献(18)~(20)．

(2) 赤木，エンジニアリングシステム設計工学，(1982)，共立出版，pp.13-16.

(3) 赤木，設計論とコンピュータ利用技術(２)，機械の研究，vol.38，no.2，(1986)，養賢堂，pp.273-277.

(4) Lyon, T. D. and Mistree, F., "A Computer-based Method for the Preliminary Design of Ships", Journal of Ship Research, vol.29, no.4, (1985), pp.251-269.

(5) Zeleny, M., Multiple Criteria Decision Making, (1982), McGraw-Hill.

(6) MacCallum, K. J., "Understanding Relationships in Marine Systems Design", Proceedings of 1st International Marine System Design Conference, (1982), pp.1-9.

(7) Steele Jr., G. L., Common Lisp: The Language, (1984), Digital Press, (邦訳，後藤・井田，(1985)，共立出版)．

(8) 例えば，Winston, P. H. and Horn, B. K. P., LISP (3rd Edition), (1989), Addison-Wesley.

(9) SunView 1 Programmer's Guide, (1988), Sun Microsystems.

(10) SunCore Reference Guide, (1988), Sun Microsystems.

(11) 吉田ほか3名，知識工学による変電所機器レイアウトＣＡＤへの応用，日立評論，vol.67，no.12，(1985)，日立製作所，pp.971-974.
(12) 池田・山本，数式処理とＬＩＳＰ，マイコンピュータ，no.15，(1984)，ＣＱ出版社，pp.52-69.
(13) 例えば，Vanderplaats，G. D.，Numerical Optimization Techniques for Engineering Design: with Applications，(1984)，McGraw-Hill.
(14) 沖野，自動設計の方法論，(1982)，養賢堂.
(15) 例えば，Drake，S. and Sela，S.，"A Foundation for Features"，Mechanical Engineering，vol.111，no.1，(1989)，ASME，pp.66-73.
(16) 例えば，今村ほか，オブジェクト指向概念による製品モデリング，日本機械学会・精密工学会 第５回設計自動化工学講演会講演論文集，(1987-7)，pp.58-60.
(17) 冨田，船舶基本設計論，(1982)，丸善出版サービスセンター.
(18) 教育テキスト研究会，商船設計の基礎知識（上巻），(1977)，リプロ.
(19) 関西造船協会 編，造船設計便覧（第３版），(1976)，海文堂.

第4章　エネルギープラントの機器構成設計支援システム

4·1 緒言

コージェネレーションプラント[1]をはじめとするエネルギープラントの設計計画においては，カタログデータの中から各種の機器を選定して，プラントを構成する「機器構成設計」が大きな比重を占めており，その結果は，設計対象のエネルギー効率などを左右する．しかし，従来の設計では，プラントの構成方法や各機器の種類に多くの代替性があるため，過去の設計事例や経験に基づいて，いくつかのケースについて設計が行われていたにすぎず，必ずしも十分な検討が行われているわけではなかった．このため，コンピュータの援用による合理的な設計支援に大きな期待が寄せられている．機器構成設計は，いわゆる「知的検索形」の問題であり，その特質は様々なカタログデータや多量の設計データを取り扱って検索を進める点にあり，そのようなデータの取扱いの方法が設計支援システムを構築する上で重要となる．また，エネルギープラントの優劣は，プラントの構成のみならず，運用計画によっても大きく左右されるため，具体的なプラントの候補を評価して望ましいものを選択していくためには，合理的な運用計画手法を融合する必要がある．赤木らは，このような点から，検索処理を知識情報処理手法であるプロダクションシステムとフレーム表現により，運用計画を数理計画法による最適化手法により取り扱うというハイブリッド手法によって，舶用動力プラントの設計を支援するシステム[2][3]を開発している．

本章では，このようなエネルギープラントの機器構成設計を，より容易に，より柔軟に支援できる手法として，オブジェクト指向プログラミング[4]に基礎を置いた方法を提案し，その手法を基本として構築した舶用動力プラント，ならびにコージェネレーションプラントの設計に対する設計支援システムにつ

いて示す．本システムの特徴は，設計知識，検索対象の機器データやプラント候補に関するデータ，エネルギー需要量に関するデータなどをそれぞれに分離し，それらをオブジェクト指向に基づいて階層的に細分化して表現する点にあり，これにより，各種の設計データの表現におけるモジュール性が向上し，データの入れ換えなども容易になり，システムの柔軟性が向上する．さらに，各種の設計処理もオブジェクトのメソッドとして容易に記述できるようになる．

4･2　プラントの機器構成設計におけるタスクとその設計手法

4･2･1　プラントの機器構成設計の性質

エネルギープラントの機器構成設計は，設計対象における様々なエネルギー需要量を総合的にバランスよく供給できるようなプラントの機器構成や各機器の種類を決定する過程である．近年，コージェネレーションプラント[1]をはじめとして，各種エネルギープラントの構成は効率的なエネルギー利用をめざして複雑化する傾向にあり，そのような設計を設計者の能力のみで合理的に行うことはますます困難になりつつある．このような設計の複雑さは以下の点に起因している．

(1) エネルギー需要量の複雑さ：エネルギープラントに対する需要量の種類は一般に複数であり，その量は時間や季節によって大きく変動する．

(2) 機器構成の複雑さ：(1)のような各エネルギー需要量を効率よく賄うためには各種の機器が必要となり，プラントの構成は複雑なものとならざるを得ない．

(3) 運用計画の代替性：(2)のような各機器の運用には多くの代替性が存在し，それによってプラントの経済性が大きく左右されるため，設計の段階で運用計画を考慮しなければ，各プラントの経済性を評価することはできない．

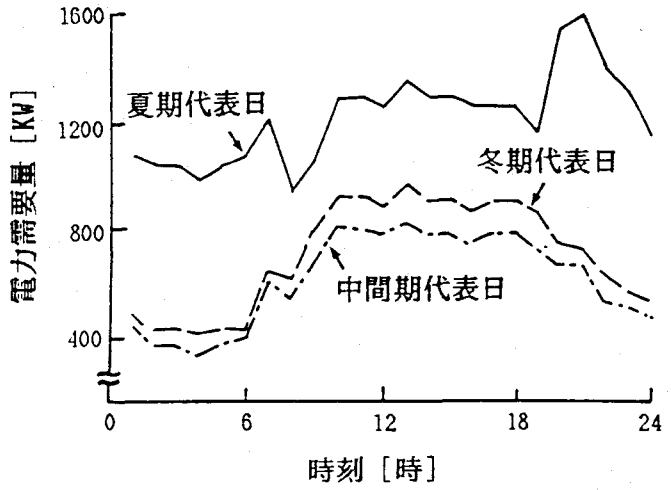
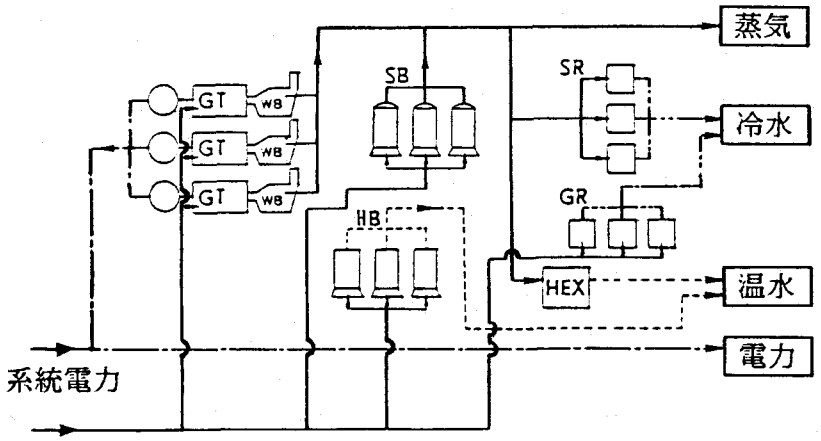


図4・1 エネルギー需要量の変動の一例
（あるコージェネレーションプラントに対する需要）



機器の記号

GT：ガスタービン	SR：蒸気吸収冷凍機
WB：廃熱ボイラ	GR：ガス焚吸収冷凍機
HB：給湯用補助ボイラ	HEX：熱交換器
SB：蒸気用補助ボイラ	

図4・2 エネルギープラントの構成の一例
（ガスタービン方式コージェネレーションプラントの場合）

例えば，コージェネレーションプラントにおいては，電力需要や冷房・暖房・給湯などの熱需要を賄う必要があり，かつ，その量は，例えば，電力需要量については図4･1のように大きく変動する．さらに，プラントの構成は図4･2のような複雑なものとなり，発電を行うための原動機を軸として，その廃熱を利用するための廃熱ボイラや蒸気吸収冷凍機，さらに，原動機廃熱のみでは不足する熱量を供給するための各種ボイラやガス焚吸収冷凍機などから構成される．このため，同じエネルギーを供給しようとする場合でも，どの機器を用いるかによって多くの代替性が存在することになる．

4･2･2　プラントの機器構成設計のタスクと設計支援過程

上記のように複雑なプラントの設計過程を支援するために，機器構成設計の過程をいくつかのタスクに分けて整理しておく．すなわち，

(1) エネルギー需要量の把握：設計対象の種類や規模に基づいて，機器構成設計の過程で参照する各種エネルギー量の変動パターンを設定する過程．

(2) 構成機器の種類の決定と各機器候補の選定：(1)をはじめとする設計条件をもとにプラントを構成する機器の種類を決定し，それぞれに対して，多様な機器候補の中から需要量を合理的に供給し得る機器を選定してその台数を絞りこむ過程．

(3) プラント候補の合成：(2)で絞りこんだ各機器を組み合わせて，それらから構成されるプラントの候補を合成する過程．

(4) 運用計画を考慮したプラント候補の評価：(3)で合成した各プラントの候補に対して最適な運用計画を算定して，それぞれの経済性を評価し，適切なプラントを選定する過程．

以上のタスクが存在するものとして設計過程を理解する．図4･3は，提案する手法において，設計を進めていくための手順を示したものである．図の手順は，上記の各タスクが厳密に(1)から(4)の手順で進行するのではなく，(2)から(4)に至る過程を繰り返しながら設計が行われていくことを示している．これは，複雑なプラントの構成に対して，(2)や(3)の段階で，プラントの候補を徐々に

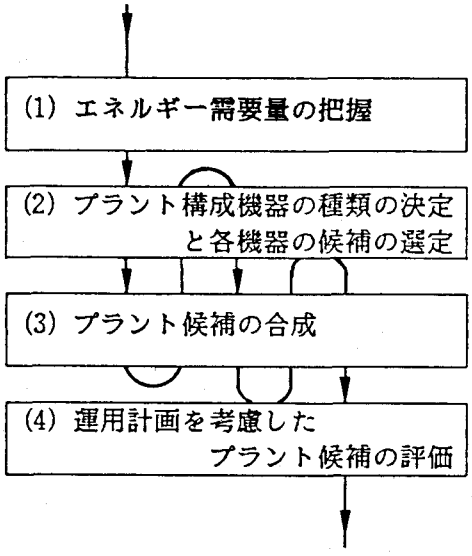


図4・3　プラントの機器構成設計の手順

具体化させつつ，(4)の評価を概算して不合理な候補を排除することにより，組合せの爆発を押えながら，設計を進めていくようにするものであり，その上で，最終的なプラントの候補に対して，数理計画法を用いて運用計画を算定する[5]ことにより詳細な評価を行うようにする．これによって，合理的な候補を検討しつつ，設計を行うことができる．

4・2・3　オブジェクト指向による設計モデル表現

4・2・1項で前述したように，プラントの機器構成設計は複雑な問題であり，その過程で扱われる設計知識や設計データの種類や数も多くのものになる．そのような知識やデータの表現に対して，以下の理由により，オブジェクト指向による表現[4]を導入する．

(1) モジュール性：各種のデータや知識をそれぞれに，オブジェクトとしてモジュール化して表現することにより，データの追加や修正，システムの構築などが容易になる．

(2) 情報隠蔽の性質：各種の設計処理を各オブジェクトのメソッドとして記

述して，メッセージ・パッシングにより実行することにより，それらを抽象化して扱うことができる．

(3) 階層性：各種のデータの種類や性質における階層性をスーパークラス・サブクラス，クラス・インスタンスの関係に対応させることによって，データや設計処理の記述を簡潔に行うことができる．

また，このような表現で用いるオブジェクトは大きく，エネルギー需要量に関するデータ，カタログデータを含んだ各種機器のデータ，プラント候補の表現，設計知識の記述に分けることができる．全体としての処理は，それらがそれぞれに設計処理のためのメソッドを保持し，メソッドが連携することにより行われることになる．

次節以降では，4･6節で取り上げる二つの事例を引きながら，本手法の具体的な内容について示す．

4･3 設計支援システムの構成

プラントの機器構成設計支援システムの構成を図4･4に示す．システムは，第3章で示した機能設計支援システムのもとで各種の設計知識を記述する一方，新たに，プラント設計に固有の知識ベースを追加することにより構成する．すなわち，第3章で示したシステムの機能のうちの設計知識（設計パラメータ）の処理機能と設計プロセスの管理機能に加えて，プラント設計のための各種の設計データを表現するためのクラスや，それに関する固有の処理を行うためのメソッドを記述した LISP 関数，検索の対象となる具体的なカタログデータを表現したオブジェクト（インスタンス）などを記述することにより構成する．また，知識ベースは，設計知識を表現した設計オブジェクト，設計プロセスの進行を管理するための述語，エネルギー需要量・機器データ・プラント候補などを表現する各種のデータオブジェクト，個別設計処理を記述した LISP 関数，ならびに最適化計算や作画を行う FORTRAN プログラムから構成される．

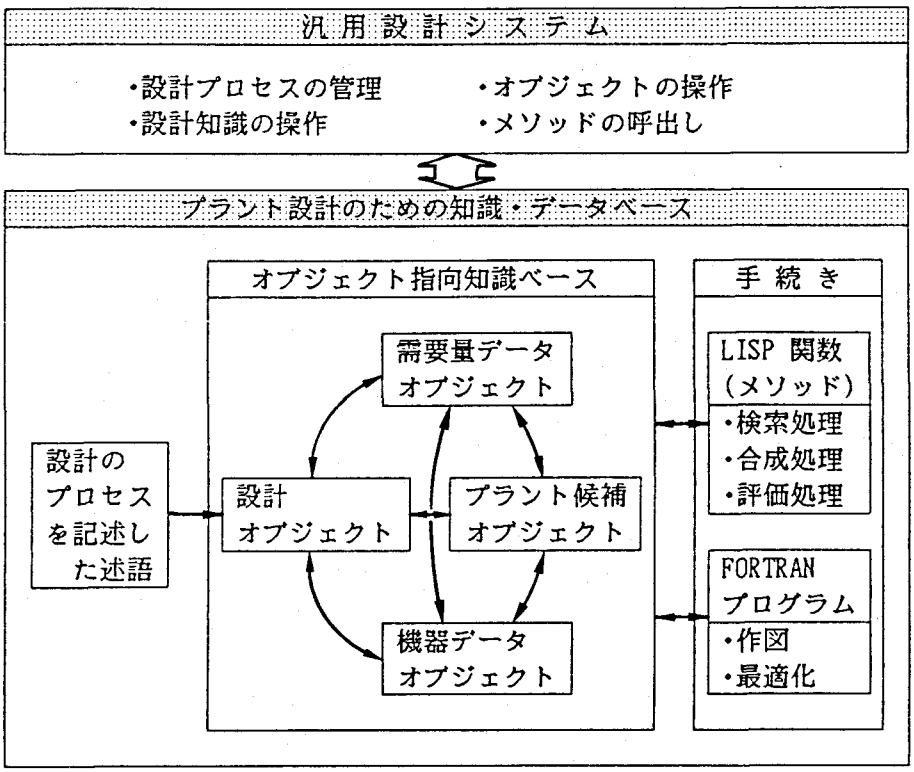


図4・4　設計支援システムの構成

なお，本システムは，大型コンピュータ ACOS2000 上の UTILISP[6] により記述されている．また，オブジェクト指向プログラミング環境は，UTILISP 上で，属性リスト，連想リストなど[7]を用いて構築されている．次節以降では，このようなデータや知識の表現，ならびに設計処理の方法について示す．

4・4　設計データ・設計知識の表現

4・4・1　エネルギー需要量の表現

前述のように，エネルギー需要量には，電力・冷房用熱量・暖房用熱量などの種類があるとともに，さらに，それらは季節的にも時間的にも変動する．例

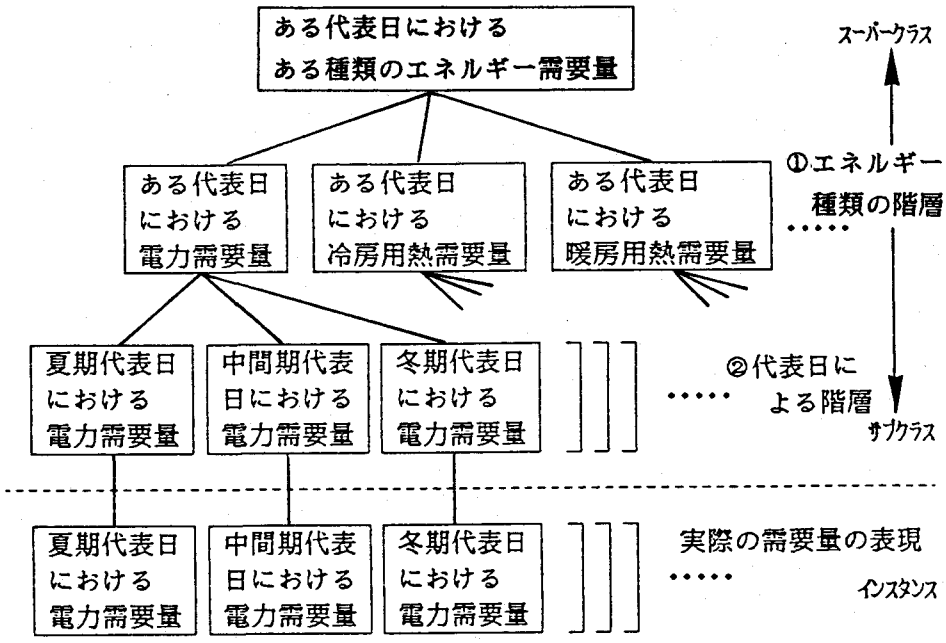


図4･5　エネルギー需要量の表現

えば，コージェネレーションプラントの場合，その設計計画においては，このような変動を取り扱うために季節ごとに代表日を設定して，それらの日におけるエネルギー需要量の変動をもって全体のエネルギー需要量にかえる場合が多い．そこで，前出図4･1のような各エネルギー，各代表日ごとのエネルギー需要量の変動をそれぞれ，あるオブジェクトの変数として表現するようにする．図4･5はこのような表現におけるクラス・インスタンスの階層関係を示したものである．最上位のクラスはエネルギー需要量という概念に対応し，①のレベルのクラスはエネルギー需要量の種類に対応し，さらに，②のレベルのクラスは各代表日に対応し，また，具体的な設計事例における需要量はそのようなクラスのインスタンスに表現するようにする．図4･6は具体的なオブジェクトの記述例である．図中(a)は，各種のエネルギー需要量を取り扱うための各種手続きをメソッドとして保持した最上位のクラスの記述である．(b)は，具体的な需要量を表現したインスタンスの記述であり，各時刻における需要量を保持

```
(CLASS (NAME DEMAND)
  (IV (DEMAND     NIL (ACCESS ((BEFORE IF-NIL-THEN-METHOD&PUT))) )
      (MAX-DEMAND NIL (ACCESS ((BEFORE IF-NIL-THEN-METHOD&PUT))) )
      (AVERAGAE   NIL (ACCESS ((BEFORE IF-NIL-THEN-METHOD&PUT))) )
      (MIN-DEMAND NIL (ACCESS ((BEFORE IF-NIL-THEN-METHOD&PUT))) ))
  (IM (DEMAND        (CALL GET-DEMAND)          )
      (MAX-DEMAND    (CALL GET-MAX-DEMAND)      )
      (AVERAGAE      (CALL GET-AVERAGE-DEMAND)  )
      (MIN-DEMAND    (CALL GET-MIN-DEMAND)      )
      (MODIFY-DEMAND (CALL MODIFY-DEMAND)       )) )
```

（a） 需要量オブジェクトに対するスーパークラス

```
(INSTANCE (NAME ELEC-SUMMER) (CLASS CLASS-OF-ELEC-SUMMER)
  (IV (DEMAND ((1  470) (2  400) (3  380) (4  360) (5  340) (6  510)
               (7  520) (8  460) (9  490) (10 125) (11 130) (12 130)
               (13 130) (14 130) (15 150) (16 130) (17 400) (18 680)
               (19 670) (20 690) (21 510) (22 490) (23 470) (24 440)) )))
```

（b） 夏期電力需要量のインスタンス

図4･6 エネルギー需要量のオブジェクトの記述例

している．なお，(b)のような設計条件となる需要量は，パターン化された典型的なサンプルデータを参照するなどして，対話的に設定することにする．また，このようなオブジェクトを，後出の様々なオブジェクトと区別するために，需要量データオブジェクトと称することにする．

4･4･2 各種機器とそれらのカタログデータの表現

プラントを構成する各種の機器についても，それぞれの機器をオブジェクトとして表現する．このような機器に関するデータ[8]は，機器本来のカタログデータと，具体的な設計事例においてエネルギー需要量やプラントの構成に対応して定まるデータとに分けられる．さらに，各種のカタログデータやそれに付随して性能などを算出するための手続きには，その種類，形式，モデルなどによって階層的な共通性がある．これに基づいてオブジェクト指向における階

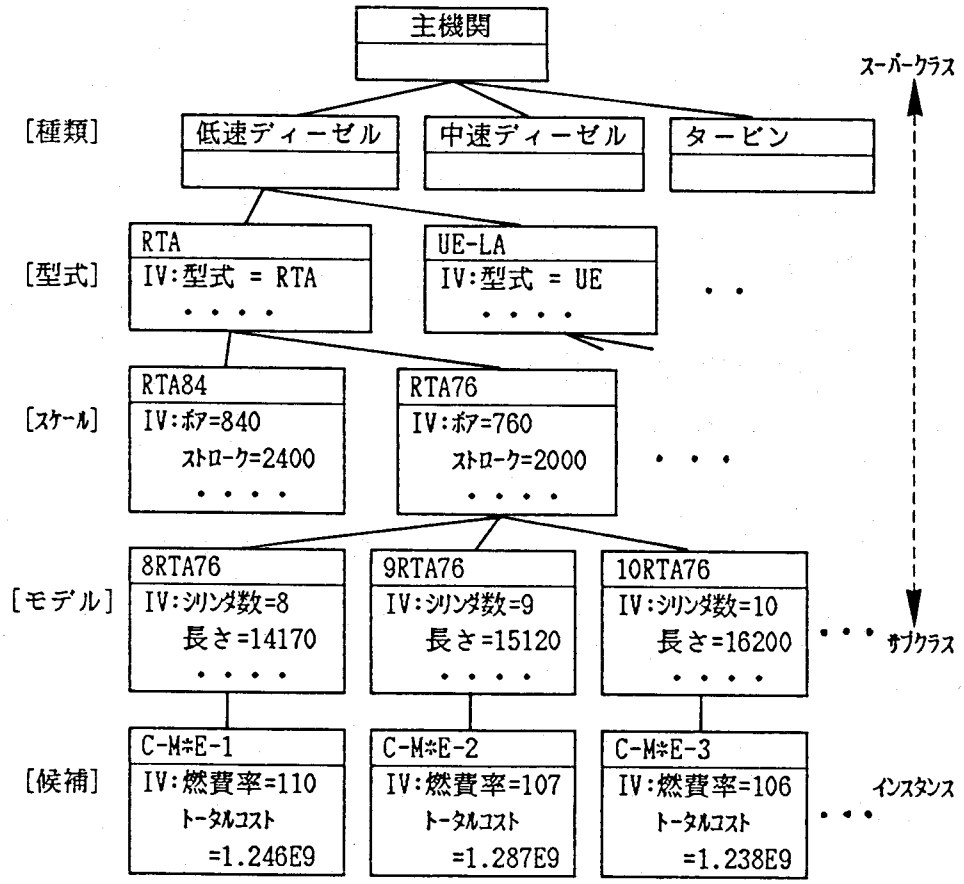


図4･7　機器データの表現とその階層関係

層性を利用することにより，各種の機器についてのデータを図4･7のようにして表現することにする．図は4･6･1項で取り上げる舶用動力プラントにおける主機関に関するデータ表現の例である．まず，最上層に主機関の表現における一般的な枠組としてクラスを設定し，主機関において共通的な処理を記述する．続いて，その下層に主機関の種類による階層を設け，「低速ディーゼル」・「中速ディーゼル」・「タービン」などの各種類に対応するクラスを上記のクラスのサブクラスとして設けて，その種類によって共通となるデータや種類に固有な処理をオブジェクトの変数やメソッドとして表現する．さらに，その下層には主機関の型式やモデルなどに対応したクラスの階層を設けて，同様に各種

```
(CLASS (NAME RTA86)
  (SUPERCLASS RTA)
  (IV (BORE          760.0    )
      (STROKE        2200.0   )
      (SPEED         ((R1  98.0) (R2  98.0) (R3  71.0)(R4  71.0)) )
      (FUEL-CON100 ((R1 127.0) (R2 121.0) (R3 126.0)(R4 121.0)) )
      (FUEL-CON85  ((R1 125.0) (R2 121.0) (R3 124.0)(R4 120.0)) )
      (PME           ((R1 16.93) (R2  9.29) (R3 16.83)(R4 12.83)) )
      (HEIGHT        1256.0   )
      (WIDTH         4100.0   )
      (POWER-PER-CYLINDER
              ((R1 3680.0) (R2 2020.0) (R3 2650.0)(R4 2020.0)) )
      (CYLINDER-RANGE ( 4 ≒ 8 10 ) )                              )
  (IM (LENGTHO (LAMBDA (N)
        (COND ((LESSP N 9)
                  (PLUS 8370.0  (TIMES 1450.0
                                       (DIFFERENCE N 4))) )
              ( T (PLUS 16655.5 (TIMES 1450.0
                                       (DIFFERENCE N 4))) )) )) ) )
```

（a） 主機関のモデルのクラス

```
(INSTANCE (NAME C-M≒E-2)
   (CLASS 6RTA86)
   (IV (FUEL-COMSUMPTION 122.591)
       (PRICE            7.218E8)
       (TOTAL-COST       1.287E9)) )
```

（b） 主機関の候補のインスタンス

図4･8 機器データのオブジェクトの記述例

のデータや手続きを記述する．そして，最下層のクラスが各種の機器の具体的なモデルに対応するようにし，個々の設計事例に依存しないデータをこのレベルまでに表現しておく．そのもとで，各設計事例における具体的な性能値などの個別データをそれらのインスタンスに表現するようにする．このような表現により，オブジェクト間の継承の性質を利用することによって機器属性の記述が簡潔になり，さらに，オブジェクトによる表現の有する高いモジュール性に

より新しい機器データの追加や機器属性の変更に対しても容易に対応できるようになる．図4･8はこのような表現の具体的な記述例である．(a)は，主機関のモデルの一つである 'RTA86' に対応するものであり，RTA 型の低速ディーゼルに対応したクラスのサブクラスで，その変数(IV)には，機器属性であるボアやストローク，シリンダ当りの出力，回転数などが保持されている．また，メソッド(IM)には，シリンダ数が確定した時に機関の全長を計算するための手続きが記述されている．(b)は，設計の過程で生成される(a)のインスタンスであり，実際の動作点における燃料消費率などを保持している．このほか，プラントを構成する各種の機器（ターボ発電機，ディーゼル発電機など）も同様の方法で表現することができる．なお，このようなオブジェクトを，他のオブジェクトと区別するために，機器データオブジェクトと称することにする．

4･4･3 プラント候補の表現

プラントの候補は4･5･3項で後述するような方法で生成するが，ここではそのデータ表現の方法について示す．プラントは各種の機器を組み合わせることによって構成され，それに対応して図4･9のような方法でそれぞれの候補を表現する．すなわち，個々のプラント候補をオブジェクトに対応させ，それを構成する機器の名称をそのオブジェクトの変数として保持させることにより表現する．例えば，図中の 'Plant-1' は，'C-M*E-1'・'D*G-1'・'T*G-1' などの機器から構成されており，変数の記述を通じて各構成機器の出力などの性能特性を参照することができる．また，そのようなオブジェクトのクラスには，プラントの構成方式に対応した様々のクラスを用意し，プラント構成によって定まる各種の性能値などを計算したり，プラント合成処理を進めていくための手続きを記述したメソッドを保持させるようにする．図4･10はそのようなオブジェクトの具体的な記述例である．(a)は，あるプラント構成方式に対応したクラスの記述であり，初期コストの算出や最適運用方策の決定を行うためのメソッドを保持している．(b)は，設計過程で生成されたあるプラント候補の記述であり，このオブジェクトが(a)のインスタンスであることのほか，各構成機器

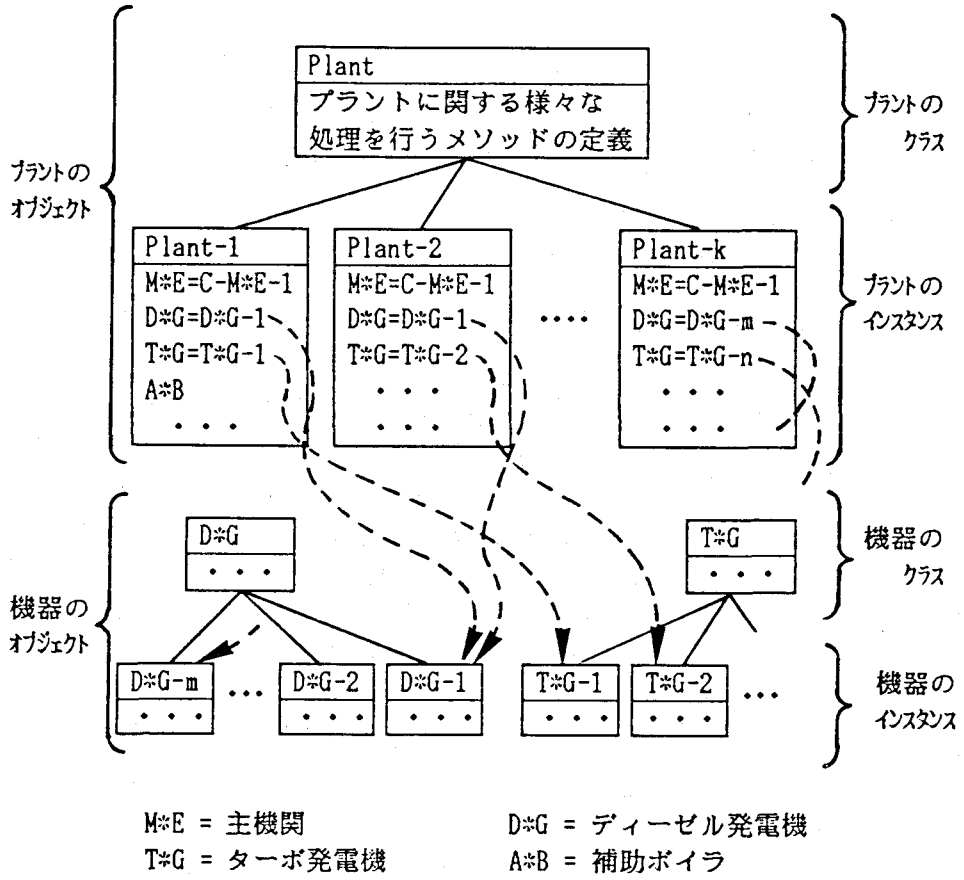


図4･9　プラント全体構成の表現

の名称を変数に保持している．なお，このようなオブジェクトを，他のオブジェクトと区別するために，プラント候補オブジェクトと称することにする．

4･4･4　設計知識の表現

プラントの機器構成設計における設計知識には多種多様のものがあるが，それらは大きく以下の二つに分けることができる．

(1) 個々の機器やプラント候補，需要量データの取扱いに関する知識．

(2) 設計における検索や合成，評価などの処理を全体として行うための知識．

前者については，前項までに示したように，個々のデータのオブジェクトによ

```
(CLASS (NAME PLANT)
  (IM (INITIAL-COST (CALL CALCULATE-INITIAL-COST))
      (RUNNING-COST (CALL CALCULATE-RUNNING-COST))
      (TOTAL-COST   (CALL CALCULATE-TOTAL-COST)  )
       .....          .....                    )
  (IV (INITIAL-COST NIL (ACCESS ((BEFORE IF-NIL-THEN-METHOD&PUT))) )
      (RUNNING-COST NIL (ACCESS ((BEFORE IF-NIL-THEN-METHOD&PUT))) )
      (TOTAL-COST   NIL (ACCESS ((BEFORE IF-NIL-THEN-METHOD&PUT))) )
       .....          .....                  )  )
```

(a) プラント候補に対するクラス

```
(INSTANCE (NAME PLANT-108)
   (CLASS PLANT)
   (IV (EXHAUST-GAS-ECONOMIZER E*E-CADIDATE1)
       (SHAFT-GENERATOR        S*G-CADIDATE1)
       (TURBO-GENERATOR        T*G-CADIDATE4)
       (AUXILIARY-BOILER       A*B-CADIDATE9)
       (DIESEL-GENERATOR       D*G-CADIDATE1)
       (INITIAL-COST           9.2934131E+08)
       (RUNNING-COST           5.2210513E+07)
       (TOTAL-COST             9.8155183E+08)) )
```

(b) プラント候補のインスタンス

図4・10 プラント全体構成のオブジェクトの記述例

る表現におけるクラスにメソッドとして記述する．後者については，第3章で示した汎用システムの機能のうち，設計知識のなかでも設計パラメータに対する処理機能を用いてオブジェクトとして表現することにする．この方法を用いることにより，個々の設計処理を各種のデータ表現と同様にモジュール化して表現することができ，知識の入れ換えなども容易になる．また，設計のプロセスも比較的容易に記述できるようになる．なお，本章では，このようなオブジェクトを，前述の各種のデータオブジェクトと区別するために，設計オブジェクトと称することにする．

設計知識のオブジェクトによる記述例を，4・6・1項で事例として取り上げる舶用動力プラントの主機関選定過程をもとにして，図4・11に示す．このような

```
(INSTANCE (NAME RPM-RTA)
   (CLASS DNA)
   (IV (PROCEDURE ( R-VS-KNOT * 60.0 * 0.514 // ( 0.65 * DF ) ) )) )
```

（a） 主機関の所要回転数を決定する手続きの記述

```
(INSTANCE (NAME KIND-OF-MAIN-ENGINE)
   (CLASS SNNA)
   (IV (PROCEDURE
         ( ( KIND-OF-SHIP = FERRY )       --> 'MID-SPEED-DIESEL
           ( ( KIND-OF-SHIP = TANKER ) | ( KIND-OF-SHIP = LNG )
               & ( A-PS-MCO > 55000 ) ) --> 'TURBINE
           ( A-PS-MCO < 55000 )           --> 'LOW-SPEED-DIESEL ) )) )
```

（b） 主機関の種類に関する記述

```
(INSTANCE (NAME CHECK-RPM-LIST-M*E)
   (CLASS DNNL)
   (IV (PROCEDURE
        ( SATISFY? (SEND-MESSAGE *** 'GET-VALUE 'CM 'CHECK-RPM) )
        ( OBJECT-OF-SEARCH ( SUBCLASS OF OBJECT-OF-SEARCH-M*E ) ) )) )
```

（c） 所要回転数を満足する主機関の検索に関する記述

```
(INSTANCE (NAME BEST-3-FUEL-COMSUMPTION-LIST-M*E)
   (CLASS DNNL)
   (IV (PROCEDURE
         ( SELECT-BEST-N 3 ( FUEL-COMSUMPTION OF *** ) )
         ( OBJECT-OF-SEARCH ( SATISFY-LIST-M*E )       ) )
       (ITEM FUEL-COMSUMPTION)                          ) )
```

（d） 燃料消費率の優れた主機関を求める処理に関する記述

図4･11 設計知識を表現したオブジェクトの記述例

表現では，個々の設計処理項目をそれぞれオブジェクトに対応させ，そのオブジェクトの変数 'PROCEDURE' に設計処理の内容を記述する．(a)は船舶の主機関の所要回転数に関するもので，手続きとして 'V_S*60*0.514/(0.65*d_F)' という算術計算式が記述されており，回転数が船の航海速度 (V_S) と吃水 (d_F)

```
( STEP 0 KIND-OF-SHIP )
( STEP 0 REQUIRED-VS-KNOT )
( STEP 0 FULL-LOAD-DRAFT )
( STEP 0 REQUIRED-POWER )
( STEP 1 MAIN-ENGINE )
      .....
      .....
( STEP 4 DECIDE-MACHINERY )
( STEP 5 GENERATE-PLANT )
( STEP 6 SELECT-BEST-PLANT )
```

図4･12　設計プロセス管理のための述語

とから計算できることを示している．(b)は主機関の種類に関するもので，「船の種類がフェリーならば中速ディーゼル，船の種類がタンカーか LNG 船で，所要出力が 55000 PS 以上であればタービン，所要出力が 55000 PS 以下であれば低速ディーゼルとする」という「if ── , then ── 」型の条件式が手続きとして記述されている．(c)は所要回転数を満足する主機関の候補を検索するためのものであり，手続きとして検索処理が記述されており，その処理の方法としてメッセージの送信が記述されている．(d)は設計条件を満足する主機関候補の中で燃料消費率が低いベスト３の候補を求める処理に関するものである．

以上のような記述に対して，全体としての設計処理は汎用システムの機能により，ある設計処理のなかで必要となる設計処理を再帰的に呼び出すことによって行われる．なお，具体的な設計処理の内容については次節で詳述する．

4･4･5　設計プロセスの記述

以上の設計データや設計知識の記述に対して設計を進めていくためには，図4･3に示した設計過程に対応して，その進行を管理する必要がある．このような管理は，汎用システムの機能に従って，図4･12に示すような形式で設計過程の各段階で何を決定していくかを記述しておくことにより行う．このような記述に従って，前項で示した設計オブジェクトに対する働きかけが行われ，それ

に基づいて設計処理が行われていく．なお，図の例は，4･6･1項の船用動力プラントの設計に対するものであり，この記述に従って，まず，主機関の選定が行われ，続いて，プラント構成方式の決定・補機計画 ･･･ の順で，設計処理が行われていくようになる．

4･5　設計処理の方法

本節では，プラントの機器構成設計における処理について，プラント構成方式の決定・各機器の検索処理・プラント候補の合成処理・プラント候補の評価処理のそれぞれの方法を示す．

4･5･1　プラント構成方式の決定

プラントの構成方式はプラントの運用などに大きな影響を及ぼすため，個々の設計事例に対して，供給すべきエネルギーの種類や絶対量，それらの相対的な量的関係などにより有効な方式や不適切な方式が存在する．また，有効な方式においても，具体的にプラントを構成しようとする段階において，その方式における個々の機器が具体化するにつれて，個別の機器の組合せが不適切であったり，有効である可能性が低くなったりして，徐々に，考慮すべき方式は設計の進行とともに絞りこむことができる．例えば，船用動力プラントの場合には，4･4･4項でも述べたように，その主機関の種類は対象とする船舶の用途やサイズによって限定される．また，コージェネレーションプラントの場合には，4･5･3項のプラント候補の合成処理の項で詳述するようにして，徐々に構成方式を合理的なものに限定しつつ設計を進めていくことができる．

4･5･2　各種機器の検索

構成機器の選定は，あらかじめ用意されたカタログデータ中の機器を，エネルギー需要量や既に検索済の他の機器の性能に従って，検索を進めて絞りこん

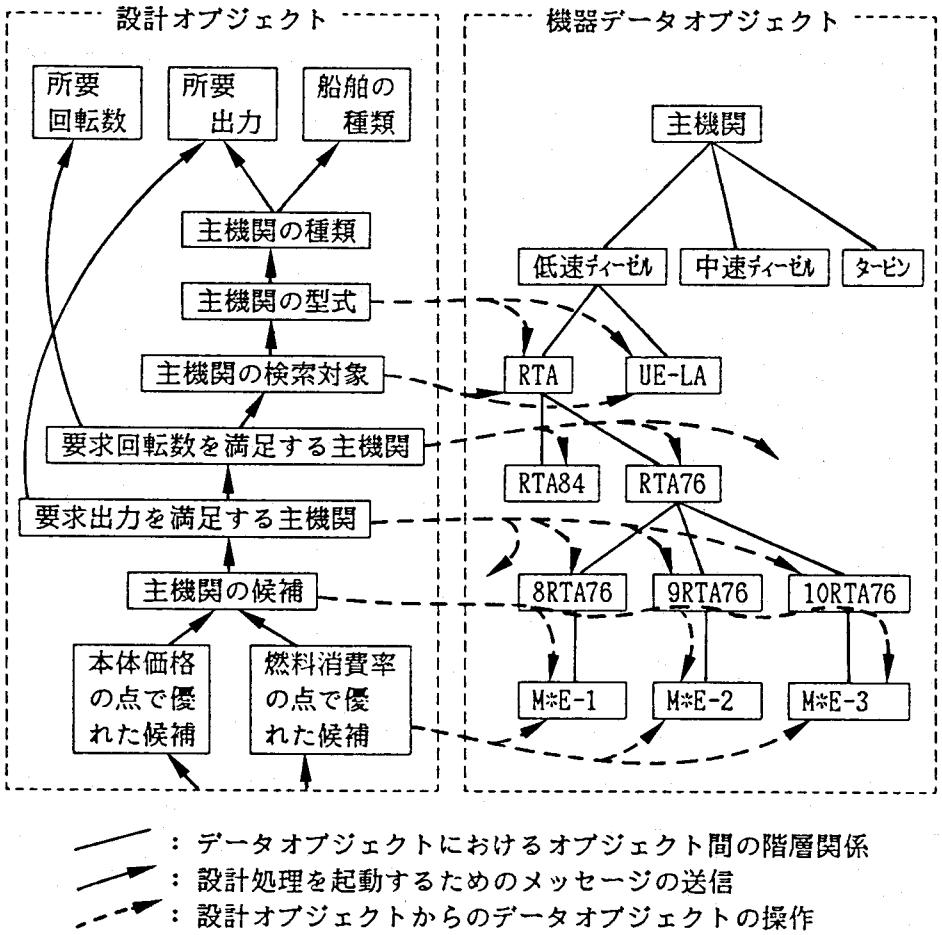


図4･13　各種機器の検索処理（船用動力プラントの主機関検索の場合）

でいく過程である．以下に，このような検索の過程を，4･6･1項で取り上げる船舶の主機関の選定過程をもとに示す．

船舶の主機関に関するカタログデータは，4･4･2項で示したように，オブジェクトとして階層的に表現される．このような表現に対して，機器の選定は図4･7に示したオブジェクトを上位から下位へと移りながら行うようにする．すなわち，種類，型式，･･･，候補の順に検索を進めていく．例えば，型式が RTA と決定されれば，次の検索対象はそのサブクラスである RTA84，RTA76，･･･となる．このような処理は，図4･7に示した機器データオブジェクトに対して，

表4・1　プラント構成方式の分類（コージェネレーションプラントの場合）

	記号	原動機の種類	原動機の型式	廃熱の形態	冷水供給熱源
I	T*1H	ガスタービン	—	高圧蒸気	高圧蒸気
II	EW1W	ガスエンジン	温水冷却型	温水	温水
III	EW2H	↑	↑	高圧蒸気／温水	高圧蒸気
IV	EW2W	↑	↑	↑	温水
V	EB1L	↑	沸騰冷却型	低圧蒸気	低圧蒸気
VI	EB2H	↑	↑	高圧／低圧蒸気	高圧蒸気
VII	EB2L	↑	↑	↑	低圧蒸気

図4・11に示した設計オブジェクトから，メッセージ・パッシングにより，様々な情報を参照したり，処理を起動したりして働きかけることにより行われる．図4・13はその様子を示したものであり，設計オブジェクトと機器データオブジェクトの間で次々とメッセージが送信されていく．

また，4・6・2項で取り上げるコージェネレーションプラントの場合は，このような処理に，さらに，需要量データオブジェクトが関連するが，同様の処理により各機器の選択が行われる．

4・5・3　プラント候補の合成処理

プラントの合成処理は，4・5・1項で示した構成方式に基づいて，4・5・2項に示した機器の検索処理を連動させて行う．しかし，一般にエネルギープラントの構成は複雑で，多数の機器から構成されるため，各構成方式に対して，単に可能な機器を組み合わせたのでは，いわゆる組合せの爆発を生じることになる．そこで，プラント候補を徐々に合成しながら，その性能を予測しつつ候補の絞りこみをはかって，設計を進めていくことにする．以下に，コージェネレーションプラントの場合を例にして，その手順を示す．

コージェネレーションプラントの構成方式には様々なものがあり，原動機の種類や型式，原動機廃熱の利用形態，エネルギー供給方式などによって，表4・

1のように，系統的に分類することができる．このような分類の各方式におけるプラントの運用方式や設計処理の方法は，その分類の系統に従って，おのおの異なったものとなり，さらに，原動機廃熱の利用方法や補機の構成を考慮すると，その分類はさらに多くのものとなる．そこで，このように複雑なプラントを合成していくにあたり，表4･1のような階層的なプラント構成方式の分類に着目することにする．すなわち，プラント候補の合成を，図4･14のようにして，原動機の選定，廃熱利用方式の決定，･･･ の順で進めていき，未検索の補機を含む仮想的なプラント候補を補機の選択に従って具体化していくことにより，最終的なプラント候補を合成するようにする．つまり，図4･9に示したプラント候補を表現するためのクラスの階層を，表4･1に対応させて，図4･14の上半部のように構成した上で，下半部のように，クラス階層における中間層の各クラスを仮想的な機器を含めたプラント候補をインスタンスとして表現するための枠組として用いる．その上で，具体的なプラント候補を最下層のクラスのインスタンスとして表現するようにする．これに基づいて，プラント候補に対応するインスタンスを徐々に分化させつつ，そのクラスをスーパークラスからサブクラスへと変更していくことにより候補を合成していくものとする．以下にその過程を具体的に示す．

コージェネレーションプラントに対するプラント候補の合成過程では，まず，プラントの中核となる原動機の検索を行う．それにより得られる候補は，図中の 'EW'，'EB' などの「原動機の型式」に対応したクラスの下の 'A群' のインスタンスとして取り扱い，その各オブジェクトにそれぞれの原動機の名称や台数を保持させる．次に，その原動機の廃熱をどのような形態で回収するかによって，'A群' のインスタンスを 'EB1'，'EB2' などの「廃熱の形態」に対応したクラスの下のインスタンスへと必要に応じて分化させて，'B群' のインスタンスを生成する．続いて，「冷水の供給熱源」を決定することにより，同様にして 'C群' のインスタンスを生成する．ここで，より詳細な設計処理を行う前に，次項で後述するような方法で，設置すべき補機類を仮想して各候補の性能を概算した上で，候補を 'C*群' へと絞りこむ．さらに，廃熱量の

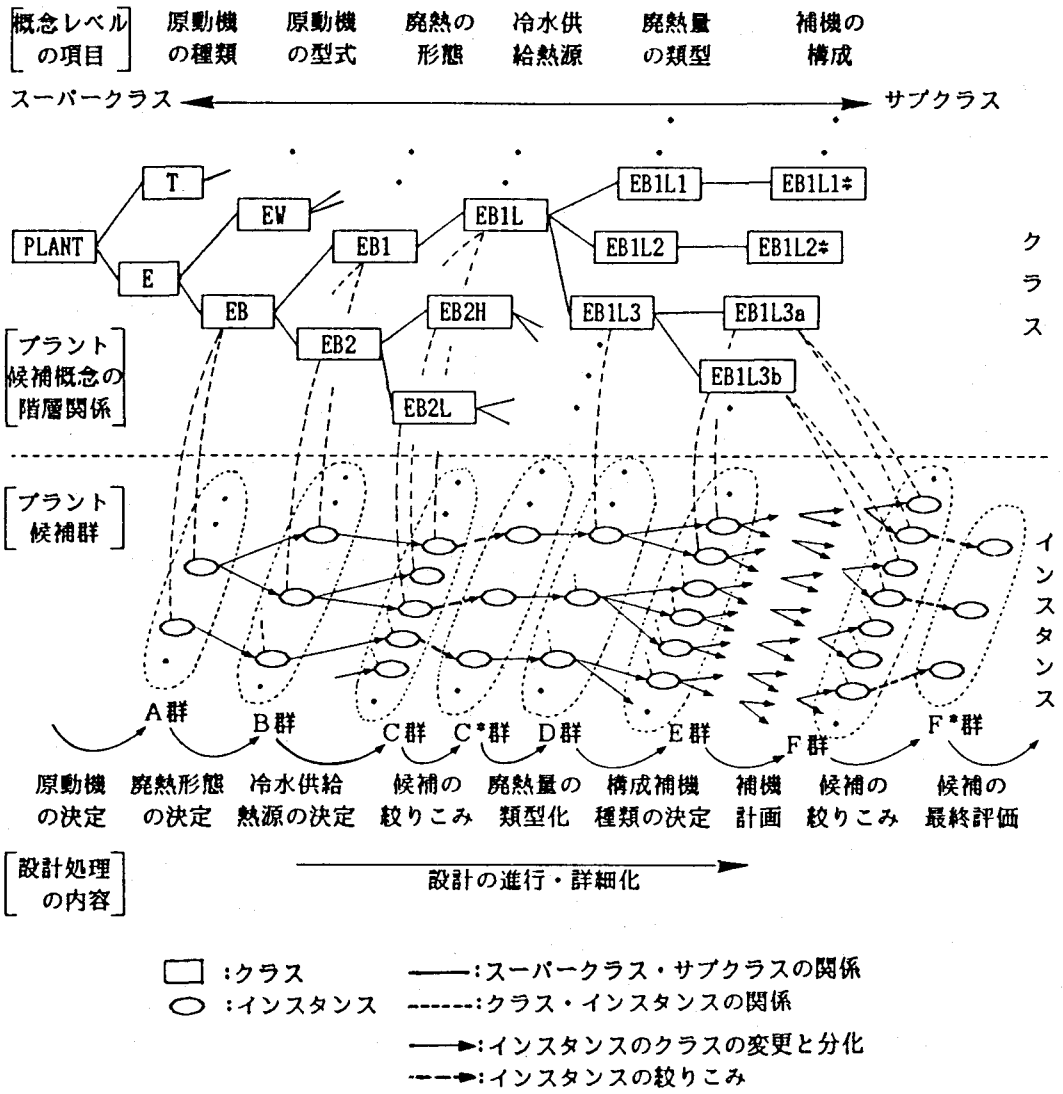


図4･14　プラント候補の合成過程（コージェネレーションプラントの場合）

類型化，補機構成種類の決定を行って，プラント候補を 'C*群' → 'D群' → 'E群' へと変更して分化させていく．そして，得られた 'E群' に対して補機計画を実行する．この過程では，各機器が順次検索されるに従って，その機器のサイズをもとにして次に検索する機器のサイズは限定するようにして，機器の検索とプラントの合成を進めていく．それにより，インスタンスを生成・

分化させていき，それぞれのインスタンスに各機器の名称と台数を保持させることにより，実際のプラントに対応する候補である 'F群' のインスタンスを生成して，プラントの候補を合成する．以上のような手順で，複雑なプラントの候補を合成するようにする．

4･5･4 プラント候補に対する評価処理

プラント候補に対する評価は，前項のように，候補合成の過程や候補合成後の各段階で行う必要がある．

エネルギープラントの経済性評価とその方法　エネルギープラントの経済性は，プラントの構成に係わる設備費（初期コスト）と，運用時における燃料費（運用コスト）によって定まる．前者は，プラントの具体的な構成が定まれば，構成機器のコストを合計することによって求めることができる．また，後者は，与えられたエネルギー需要に対してプラントの運用方策が定まれば，それに必要となる各種の燃料費を合計することによって求めることができる．一方，その評価の方法としては，プラントの償却期間を設定した上で，同一額の毎年経費に換算した初期コストと年間の運用コストの和により評価を行う「年間総経費」による評価や，プラントの利用年数によって初期コストと累計の運用コストの和がどのように変化するかを，プラント候補の間で比較する「簡易償却年数」による評価などがある[9]．これらの処理はいずれもそれぞれのプラント候補や機器に関した固有の処理であり，その算出方法はそれぞれのオブジェクトのクラスにメソッドとして記述しておく．

プラントの合成過程における候補の評価　前項で示したプラントの合成過程における評価では，原動機のみが定まっている状態で，原動機から生じる廃熱の見込み量やエネルギー需要量の分析などに基づいて，残りの補機の能力を仮定して候補の評価を行う必要がある．コージェネレーションプラントの場合には，このような評価を以下のようにして行う．すなわち，まず，各種の供給すべき熱エネルギーを一元的にエネルギー量ベースで考えた上で，原動機の運転負荷率を決定し，その廃熱をあらかじめ定められた順序で各種の補機に割り

当てて熱エネルギーを供給するものとする．このとき，各機器のサイズは任意のものが連続的に存在するものとし，また，各種補機の設備コストもそのサイズの関数として得られるものとする．そして，そのような仮想的な機器を含んだプラントに対して，運用計画なども考慮した上で評価を行う．このような評価に基づいて，前出の図4･14における 'C群' から 'C*群' への絞りこみを行う．

プラント候補の評価と運用方策　　具体的なプラント候補が定まり，それらに対する評価を行うためには，前述のように，最適な運用方策を算定する必要がある．最適な運用方策の決定は，エネルギー需要量やプラントを構成する機器の特性を考慮しながら，年間の総燃料費が最小になる運用方策を決定する問題であり，各機器の運転・停止を考慮する必要があるため，0-1 整数変数を含む混合整数計画問題として定式化される[5]．しかし，この種の最適化計算は組合せ数の著しい増加を伴うため，得られた全てのプラント候補に対してこの計算を行うことは困難である．そこで，例えば，コージェネレーションプラントの場合には，以下のような２段階の処理により評価を行う．

(1) すべてのプラント候補に対して，「まず，需要量の変動に対応して原動機の負荷を定め，その負荷において生じる廃熱をあらかじめ定められた優先順位で各エネルギー需要に振り分ける」方法（熱追従方式／電力追従方式）[10] により運用方策を定めて，経済性を評価する．

(2) (1)の評価に基づいて優れたプラント候補を絞りこみ，その候補に対して数理計画法を適用して最適な運用方策の立案を行い，それをもとにして最終的な経済性の評価を行って，プラントを選定する．

これにより，効率的かつ合理的に適切なプラントを選定することができるようになる．このような処理もすべてオブジェクトのメソッドとして記述する．例えば，数理計画法による最適化計算は，図4･15のようにして，メソッドの処理の中から，混合整数計画問題を解く FORTRAN プログラムを呼び出すことにより行う．すなわち，プラント候補に対応するオブジェクトのそれぞれに対して，最適な年間総経費を決定するためのメソッドを起動するメッセージを送信し

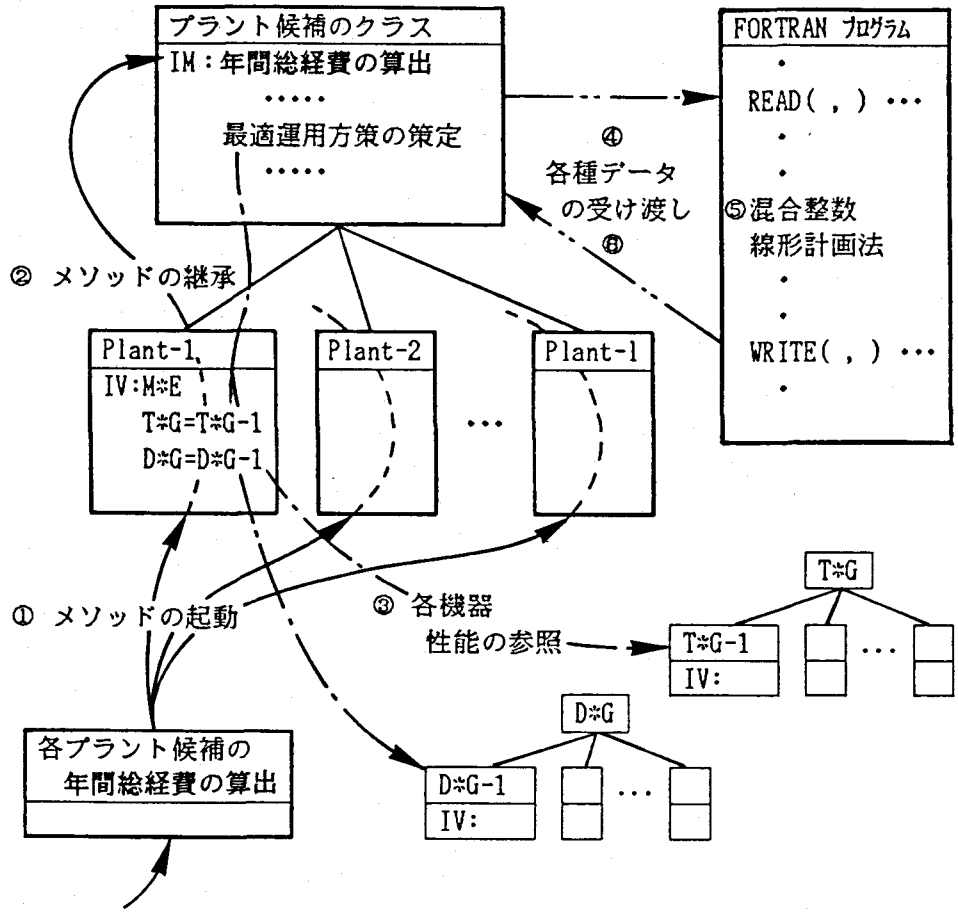


図4･15　最適化処理の実行過程

(①)，それぞれのオブジェクトにおいて，クラスから継承されるメソッド(②)が最適運用方策を決定するためのメソッドを起動する．さらに，変数として記述された各機器の名称を通じ，それぞれの特性を参照し(③)，そのデータをFORTRAN プログラムに引き渡した(④)上で，混合整数計画問題を解き(⑤)，その計算結果を受け取る(⑥)ことにより行う．

以上が，本手法における各種設計知識やデータの表現，ならびに設計処理の方法である．

4･6　事 例

最後に，システムの実行例として，舶用動力プラントならびにコージェネレーションプラントの機器構成設計を行った例を示す．

4･6･1　舶用動力プラント設計への適用

舶用動力プラントは船を推進するための主機関と熱や電力を供給するための補機から構成される．このうち，主機関の選定は推進性能に関する面から行われ，それに続いて，熱・電力供給プラントの設計が行われる．図4･16は，本システムによる舶用動力プラント設計の実行例である．図中(a)は，主機関の選定を行っている過程を示したものであり，まず，設計条件である船の要目を把握した上で，条件を満たす９種類の主機関候補を検索して，それぞれ燃費，初期価格，総経費のベスト３の候補を提示した後，'C-M※E-2' を主機関に決定している．図中(b)は，熱・電力供給プラントの設計過程を示したものであり，まず，設計条件である電力と蒸気の需要量を把握した上で，構成機器種類を決定し，それらの機器を組み合わせてプラント候補を合成する．そして，初期コストと総経費の観点から候補を評価して，最適なプラントの選定している．最後に，選定された機器要目と概略レイアウトが図示される．

4･6･2　コージェネレーションプラントの設計への適用

コージェネレーションプラントは，ある地域や建物における各種のエネルギー需要を総合的に賄うことにより，エネルギーの有効利用，すなわち，エネルギー利用における総合効率の向上をねらったものであり，その設計計画は，複雑なプラント構成（図4･2参照）や多様なエネルギー需要（図4･1参照）によって，複雑なものとなる．図4･17は，本システムによるコージェネレーションプラントの設計過程を示したものである．この事例は，延床面積 20000 m^2 をもつあるホテルに対するものである．まず，設計対象におけるエネルギー需要量が設

主機関の検索条件

```
KIND-OF-SHIP ..... CONTAIN
REQUIRED-SERVICE-SPEED ..... 20.0000
FULL-LOAD-DRAFT ..... 11.6000
APPROXIMATE-PS-MCO ..... 22000.0
```

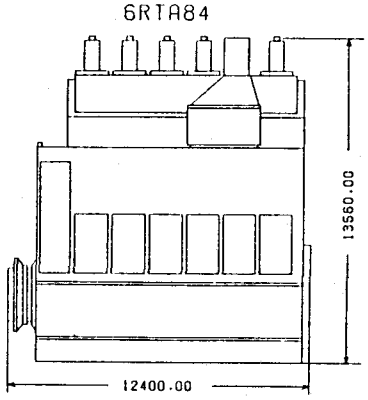
主機関の候補の検索

```
>>> OK! I FOUND CANDIDATES OF MAIN-ENGINE.
    9 ENGINES SATISFY THE REQUIREMENT.
********CANDIDATES-OF-MAIN-ENGINE********
       C-M*E-1 ----- 5RTA84
       C-M*E-2 ----- 6RTA84
       C-M*E-3 ----- 7RTA84
       C-M*E-4 ----- 8RTA84
       C-M*E-5 ----- 7RTA76
       C-M*E-6 ----- 8RTA76
       C-M*E-7 ----- 9RTA76
       C-M*E-8 ----- 10RTA76
       C-M*E-9 ----- 9RTA68
*****************************************
```

最適な主機関の選定

```
>>> I RECOMMEND NEXT CANDIDATES FROM THE POINT OF FUEL-CONSUMPTION
          C-M*E-4    8RTA84    120.4606          [ GRAM/PS/HOUR ]
          C-M*E-3    7RTA84    121.1673
          C-M*E-8    10RTA76   121.4716
>>> I RECOMMEND NEXT CANDIDATES FROM THE POINT OF PRICE [ YEN ]
          C-M*E-1    5RTA84    637000000
          C-M*E-5    7RTA76    721280000
          C-M*E-9    9RTA68    743400000
>>> I RECOMMEND NEXT CANDIDATES FROM THE POINT OF TOTAL-COST [ YEN ]
          C-M*E-2    6RTA84    775632860
          C-M*E-1    5RTA84    777522270
          C-M*E-5    7RTA76    784915080
```

選定された主機関の要目の表示




```
APPROXIMATE-PS-MCO ..... 22000.0
KIND-OF-MAIN-ENGINE ..... 6RTA84
PS ..... 22000.0
N ..... 89.9840
PS-MCO ..... 27300.0
N-MCO ..... 90.0000
LENGTH-OF-MAIN-ENGINE ... 12.3999
FUEL-CONSUMPTION ..... 122.398
```

（a） 主機関の選定過程

図4･16 舶用動力プラントの設計事例（1）

設計条件

```
ELECTRIC-DEMAND1 ..... 1200.00
ELECTRIC-DEMAND2 ..... 1060.00
STEAM-DEMAND1 ........ 1150.00
STEAM-DEMAND2 ........ 1100.00
```

プラントの構成機器の種類

```
>>> I SELECTED THE KIND OF PLANT COMPONENTS.
>>> IN THIS CASE, THE KINDS OF MACHINERY IS
************ KIND-OF-MACHINERY ************
      #1  ..  EXHAUST-GAS-ECNOMIZER
      #2  ..  TURBO-GENERATOR
      #3  ..  AUXILIARY-BOILER
      #4  ..  SHAFT-GENERATOR
      #5  ..  DIESEL-GENERATOR
*****************************************
```

プラント候補の合成

```
>>> 108 COMBINATIONS ARE GENERATED.
```

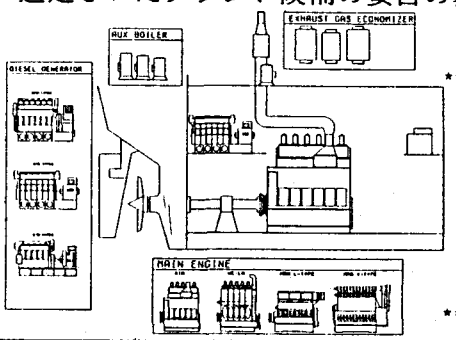
```
************ TABLE-106 ************
TURBO-GENERATOR  -- (T*G-TYPE1 * 1)
DIESEL-GENERATOR -- (D*G-TYPE3 * 1)
AUXILIARY-BOILER -- (A*B-TYPE1 * 1)
SHAFT-GENERATOR  -- (S*G-TYPE1 * 1)
E/E-POWER        --   4800.0 [KW]
************ TABLE-97 ************
(T*G-TYPE1 * 1)
(D*G-TYPE3 * 1)
(A*B-TYPE2 * 1)
(S*G-TYPE1 * 1)
  4800.0 [KW]
**********
************ TABLE-26 ************
TYPE3 * 1)
TYPE3 * 1)
TYPE1 * 1)
TYPE1 * 1)
00.0 [KW]
**********
```

最適なプラント候補の選定

```
>>> I RECOMMEND NEXT PLANT CANDIDATES
FROM THE POINT OF INITIAL-TOTAL-COST.
********* PLANT CANDIDATES **********
    #1    TABLE-106    929341310
    #2    TABLE-97     932438990
    #3    TABLE-25     935257890
    #4    TABLE-88     937286210
    #5    TABLE-16     938355570
    #6    TABLE-52     940819700
    #7    TABLE-103    941331920
    #8    TABLE-7      943202780
    #9    TABLE-43     943917380
    #10   TABLE-108    946445600
*************************************
```

```
>>> I RECOMMEND NEXT PLANT CANDIDATES
FROM THE POINT OF TOTAL-COST.
********* PLANT CANDIDATES **********
    #1    TABLE-106    981551830
    #2    TABLE-97     984649510
    #3    TABLE-88     990116950
    #4    TABLE-25     990365940
    #5    TABLE-16     993463620
    #6    TABLE-103    993542430
    #7    TABLE-52     997029700
    #8    TABLE-7      998310840
    #9    TABLE-108    999106920
    #10   TABLE-43    1000127400
*************************************
```

選定されたプラント候補の要目の表示




```
********** SELECTED PLANT ***********
  PLANT-NO           --  TABLE-106
  MAIN-ENGINE        --  6RTA84
  USED-POWER         --  22000.0
  REQUIRED-POWER     --  22000.0
  TURBO-GENERATOR    --  T*G-TYPE1
  DIESEL-GENERATOR   --  D*G-TYPE3
  SHAFT-GENERATOR    --  S*G-CANDIDATE1
  AUXILIARY-BOILER   --  A*B-TYPE1
  F*E-OUTPUT         --  4800.00 [KW]
  PLANT-TOTAL-COST   --  981551830
*************************************
```

（b） 舶用熱・電力供給プラントの設計過程

図4・16 舶用動力プラントの設計事例（2）

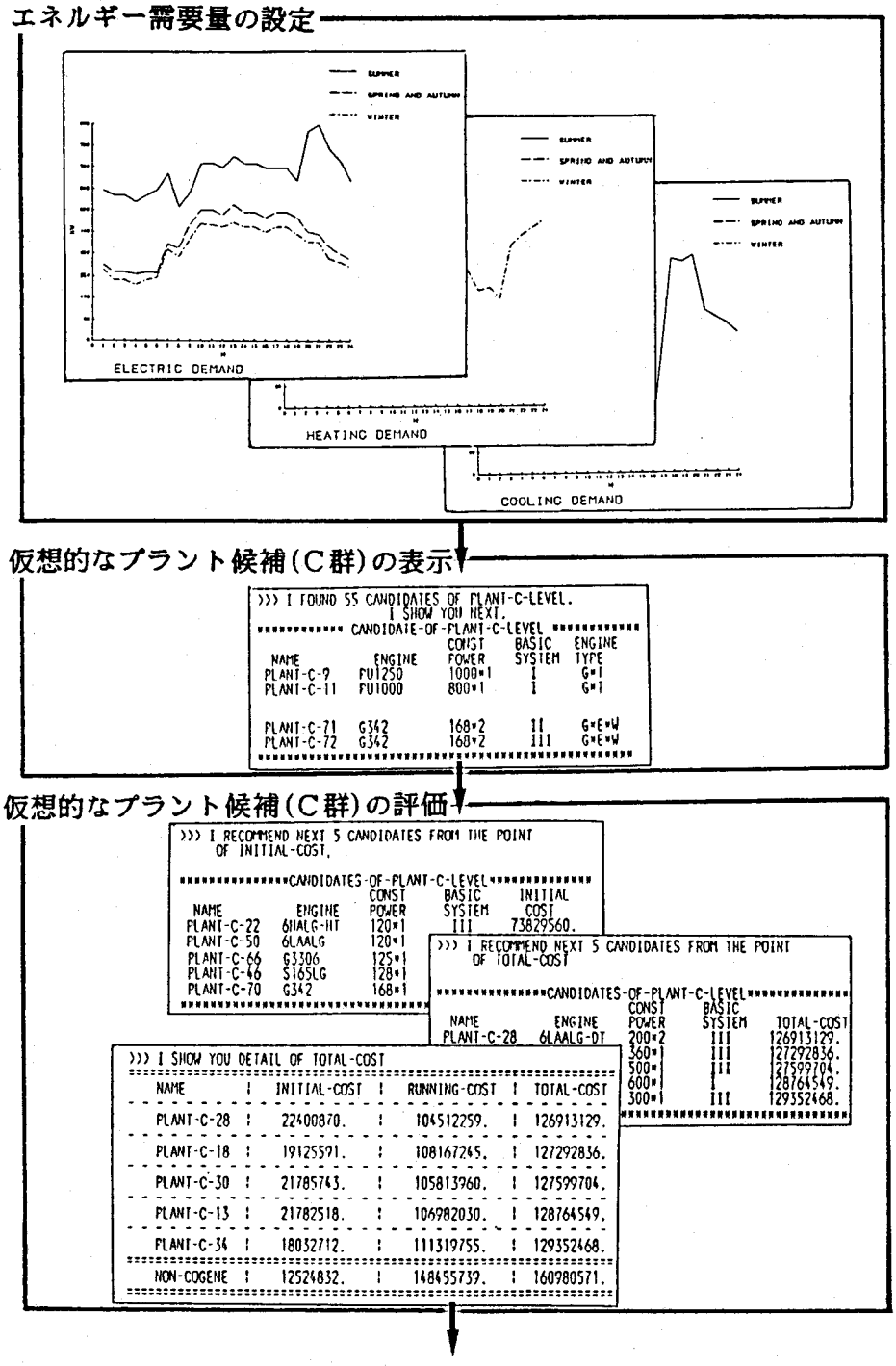


図4･17　コージェネレーションプラントの設計事例（1）

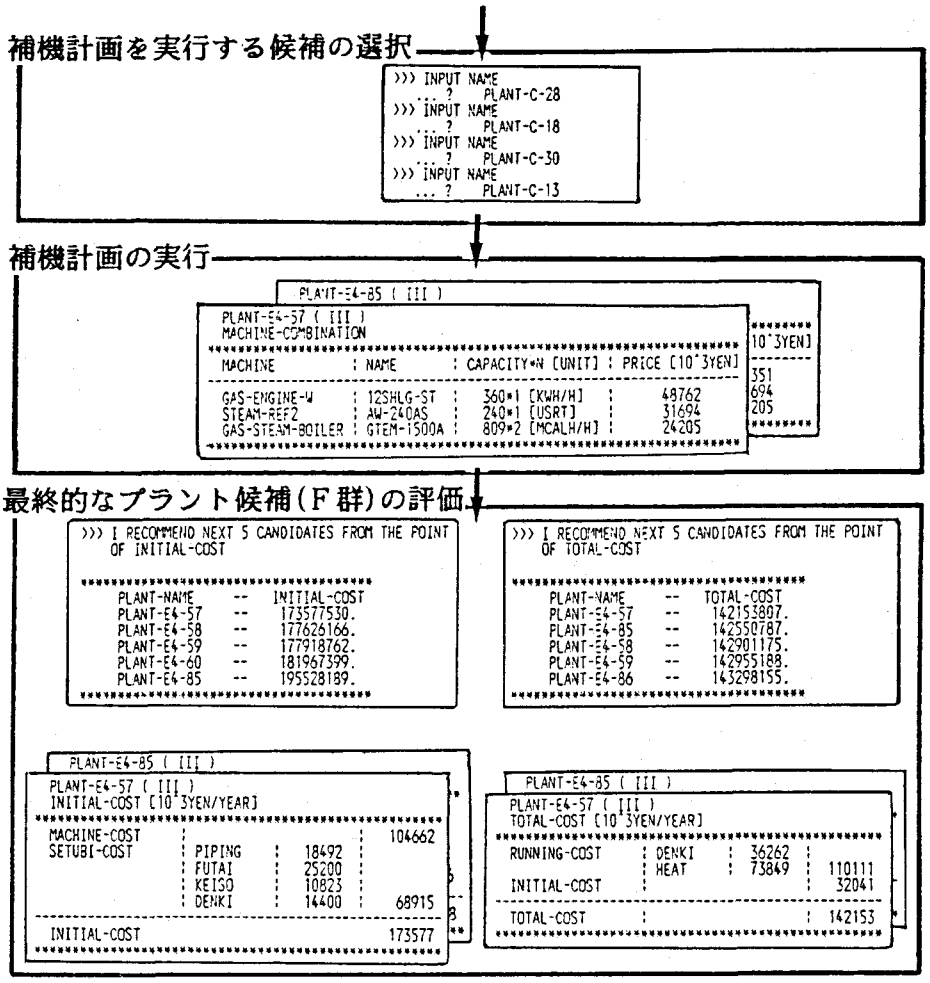


図4･17　コージェネレーションプラントの設計事例（2）

定され，その後，原動機候補の出力に関する検索範囲が設定される．続いて，それに基づいて，55台の原動機が選定されて，それぞれに対応する仮想的なプラント候補（前出図4･14の 'C群' の候補）が合成され，初期コスト，償却年数，年間総経費による評価が行われる．その中から，評価結果に基づいて，4種類の仮想的なプラント候補が選択され，さらに，具体的な補機計画が行われていく．そして，86個のプラント候補が生成され，各種の評価を行った上で，最終的に，'PLANT-E4-57' が適切なプラントとして選定される．

4･7 結 言

本章では，エネルギープラントの機器構成設計に対して，オブジェクト指向によるデータ表現を基本とした設計支援手法を提案し，舶用動力プラント，ならびにコージェネレーションプラントを事例として取り上げ，その手法に基づいて構築した設計支援システムに関して，その設計処理の方法や各種データの表現方法を示した．従来のプラント設計においては，過去の設計事例などを参照するなどして数ケースの設計案を検証するに過ぎなかったが，本手法を適用することにより，合理的に有効なプラント候補を検討しつつ設計を進めていくことができるようになる．また，各設計支援システムは，機器構成設計における，エネルギー需要量，各種の機器，プラント候補などのデータをオブジェクトとして表現することにより高いモジュール性を有しており，データの入れ換えなども容易な柔軟性の高いシステムとなっている．

このようなオブジェクト指向の様々な性質を有効に利用した「知的検索形」の問題に対する設計支援手法は，プラント設計のみならず，種々のシステム設計に対しても有効であると考えられる．

文 献

(1) 例えば，日本コージェネレーション研究会，コージェネレーション，vol.1，no.1，(1986)．

(2) 赤木ほか4名，ＡＩ技術を応用した舶用動力プラントのエキスパートＣＡＤシステムの研究，日本機械学会論文集 Ｃ編，vol.53，no.486，(1987)，pp.512-517.

(3) 赤木ほか4名，ＡＩ技術を応用した舶用動力プラントのエキスパートＣＡＤシステムの研究（第２報），日本機械学会論文集 Ｃ編，vol.53，no.491，(1987)，pp.1622-1628.

(4) 本論文，第２章の文献(18)~(20)．

(5) 赤木・横山・伊東，混合整数線形計画法に基づくＬＮＧ船機関部システ

ムの最適計画，日本機械学会論文集 C編，vol.52，no.476，(1986)，pp.1469-1476.
(6) UTILISP説明書，(1986)，日本電気.
(7) 例えば，Winston，P. H. and Horn，B. K. P.，LISP，(1981)，Addison-Wesley，(邦訳，白井・安部，LISP，(1982)，培風館).
(8) 三菱，スルザーＲＴＡ機関カタログ (H 400-0390)，(1985)，三菱重工.
(9) 例えば，赤木，エンジニアリングシステム設計工学，(1982)，共立出版.
(10) 牧村，コージェネレーションシステムとは，コージェネレーションセミナー資料集 1-(1)，(1986)，日本コージェネレーション研究会.

第5章　制約指向に基づく基本配置設計支援システム

5・1　緒　言

各種の発電設備をはじめとする大型のプラント設計においては，プラントを構成する各種構成機器の性能を発揮させ，かつ空間的な制約を満たすように，それらの位置関係を定めるいわゆる配置設計，なかでも基本配置設計が高い比率を占めている．このような設計は，設計に要する手間が非常に大きいことに加えて，熟練設計者への依存度が高いため，コンピュータの援用による省力設計への期待が大きい．一方，配置設計においても，知識情報処理技術を援用して設計専門家に代わる設計支援システムを開発しようとする試みがいくつか行われており，McDermott らによるコンピュータ部品の筐体への組み込みを行うＲ１[(1)]のほか，プラント設計や設備設計などを対象にして，いくつかのシステム[(2)(3)]が構築されている．しかし，これらの試みのほとんどは設計者の有する知識を表層的に単純なルールとして表現することに基本を置いているため，個別性が強く，保守・管理が容易ではないなどの点で，なお十分な成果が得られていないように思われる．これを克服するためには，基本配置設計という設計タスクの本質に基づいた汎用的な支援手法の確立が必要である．

本章では，配置設計の過程を設計対象物を構成する要素間の空間的な制約条件を満足するようにそれらの位置を定める過程として理解した上で，知識情報処理技術の一つである「制約指向プログラミング[(4)(5)]」に基礎を置いた配置設計に対する方法論を提案し，それに従って，求められる空間的な制約などの設計要求と，それを満足する設計解を得るための設計機構とを明確に分離することを基本として，前者の宣言的な記述と後者の一般的で汎用性のある機能から構築した設計支援システムについて示す．本手法では，基本配置設計におけるタスクの抽出や各種の制約条件の種類や性質についての検討に基づいており，

基本的な手法として，基本配置設計における状態量が有限領域に限定できることから値域の概念[6]による制約指向を採用し，さらに，各種の制約を有効に処理するためにいわゆる生成検査法を用いる．このような手法によって，汎用的で拡張性に優れたシステムの構築が可能となる．また，本システムは，配置問題の構成についての検討に基づいて，設計対象物の構成要素や配置制約などがネットワーク状の関係として整理することができる点に注目して，「オブジェクト指向プログラミング[7]」のもとで構築されている．最後に，構築したシステムを原子力発電所の基本配置設計に適用した事例を示し，本システムの有効性を検証する．

5・2　配置設計の特質と支援手法

5・2・1　配置設計の問題点と従来の支援手法

配置設計における問題点は，前述のように設計に要する手間が非常に大きく，熟練設計者への依存度が高い点にあり，これに加えて，設計処理の内容が明確に示しにくいことも問題点の一つである．このため，設計結果は単に「図面」として表現されるのみで処理内容も明らかでなく，得られた結果に客観的な評価を下すことも容易ではない．また，従来のコンピュータの援用についても，設計結果の単なる作図に止どまり，設計本来の処理や評価については専ら設計者に依存している．

配置設計における方法論についても，問題がごく小規模で単純な場合には，数理計画法を適用することも可能である[8]が，現実の問題は複雑で規模も大きいことから，実際に適用することが困難である場合が多い．このため，近似解法としてヒューリステックスな解法が有効とされており，知識情報処理技術のひとつであるエキスパートシステムの手法を用いて，熟練設計者の有する経験的な配置知識を「if ―― ，then ―― 」という形式で表現することにより，実際の問題を扱おうとするシステムが構築されつつある[1]~[3]．しかし，この

ような手法は，その解法が個々の配置問題に大きく依存し，設計者の有する表層的な知識をもとにしているため，設計対象が変わったり，新たな要素が加わったりした場合には，そのままでは対応することができず，一般性や拡張性などの点で決して優れたものとなってはいない．これは，配置設計の根源に潜む設計処理の内容が十分把握されていないことに原因があり，優れたシステムを構築するためには，配置処理の内容を本質から理解することが必要である．

5・2・2　制約指向と配置設計

本章では，以上のような問題点を解決する手法として，緒言でも述べたように制約指向[4][5]に基づいた手法を提案する．制約指向の考え方は，与えられた問題に対して，具体的な手続きを直接記述して解を求めるのではなく，解に求められる要求すなわち「制約」を記述し，それを問題の状況に応じて双方向に展開することにより具体的な手続きを定めて解を得ようとするものであり，より一般的で適用性に優れた問題解決を目指したものである[5]．前項で述べた配置設計におけるヒューリステックスな解法の問題点は，配置手続きが直接記述されていることに起因しており，制約指向はこれを克服する有効な方法論となり得る．つまり，設計者の有する経験的な配置知識の背後には，配置結果に求められる具体的な設計要求に対応する制約と，それを配置結果において実現するための抽象的な戦略とが存在し，設計者は具体的な問題に対して，それらをもとに全体的な配置を見通して具体的な配置知識を生成しつつ解を得ているものと認識すれば，その抽象的な戦略を抽出してシステム化することにより，汎用的な配置手法を構築することができるはずである．また，このような手法を構築することは，制約という観点から汎用的な配置処理のアルゴリズムを記述することであり，配置制約と設計機構とから構成される一般的で拡張性に優れたシステムを構築するための礎となり，加えて，設計処理の内容が制約によって明示的かつ客観的に記述・評価できることが期待できる．一方，制約指向については，それに基づいた設計支援システム[9]も開発されてはいるものの，適用されてきた対象は，Constraints[10]で扱われた代数関係や ThingLab[11]

で扱われた単純な幾何学的関係に限られており，複雑な配置設計問題を扱うためには，問題の構成や性質について十分吟味した上で，新たな視点から配置手法を構成する必要がある．次節では，このような点から対象とする基本配置設計の過程について述べる．

5・3　基本配置設計の過程

5・3・1　配置設計問題

配置設計についての議論を進めていくにあたり，その過程で扱われる対象物を分類し，以下の用語を定義しておく．

(1) 配置要素：配置すべき設計対象物の構成要素．

(2) 配置空間：(1)の要素を配置していく空間の全体や部分．

(3) 配置制約：上記の要素や空間の間で満たされるべき空間的な位置関係に対する条件．

例えば，事例として取り上げる原子力発電所の配置設計においては，(1)にはプラントを構成するポンプやタンクなどの各種機器や特定の部屋が，(2)にはプラントを設置する建屋やその特定の床面といった部分などが対応する．(3)には「Ｖタンクは Ｆポンプよりも上階に設置する」，「Ｒ冷却器は Ｒポンプの真上に設置する」，「Ｐ冷却器は Ｐに近い方が良い」などといった要素間の位置関係が対応する．以上の用語を用いると，配置設計は「配置制約を満足するように配置要素を配置空間内に割り当てていく」という問題に帰着する．

5・3・2　基本配置設計の過程と単位格子の概念

配置設計に係わる設計の過程は図5・1のような構成になっており，配置設計過程の前には設計対象物の構成を定める過程が，後には配管設計や建設設計などの設計過程が位置する．それらの中で，配置設計そのものの過程を以下の二つの過程に分割して考え，本章で提案する手法は，その前者を対象とする．

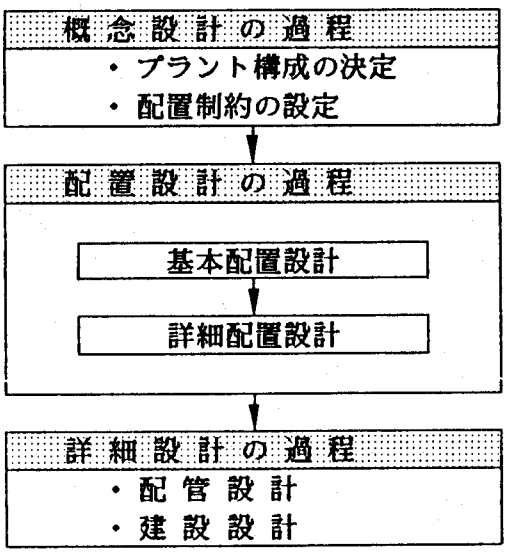


図5・1　配置設計過程の位置付け

(1) 基本配置設計の過程：各種の空間的な位置関係が満足されることを保証する程度の各要素間のトポロジカルな位置関係を決定する過程.

(2) 詳細配置設計の過程：(1)の結果に基づいて，各要素の空間内の位置寸法を厳密に決定する過程.

両者の本質的な違いはその過程で扱われる情報の種類にあり，前者においては非数値的な情報が主に扱われ，後者においてはもっぱら数値的な処理が中心となる．本手法では，上記の基本配置設計の過程において，要素間のトポロジカルな位置関係を効率的に扱うために，配置空間内において「単位格子」の概念を導入する．つまり，個々の配置問題におけるその特質や各種配置要素の標準的なサイズに従って配置空間を一定の大きさの単位格子に分割し，その大きさのもとで各要素の大きさや位置を議論するようにする（図5・2)．このような取扱いは船舶設計における "フレームスペース（補強材の間隔)"，原子力発電所の設計における "通り芯（建屋の壁の位置)" に対する考慮などに見られる．これにより，基本配置設計は「各配置要素に対して，その形状に見合う単位格

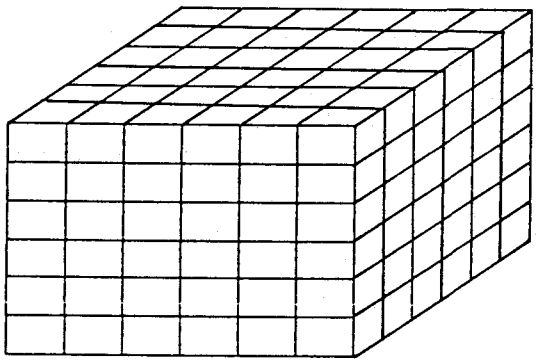


図5・2　配置空間の単位格子

子の組合せを配置制約に従って割当てていく」というある種の探索問題としてとらえることができる．

5・4　制約指向プログラミングによる基本アプローチ

5・4・1　配置設計手法の基本構成

制約指向を基本とする本手法の基本的な構成を図5・3に示す．本手法は，(1)：汎用的な配置設計機構と，(2)：5・3・1項で列挙した配置制約をはじめとする問題構成要素の表現とから構成され，前者が後者を動作領域として配置処理を行うようにする．このように設計機能と設計条件とを明確に分離することによって，一度前者を構築すれば，ユーザは複雑なプログラミングから解放され，単に制約を記述するだけで個々の配置問題を解くことができるようになる．また，後者の宣言的な制約の記述は，配置における要求を明確かつ客観的に記述したドキュメントともなる．このような構成において配置機構すなわち制約解消システムを構築するためには，前述のように，配置問題における対象の状態を表現する値の取り得る範囲や，扱おうとする制約の種類や性質を十分検討しておく必要がある．以下では，このような点から配置設計の特徴を示す．

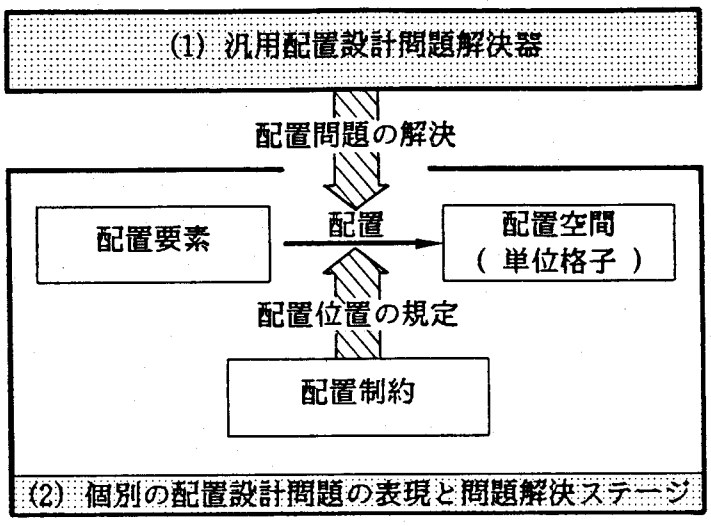


図5・3　配置設計システムの基本構成

5・4・2　配置設計における状態量

配置設計における問題の状態は各配置要素が空間内において占める位置に対応し，5・3・2項で述べたように，その位置は単位格子の組合せとして表現される．したがって，有限の単位格子を組み合わせた要素の位置は高々有限個の数に限定される．また，配置制約はそのような有限個の位置の中からただ一つの位置を選択するものではなく，順次その数を限定していくものであり，配置設計の過程は可能な組合せを制約によって次々と絞り込んでいく過程となる．これらの点で，配置設計における制約問題は従来の制約指向システム[10][11]が扱ったものとは異なり，基本配置設計においては値域の概念に基づいた手法[6]が基本的には有効となる．この手法は，各状態量の取り得る範囲が有限であることに着目して，制約に従ってその範囲を順次狭めていくことにより，最終的に各状態量の値を確定しようとするものである．

5・4・3　配置制約の階層的分類

本手法におけるポイントの一つは，前述のように配置制約の設定にあり，このため，配置を定めている制約を可能な限り一般化し，さらに配置処理に関す

る取扱いに従って分類することが必要である．ここでは，種々の配置制約を洗い出して約20種類の基本的な制約を設定する．図5･4は，それらの階層的な分類と制約の一例を示したものであり，後述する配置処理の面から各種の制約は以下の４種類に大別することができる．

(1) 領域制約：配置要素に対して，組合せの対象となり得る単位格子を限定する義務的な制約．

(2) 検査制約：生成された配置候補を検査することにより，その制約が満たされているかどうかを判断できるような義務的な制約．

(3) 選択制約：上記の義務的な制約をすべて満足させた上で，さらに配置候補を優先付ける示唆的な制約．

(4) 大域制約：個々の配置要素や特定の配置空間の間で取り扱うことができない，全体の配置状況に係わる義務的な制約．

これらの制約の内容は様々であるが，さらに大きく義務的なものと示唆的なものに分けることができ，前者は設計対象物の物理的な条件や安全性などにより必要不可決な条件に対応し，後者は経済性や操作性の点から希求される条件を反映している．また，配置要素の形状や大きさについても，配置における単位格子の組合せに対して義務的に係わるため，広義の制約としてとらえることもできる．以上のように，配置設計で扱われる制約は複雑であり，本手法では，これに対応するために，前述の値域の概念に基づいた制約指向[6]に，いわゆる生成検査法を組み合わせる．次節では，このような本手法の詳細なアルゴリズムについて示す．

5･5　配置アルゴリズム

5･5･1　配置設計の探索木と配置戦略

前述のように，基本配置設計は各配置要素に対して単位格子の組合せを求めていく問題であり，ある種の探索問題として扱うことができる．その探索木は

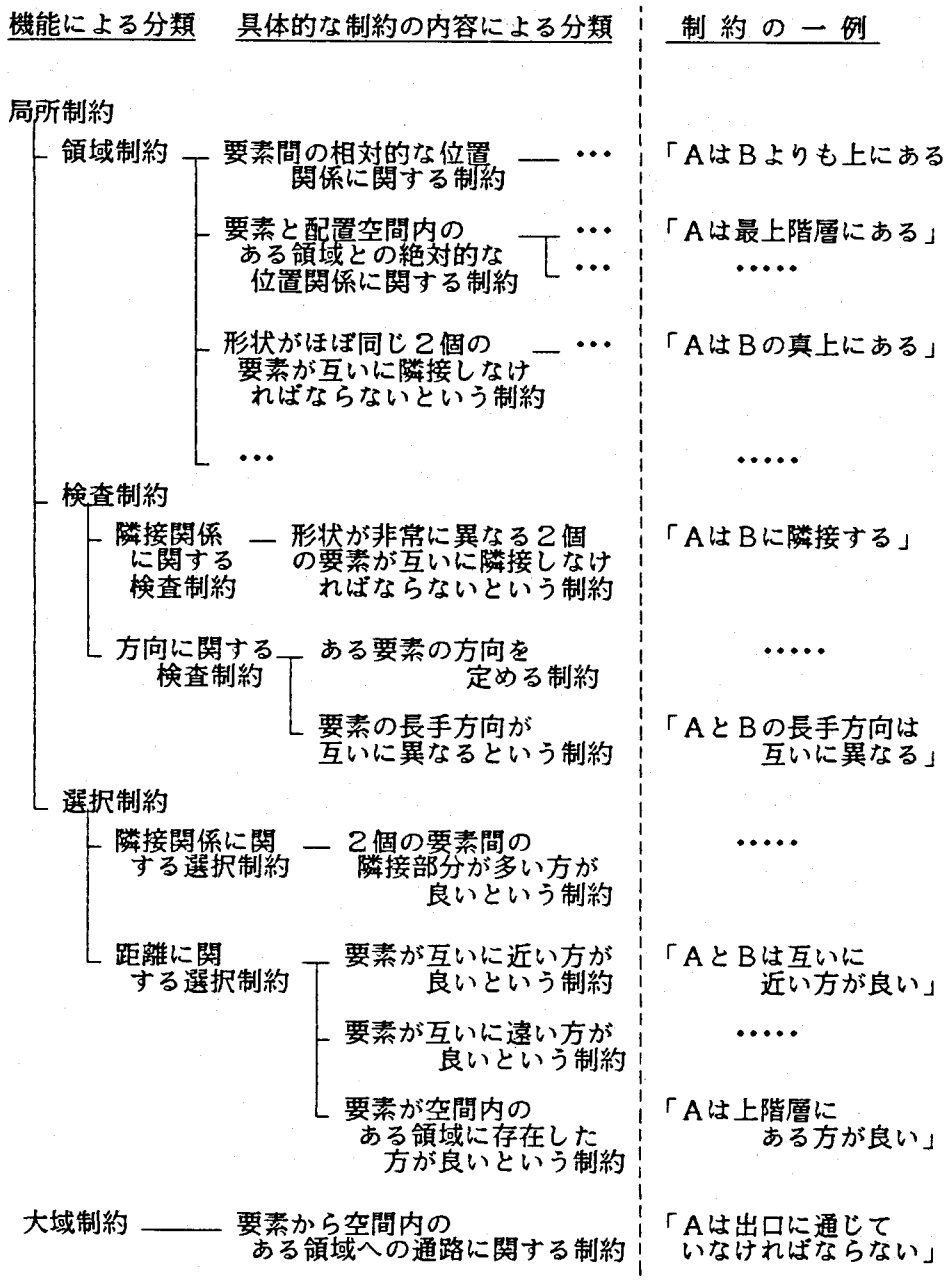

機能による分類
具体的な制約の内容による分類
制約の一例
局所制約
領域制約
要素間の相対的な位置関係に関する制約
「AはBよりも上にある」
要素と配置空間内のある領域との絶対的な位置関係に関する制約
「Aは最上階層にある」
形状がほぼ同じ2個の要素が互いに隣接しなければならないという制約
「AはBの真上にある」
検査制約
隣接関係に関する検査制約
形状が非常に異なる2個の要素が互いに隣接しなければならないという制約
「AはBに隣接する」
方向に関する検査制約
ある要素の方向を定める制約
要素の長手方向が互いに異なるという制約
「AとBの長手方向は互いに異なる」
選択制約
隣接関係に関する選択制約
2個の要素間の隣接部分が多い方が良いという制約
距離に関する選択制約
要素が互いに近い方が良いという制約
「AとBは互いに近い方が良い」
要素が互いに遠い方が良いという制約
要素が空間内のある領域に存在した方が良いという制約
「Aは上階層にある方が良い」
大域制約
要素から空間内のある領域への通路に関する制約
「Aは出口に通じていなければならない」

図5・4　配置制約の分類

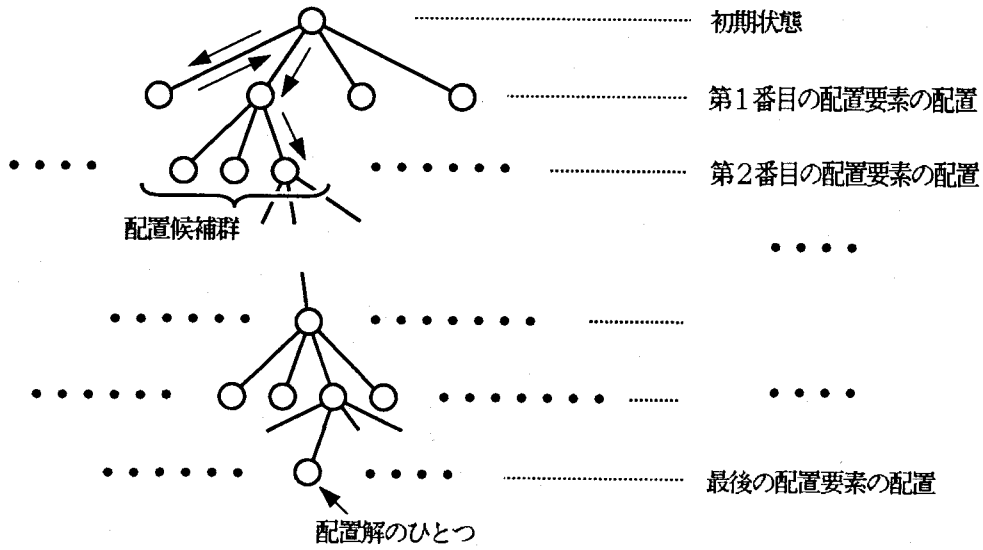
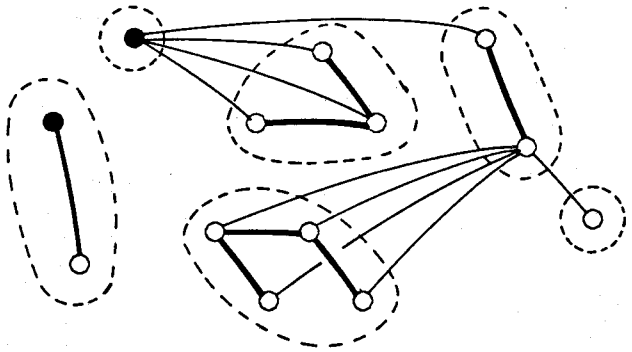


図5･5　配置設計の探索木

図5･5のように縦方向に各配置要素に対する配置処理，横方向に実行可能な配置候補（すなわち配置空間の単位格子の組合せ）を列挙して割当てた構造を持つ．効率的な探索を行い，より優れた配置を行うためには，与えられた問題の性質に従って探索木を構成する必要がある．効率に関しては失敗する枝を刈っていくことが重要であり，配置設計の場合，個々の配置要素の大きさが異なっていることや配置制約の限定の程度に強弱があるなどの点で探索に大きな効率化が期待でき，逆に，問題が複雑で大規模であるため，適切な探索木を構成しなければ設計解を得ることも現実的には不可能となる．そこで，探索過程を次の二つの過程に大別して考える．

(1) 配置順序の決定過程：各配置要素の配置順序を決定する過程であり，配置順序によって潜在的に各要素の配置における候補の数，すなわち，図5･5の各節点における分枝の数を減少させて探索の効率化をはかるとともに，各種制約の連なりに従って順序を決定して優れた配置が得られるようにする．

(2) 配置処理の過程：(1)の順序に従って個々の配置要素に対して配置処理



● ：既配置の配置要素　　：配置要素間の強連接関係
○ ：未配置の配置要素　　：配置要素間の弱連接関係
◌ ：強連接関係によって構成される要素のグループ

図5･6　連接性に関するグラフ

を実行する過程であり，個々の要素に対して可能な枝を生成した上で，配置結果の優劣や探索の効率化などに配慮して，より有効な枝を選択しつつ配置を進めていく．

以下の項では，各処理の詳細を示す．

5･5･2　配置順序決定アルゴリズム

各配置要素の配置順序を決定するにあたって，図5･4に示した各種の配置制約のうち，次の制約に基づく関係に着目する．

(1) 強連接関係：要素間の隣接関係に関する領域制約および検査制約に基づいて生じる関係．

(2) 弱連接関係：距離に関する選択制約に基づいて生じる関係．

ここでいう「連接」とは２要素間あるいは要素と空間内の一部領域との間において，一方の位置が定まることによって他方の位置がその近くに限定されるという意味であり，前者は限定の程度が義務的であるのに対して，後者は示唆的である．これらの制約に関して，要素や空間の間の関係は図5･6のような無向グラフを形成する．さらに，強連接関係によって関係付けられた要素は互いに

連続して配置を行ったほうが効率的に優れた配置が得られると考えられることから，それらの要素をひとつのグループとして単一視すると，図はそのようなグループと弱連接関係とからなるグラフと見ることもできる．このようなグラフに基づいてグループを考慮すると配置順序の決定は，

(1) グループ間の順序の決定．

(2) グループに属する要素間の順序の決定．

の２つの問題に分割できる．前者の順序は，

・既配置であるグループや要素と弱連接関係によって関係付けられているグループは次に配置すべきグループである．

・グループに属する要素の大きさの和が大きいグループは小さいグループよりも先に配置すべきである．

という二つの観点から次に配置すべきグループを選択し，そのグループを既配置であるとした上で，次のグループを選択する処理をグループが尽きるまで繰り返すことにより決定する．なお，上記の２つの観点は適当にバランスさせる必要があり，その程度は個々の問題に依存する．また，後者の順序についてはグループに属する各要素の大きさやグループを構成する強連接関係の連なりの順序に従って決定する．以上の二つの処理を組み合わせることにより，すべての要素の配置順序を決定することができる．

5･5･3　配置処理アルゴリズム

図5･7に，配置処理の全体アルゴリズムを示す．これは図5･5に示した探索木を縦方向にたどっていくアルゴリズムであり，前項のようにして決定される配置順序に従って個々の要素に対する配置処理を行い，必要に応じてバックトラックを行いつつ配置解を得るものである．以下に，この過程の個々の要素に対する配置処理について示す．

図5･8に，ある要素に対する配置処理のアルゴリズムを示す．配置要素の配置とは，その要素の形状と大きさに見合う配置空間の単位格子の組合せを求めて要素と格子を結び付ける処理であり，5･4･3項で示した制約の分類に従って，

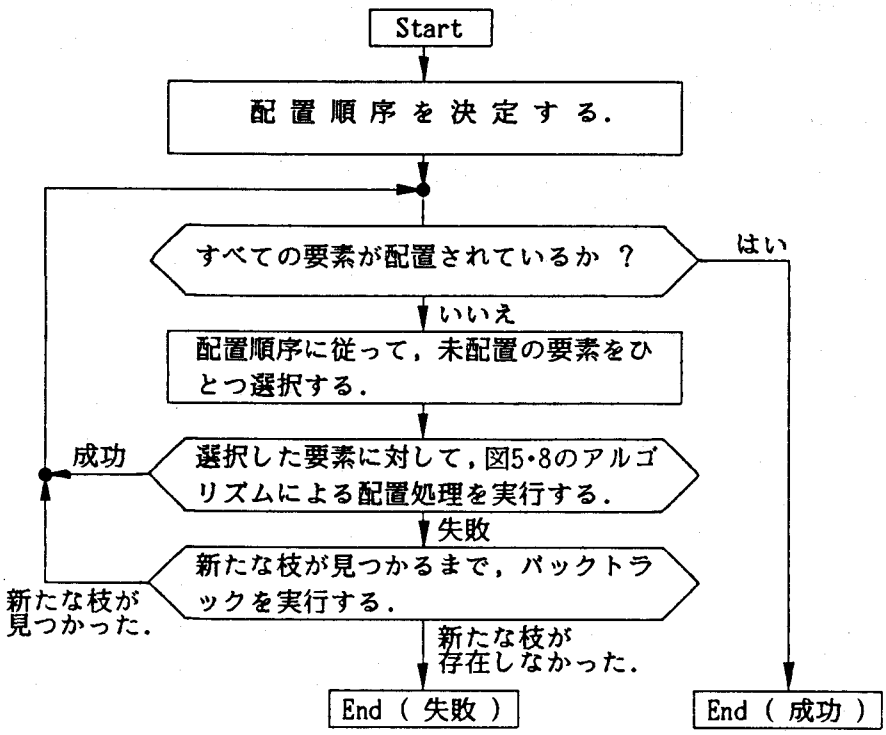


図5・7　配置全体アルゴリズム

まず，組合せの対象となる単位格子を集めてそれらを領域制約によって検査し，次に，要素の形状に従ってそれらを組み合わせて配置候補を生成し，さらに，検査制約によって検査して，続いて，得られた候補に対して選択制約や次項で後述する配置ポテンシャルによって優先付けを行い，優れた候補から配置を試みていくようにする．このとき，残りの候補は，他の要素の配置に失敗しバックトラックが行われてきたときのために保持しておく．最後に，選択した候補に従って配置空間に実際に配置した上で，その配置を大域制約によって検査し，許容されれば次の要素の配置を進め，棄却されればバックトラックを行うようにする．

図5・9は，以上の処理の過程を簡単な例により示したものである．配置空間として２次元の 6×6 個の単位格子からなる空間を考える．図中(a)は，要素Ａと要素Ｂが ▒▒ 部に既に配置されており，要素Ｃの配置を始めようとして

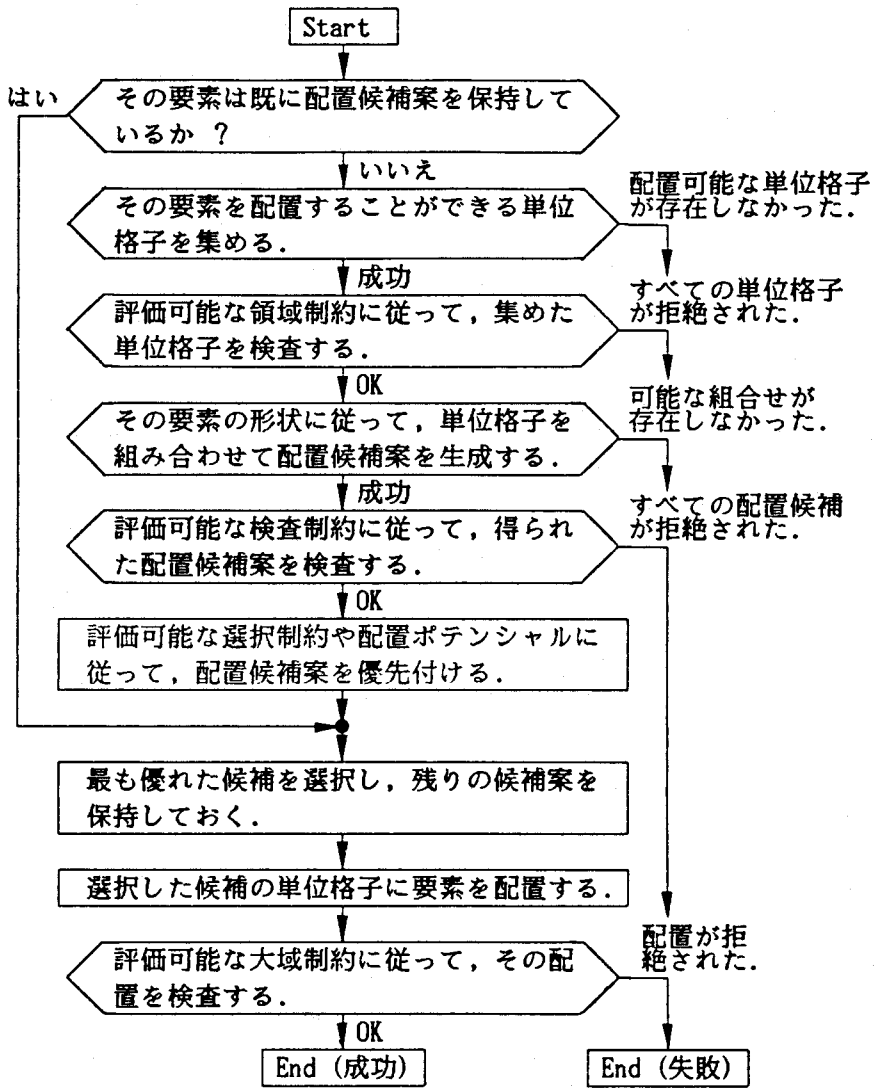


図5･8　個々の配置要素に対する処理アルゴリズム

いる状況を示している．まず，ＣはＡとＢが配置されていない部分の単位格子に配置しなければならない．これに対して，Ｃに関係する領域制約として「ＣはＡよりも上にある」，「ＣとＤは同じ行に存在する」の二つが存在するものとする．前者を適用することにより，配置可能な単位格子は図中(ｂ)の空白部となり，続いて後者を適用するが，要素Ｄが未配置であるため，この制約によっ

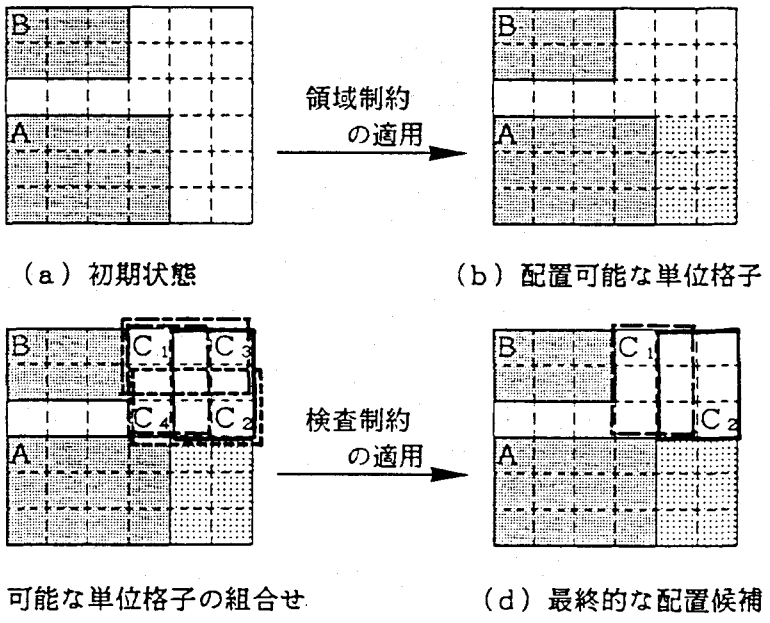


図5･9　配置処理過程の一例

て単位格子を絞りこむことはできない．その結果，Cの配置候補は図中(b)の空白部の中でその形状に従って生成することになる．Cの形状が 2×3 個の単位格子に相当するとすると，この中には計4個の候補（同図(c)）が生成可能であるが，「Cは垂直方向に長く配置する」という検査制約によって，すべての義務的な制約を満足する候補は同図(d)の C_1 と C_2 の計2個となる．最後に，得られた候補群に対して選択制約を適用すると，「BとCは互いに近い方が良い」という制約から C_1 が優先されてCの配置が行われることになる．残る C_2 は，その後の配置処理においてバックトラックを生じCの配置が棄却された場合に，次に採用される候補として保持される．以上のようにして個々の要素に対する配置処理を実行する．

なお，以上のような生成と検査の繰返しによる手法は，5･3･2項で示した単位格子の導入によって，各配置要素に対する配置候補の数が有限個に限定され

ることにより可能となる．ところで，配置候補の優先付けにおける選択制約は極めて局所的なものであり，これを考慮するあまり全体的な配置を行き詰まらせてしまう可能性があり，それを避けて効率的に配置処理を進めていくために，次に示す配置ポテンシャルの概念を導入する．

5･5･4　配置ポテンシャル

ある配置要素における配置候補の優先付けに関して，その要素に関する選択制約が存在しない場合には，すべての候補に対して代替性が存在し，選択制約が存在したとしても比較的優れているいくつかの候補の選択制約による優劣は必ずしも本質的ではないため，若干の代替性が残っている．そこで，このような代替性の中からその配置状況に基づいて全体的な配置処理の観点からより有効な候補を選択するために，配置ポテンシャルの概念を提案して導入する．すなわち，以下のような点に基づいて，選択制約による配置候補の優先付けに対して若干の変更を加えるようにする．

- 選択制約が存在しない場合には，配置空間の隅への配置を優先する．
- 配置結果に無駄な隙間が多く生じて，他の要素が配置できなくなることを避けるために，ある要素に対する配置候補の中で，互いにその配置位置が近いもののうち，その周りに空の単位格子が少ない方の候補を優先させるようにする．
- ある要素を配置しようとしたときに，既配置の要素の周辺のうち，配置しようとする要素と弱連接関係によって関係付けられていない要素で，かつ他の未配置の要素と弱連接関係によって関係付けられている要素の周辺には，他の未配置の要素が配置される可能性が高いため，できればその要素を配置しないようにする．

なお，これらの観点のうちいずれが重要であるかは，個々の配置問題の性質に従って異なってくるため，それぞれを適切にバランスさせる必要がある．また，このような処理は，配置結果全体に対する最適基準に基づくものではないが，

局所的な配置において総合的な観点からより優れていると思われる配置を行うことによって，近似的に準最適な解を得ようとするものでもある．

以上が，値域の概念に基づいた制約指向と生成検査法を組み合わせた本手法の配置アルゴリズムである．

5・6　システム構築方法とその構成

5・6・1　オブジェクト指向によるシステム構築

上述の手法に対して具体的なシステムを構築するためには，その問題解決方法に合致した対象モデルをコンピュータ上で構成する必要があり，その構成がシステムの最終的な汎用性や拡張性を左右することになる[12]．本システムでは，その構築において，各種の情報や知識を表現して配置処理を実現する方法として，以下の理由により，オブジェクト指向プログラミング[7]を用いる．

(1) 制約の伝播に係わる非決定的な処理を従来の手続き的なプログラミング手法により実現することは容易ではない．

(2) 制約系を構成する空間や制約をそれぞれオブジェクトとしてモジュール化して扱うことができ，それらの連携した処理をオブジェクト指向における各オブジェクト間のメッセージ・パッシングによる連携に対応させることができる[13][14]．

(3) 制約などに対する一般的な処理方法はクラスにメソッドとして，具体的な対象の構成や状態についてはインスタンスに変数として表現することによって，汎用性と拡張性に優れたシステムを比較的容易に構成できることが期待できる．

なお，従来の制約指向プログラミングに基づいたシステムのいくつかもオブジェクト指向のもとで構築されている[10][11]が，配置問題においては，制約が複雑で，その性質や取扱いも従来のシステムが扱ったものとは異なるため，独自のシステム構築が必要である．

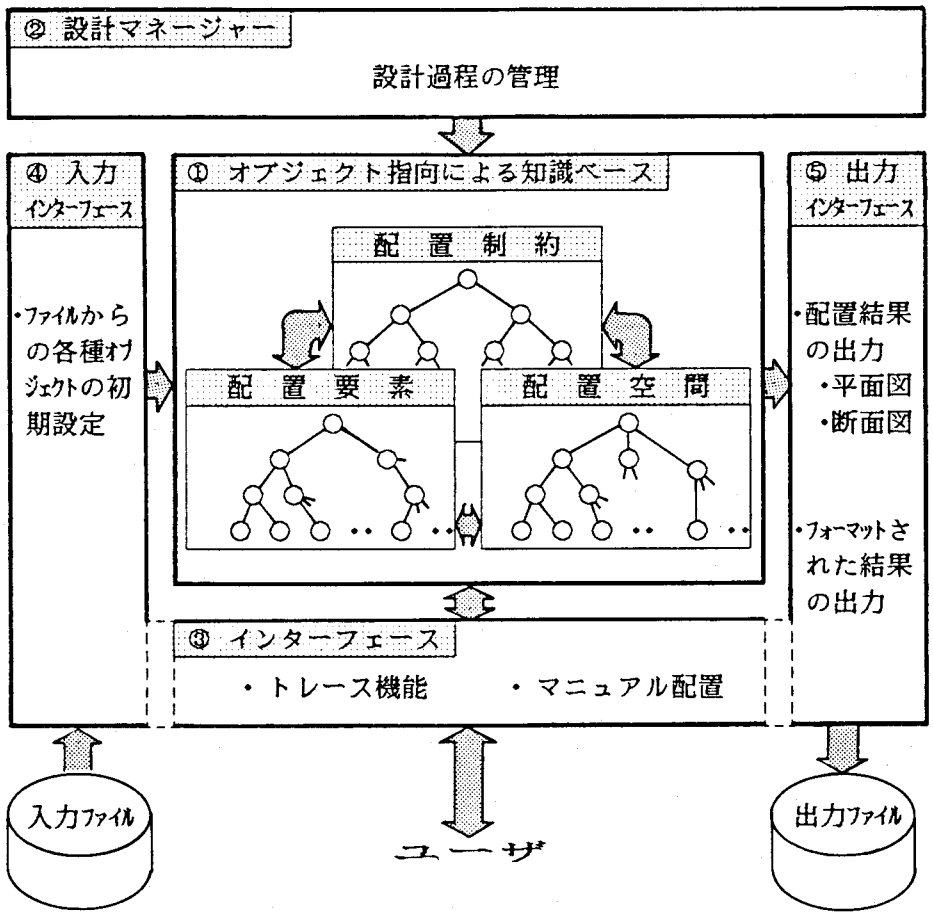


図5・10　配置システムの構成

5・6・2　設計支援システムの構成

本システムの構成を図5・10に示す．システムは，Lisp をプログラミング言語として用い，Lisp により記述したオブジェクト指向プログラミング環境のもとで，①：オブジェクト指向による知識ベースを中心に，②：設計全体を管理する設計マネージャー，③：設計の過程においてユーザとインターフェースをとる部分，④：設計条件をもとにして各種のオブジェクトを初期設定する部分，⑤：配置設計の結果を視覚的に表示したり，配置設計の結果を後の過程のために保存する部分から構成される．それぞれの詳細は以下の節で示すが，①には Lisp 関数として記述された各種のメソッドが含まれ，また，視覚的なイ

ンターフェースの実現にはワークステーション上のウインドウシステムなどの機能を用いる．

なお，本システムはエンジニアリングワークステーション Sun-3 上で構築されている．知識処理の部分は Common Lisp[15]により記述され，ユーザインターフェース機能の一部についてはウインドウシステム SunView[16]，ならびに，グラフィックインターフェース SunCore[17]を用いて C 言語により記述している．また，オブジェクト指向プログラミング環境は，Lisp 上のハッシュ表，構造体，連想リストなど[15]を用いて構築したものである．

5･7　オブジェクトの連携による配置処理

5･7･1　配置設計におけるネットワーク状の関係

配置設計の過程では，5･3･1項で列挙した様々の問題構成要素を取り扱う必要があり，これらの間には以下のような種々の関係が存在する．

(1) 制約関係：ある制約がどの要素と要素あるいは配置空間との空間的な位置関係を規定しているかを表現する関係．なお，この制約関係は，制約自体が前述のように基本的に４種類に分類できることに対応して，同様に分類することができる．

(2) 配置関係：ある要素がどの単位格子に配置されているかを表現する関係．

(3) 空間部分関係：配置空間における全体と部分との集合関係，その部分と単位格子との所属関係を表現する関係．

(4) 単位格子間関係：ある隣接する２個の単位格子間の相対的な位置関係を表現する関係．

これらの関係は，図5･11に示すようにそれぞれの種類に応じて，各問題構成要素をノード，関係をアークとするネットワークを形成している．配置設計における処理は，5･7･3項で後述するような方法により，配置関係を生成しながらその上をノードからノードへと渡り歩いていくことにより行われるものとして

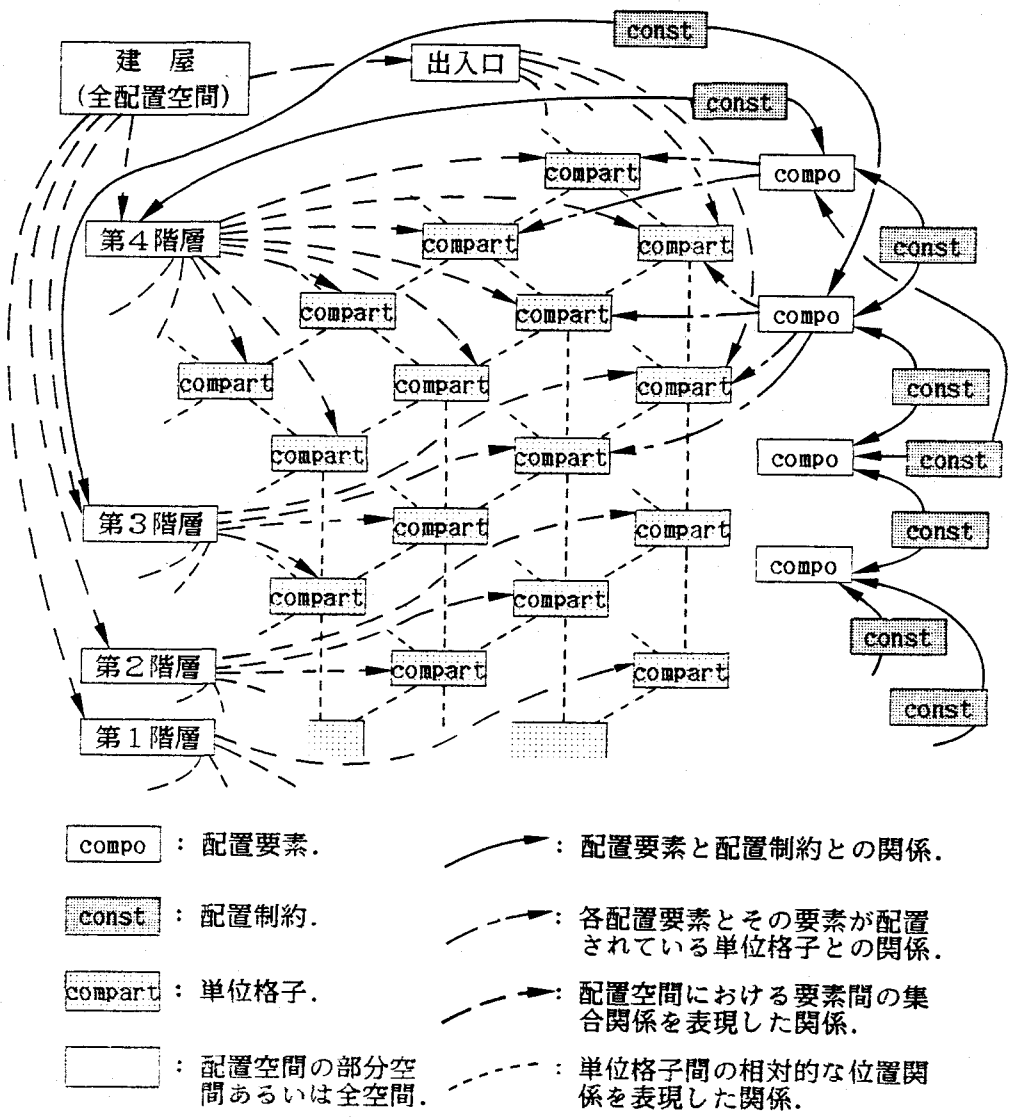


図5･11　問題構成要素間のネットワーク

理解することができる．本システムでは，このようなネットワークに対してオブジェクト指向プログラミングを導入して，図5･11の各ノードをモジュール化してインスタンスとして表現し，上記の各種の関係に対しては新たに「オブジェクト間の関係」として扱うようにする．また，配置要素におけるその形状や配置候補群，配置空間におけるそれぞれの絶対的な位置などの各インスタンスに

固有の情報は，それぞれのインスタンスに変数として保持するようにする．

なお，上記の「関係」による表現方法を加えることにより，オブジェクトそれぞれの内部状態を変数として表現することに加えて，オブジェクト間の帰属関係や支配関係などが容易に扱えるようになる．この表現は，関係が一度定義されると双方のインスタンスから参照できる，インスタンスが消去されるとそれに関連する関係はすべて消去されるなどの性質をもつ．

5･7･2　オブジェクト指向による知識表現

前項で示したネットワーク上の処理は，それぞれのインスタンスが固有の処理を行い，それらが連携することによって実行される．オブジェクト指向においては，インスタンスにおける処理はそのクラスから継承されるメソッドにより行われるため，各種処理の種類や内容に応じてクラスを用意して，スーパークラス・サブクラスの階層構造を構成する必要がある．特に，配置制約においては，前述の領域制約・検査制約・選択制約などの区分に加えて，さらに前出図5･4のような詳細な区分があり，各制約において同じ抽象的な処理を行おうとする場合に，その具体的な方法は，共通の処理で対応できる場合もあれば，それぞれに個別の方法によって行う必要がある場合もある．そこで，このような処理内容の違いに対応してインスタンスを分類し，その分類に従ってクラスの階層を構成して，共通的な処理は上位のクラスに，個別の処理は下位のクラスに記述するようにする．これにより，オブジェクト指向における差分プログラミングの性質を生かした効率的な処理の記述が可能になる．以下に，オブジェクトによる表現の具体的な実例を示す．

図5･12は，5･9節で取り上げる原子力発電所の事例における各種インスタンスの記述例である．図中(a)は配置要素の記述例であり，その形状や配置候補群を変数として，関連する制約を関係により保持している．(b)は配置空間内のある単位格子の記述例であり，その絶対的な位置を変数として，配置されている要素を関係により保持している．(c)は(b)のような単位格子を元とする配置空間の部分集合の記述例であり，それに属する単位格子を関係により保持

```
(inst (name sfp-p) (class regular-area)
      (iv (width 1) (depth 1) (height 1)
          (candidates
             ((comp-5-14-2) (comp-5-15-2))) ···      )
      (as (laid-to comp-6-13-2))
      (as-re (limit-search-comparts vertical-56 ··· )
             ···  ···                                ) )
```

（a） 配置要素の記述例

```
(inst (name comp-6-13-2) (class compartment)
      (iv (column 6) (row 13) (floor 2))
      (as (composed-of floor-2))
      (as-re (laid-to sfp-p)) )
```

（b） 配置空間の単位格子の記述例

```
(inst (name grand-level) (class boundary-compartments-set)
      (as-re (connect-to comp-12-1-4 comp-12-2-4 ··· )) )
```

（c） 配置空間の部分集合の記述例

```
(inst (name vertical-56) (class vertical-const)
      (iv (lower sfp-p) (upper sfp-hx) (inter-floors -1))
      (as (limit-search-comparts sfp-hx sfp-p)) )
```

（d） 配置制約の記述例

図5･12 インスタンスの記述例

している．(d)は配置制約の記述例であり，その制約に関連している要素や領域を関係により保持している．なお，このようなインスタンスは個々の問題に応じて，5･7･4項で後述するような記述から自動的に生成するようにする．

図5･13はクラスの記述例である．図中(a)はある領域制約のクラス，(b)はある検査制約のクラスであり，両者のスーパークラスには共通のものが存在し，同名のメソッドを持っているが，その処理の実体である Lisp 関数はそれぞれ

```
(class (name search-comparts-const)
  (super region-constraint)
  (im (can-evaluate      (call can-eval-search-comparts-const) )
      (get-opposite-side
            (call get-opposite-side-of-search-comparts-const) )) )
```

(a) ある領域制約のクラスの記述

```
(class (name contain-comparts-const)
  (super test-constraint)
  (im (can-evaluate      (call can-eval-contain-comparts-const) )
      (get-opposite-side
            (call get-opposite-side-of-contain-comparts-const) )) )
```

(b) ある検査制約のクラスの記述

```
(class (name just-close-const)
  (super search-comparts-const)
  (cm (create (call create-just-close-const              ) ))
  (im (reduce-compartments
            (call reduce-comparts-by-just-close-const) )) )
```

(c) 上下関係に関する制約のクラスの記述

図5・13 クラスの記述例

異なったものを保持している．例えば，制約において対をなす要素の一方から他方を求めるための 'get-opposite-side' というメソッドは，共通的な処理であるが，要素と制約を関係付けている関係の種類が異なるため，具体的な処理は異なったものとなる．また，(c)は(a)のサブクラスである具体的な制約に対するクラスであり，インスタンスを生成して必要な関係や情報を保持させる 'create' というメソッドや，インスタンスの内容に応じて候補となる単位格子を絞りこむ処理を行う 'reduce-compartments' というメソッドを保持している．

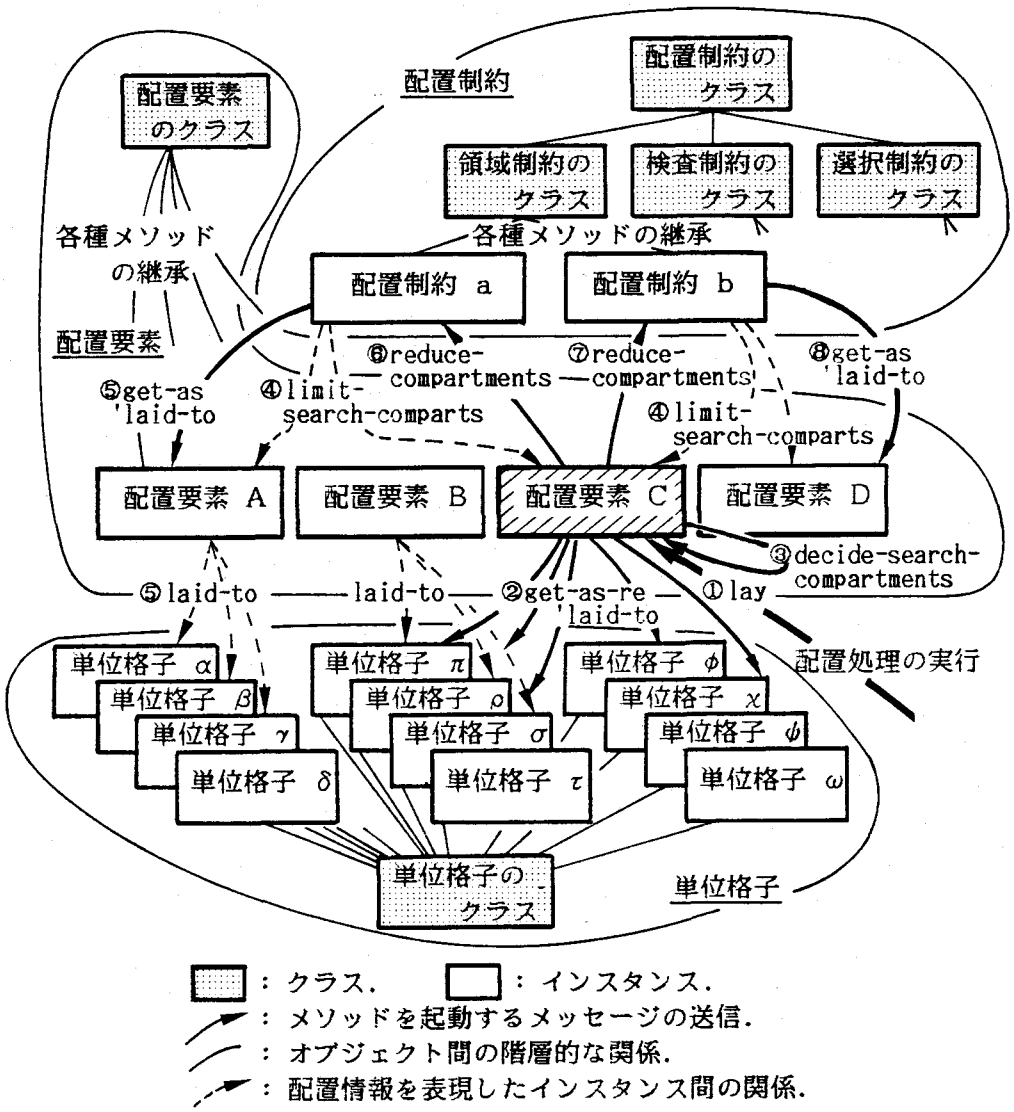


図5･14　メッセージ・パッシングによる配置処理の一例

5･7･3　オブジェクトの連携による配置処理の一例

続いて，上述のオブジェクトによる表現のもとで配置処理を行う方法について示す．図5･14は，5･5･3項で示した処理の中で，ある要素(C)の配置を行うために，何も配置されていない単位格子を領域制約によって絞りこんで，配置候補を生成する対象となる単位格子を集める処理の過程を示したものであり，

以下の手順で制約伝播を行いながら処理が進められる．まず最初に，要素Cのインスタンスに対して配置処理を起動するメッセージ 'lay' が送信される(①)と，直ちに各単位格子のインスタンスに対する働きかけ(②)によって空の単位格子が集められる(③)．続いて，要素Cに関連する領域制約(aとb)がインスタンス間の関係 'limit-search-comparts'(④) から参照され，これらの制約によって単位格子群が絞りこまれていく．まず，制約aによって要素Aの位置が関係 'laid-to'(⑤) によって参照され，制約の内容に従って一部の単位格子が除外される(⑥)．次に，得られた単位格子群がさらに制約bによって処理される(⑦)が，この場合は，要素Cと対をなす要素Dが配置されていない(⑧)ため，制約bによる単位格子の絞りこみは実際には行われない．このようにして目的とする単位格子の集合すなわち配置可能領域が得られる．なお，上記の⑥と⑦の処理を行うメソッドはそれぞれの配置制約のクラスに共通的に記述されており，各インスタンスに宣言的に保持された情報を参照しながら，対をなす反対側の要素の配置が行われているかどうかを判断することによって，その処理内容が配置過程において確定するようになっている．さらに，以上の処理に続いて，他の制約関係を参照しつつ処理を進めていき配置すべき単位格子が定まると，この要素とそれらの単位格子の間で 'laid-to' という関係が生成されて，配置関係が表現される．

このほか，各種の制約を用いた処理も同様の方法により，状況に応じた方向に制約を展開しながら行われるようにする．

5･7･4　各種インスタンスの初期設定

最後に，各インスタンスを配置条件に従って初期設定する方法について示す．本システムでは，図5･15に示すような簡便な形式で配置条件をファイルに表現しておき，配置設計を行うにあたり，その内容に応じて自動的に図5･12のようなインスタンスを生成して必要な情報を保持させる．図中(a)は，配置要素とその形状などを定義した部分であり，例えば，①の行は「charge-pa という要素が存在し，その形状が 3×1×1 個の単位格子に相当する」ことなどを示し

```
(area charge-pa    (3 1 1) controlled   connect-to-access)  ·····①
(area boric-ac-ta  (1 1 2) controlled   connect-to-access)
                ···   ···   ···
(area n-relay      (4 3 1) uncontrolled connect-to-access)
                ···   ···   ···
```

（a） 配置要素の記述例

```
(height            4  )
(north-west-length 16 )   } ·····②
(east-west-length  12 )
(grand-direction (4 north))
(grand-direction (4 west ))
      ···   ···   ···
```

（b） 配置空間の記述例

```
(close               spray-pb   safeaux-ps)  ·····③
(different-direction spray-pb   safeaux-ps)
(just-under          spray-pb   spray-hxb )
(near                spray-addt spray-pb  )
            ···   ···   ···
```

（c） 配置制約の記述例

図5･15 設計条件の入力形式

ている．(b)は，配置空間に関する条件を与えている部分であり，②の部分は設計対象のプラントのサイズを単位格子の縦・横・高さ方向の個数として定義している．(c)は，配置制約の記述であり，例えば，③の行は「spray-pb と safeaux-ps が隣接する」という制約を表している．なお，このような制約の記述形式として，基本となるものを22個用意し，さらに，それに対して拡張子を付加することにより，数十種類を用意した．このような機能によって，ユーザは図5･12に示したようなシステム内部での表現を意識する必要がなくなる．また，基本配置の結果についても同様の形式でファイルに出力し，続いて行われるべき詳細配置設計過程との連携が行えるようにする．

5・8　ユーザインターフェース

以下に本システムにおけるユーザインターフェースの機能を示す．

5・8・1　配置過程のトレース機能

一般に，設計支援システムにおいては，そのユーザが設計実務者であるため，設計処理の過程を示すことにより，解の背後に潜んだ個々の設計問題の性質を把握させ，その解の合理性を具体的に示すことが重要である．さらに，配置設計の場合には，与えた配置制約に不備があったり，必要な制約が欠如している場合にも対処できるように，個々の配置処理の進行状況をトレースしてユーザに提示することは特に重要となる．本システムにおけるこのようなトレース機能の実例として，5・9節で取り上げる事例における例を，図5・16および図5・17に示す．前者は文字レベルの機能で，後者は図形レベルの機能であり，図中の丸付きの番号で示した部分が互いに対応している．前者は，配置処理内容を適当な段付を行って表示するものであり，各処理が時間的にどのようにして行われていくかを容易に知ることができる．後者は，その時点での配置状況を逐次視覚的に表示していくものであり，一目で配置状況を把握することができる．このような両者の機能が補完し合うことにより，個々の処理内容を容易に理解できるようになる．

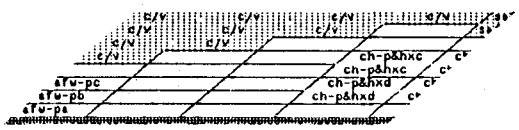
なお，後者の機能については，Lisp プロセスの側から C 言語によりウインドウシステム上で記述された図形処理プロセスを起動し，Lisp 側のオブジェクトにおけるメソッドの処理の中から，両者の間で配置情報を送信して，逐次配置状況を表示していくことにより実現する．

5・8・2　ユーザによるマニュアル配置の機能

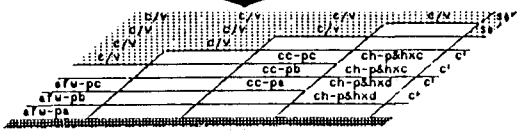
本システムによる配置は，5・5・4項で示した配置ポテンシャルの概念によって大域的な配置が考慮されているが，場合によっては問題のある配置状況に陥ったり，また，ユーザ自身が，システムが提示する以外の配置解を得ようとする

```
                              .
                              .
                              .
                              .
                              .
>>> Is the current layout acceptable ?  : y
58                  CH-P&HXA's layout to ( COMP-2-1-2 ... ) is rejectec
58                  CH-P&HXA's layout to ( COMP-3-1-2 ... ) is rejectec
58                  CH-P&HXA is laid to ( COMP-4-2-2 ... ). 98 candidat
59                   CH-P&HXB is laid to ( COMP-4-4-2 ... ). 1 candidat
60                    CH-P&HXC is laid to ( COMP-5-4-2 ... ).
61 CH-P&HXD is laid to ( COMP-5-2-2 ... ). 1 candidate is rest.
>>> Is the current layout acceptable ?  : y                    ......①
62  CC-PA's layout to COMP-5-3-3 is rejected. 94 candidates are rest.
62  CC-PA's layout to COMP-5-4-3 is rejected. 93 candidates are rest.
62  CC-PA's layout to COMP-5-2-3 is rejected. 92 candidates are rest.
62  CC-PA's layout to COMP-4-1-3 is rejected. 91 candidates are rest.
62  CC-PA's layout to COMP-5-1-3 is rejected. 90 candidates are rest.
62  CC-PA's layout to COMP-3-1-3 is rejected. 89 candidates are rest.
62  CC-PA's layout to COMP-2-1-3 is rejected. 88 candidates are rest.
62  CC-PA is laid to COMP-6-3-2. 87 candidates are rest.
63   CC-PB is laid to COMP-6-4-2. 2 candidates are rest.
64    CC-PC is laid to COMP-6-5-2. 1 candidate is rest.·············②
65     CC-PD cannot be laid.
64    CC-PC is removed.
64    CC-PC's layout to COMP-7-4-2 is rejected.···················③
63   CC-PB is removed.
63   CC-PB is laid to COMP-6-2-2. 1 candidate is rest.
64    CC-PC's layout to COMP-6-1-2 is rejected. 1 candidate is rest.··④
64    CC-PC's layout to COMP-7-2-2 is rejected.···················⑤
63   CC-PB is removed.
63   CC-PB's layout to COMP-7-3-2 is rejected.
62  CC-PA is removed.
62  CC-PA is laid to COMP-6-4-2. 86 candidates are rest.
63   CC-PB is laid to COMP-6-3-2. 2 candidates are rest.
64    CC-PC is laid to COMP-6-2-2. 1 candidate is rest.
65     CC-PD's layout to COMP-6-1-2 is rejected. 1 candidate is rest.··⑥
65     CC-PD's layout to COMP-7-2-2 is rejected.···················⑦
64    CC-PC is removed.
64    CC-PC's layout to COMP-7-3-2 is rejected.
63   CC-PB is removed.
63   CC-PB is laid to COMP-6-5-2. 1 candidate is rest.
64    CC-PC cannot be laid.
63   CC-PB is removed.
63   CC-PB's layout to COMP-7-4-2 is rejected.···················⑧
62  CC-PA is removed.
62  CC-PA's layout to COMP-5-1-2 is rejected. 85 candidates are rest.
62  CC-PA's layout to COMP-4-1-2 is rejected. 84 candidates are rest.
62  CC-PA's layout to COMP-3-1-2 is rejected. 83 candidates are rest.
62  CC-PA's layout to COMP-2-1-2 is rejected. 82 candidates are rest.
62  CC-PA's layout to COMP-7-1-2 is rejected. 81 candidates are rest.
62  CC-PA is laid to COMP-6-2-2. 80 candidates are rest.
63   CC-PB is laid to COMP-6-3-2. 2 candidates are rest.
64    CC-PC is laid to COMP-6-4-2. 1 candidate is rest.
65     CC-PD is laid to COMP-6-5-2. 1 candidate is rest.·············⑨
>>> Is the current layout acceptable ?  : y
66      SFP-HX is laid to ( COMP-7-15-2 ... ). 88 candidates are rest.
67       SFP-P is laid to COMP-7-14-2.
>>> Is the current layout acceptable ?  : y
                              .
                              .
                              .
                              .
```

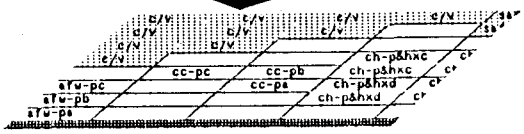
図5・16　文字レベルの配置トレース機能



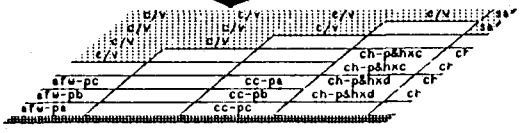
①: 'cc-pa'を配置しようとする直前の配置状況．この状態に続いて，'cc-pa'を第３階層に配置する試みが７回行われるが，すべての配置は通路の問題により棄却される．



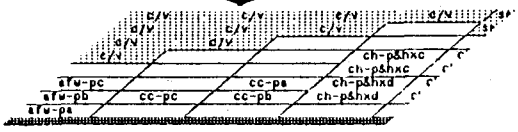
②: ①の後，'cc-pa'が第２階層内に配置され，続いて，'cc-pb'，'cc-pc'の順で配置が行われていく．しかし，この状況で'cc-pd'を配置することができない．



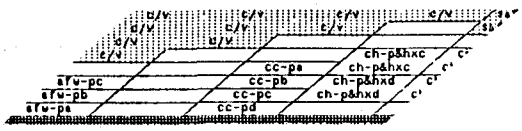
③: 'cc-pc'の配置が一度棄却され，別の場所に配置されるが，その配置も通路をふさいでしまうため棄却される．



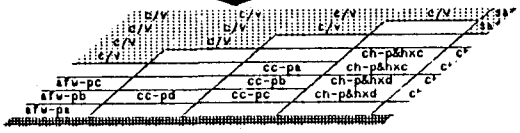
④: 'cc-pc'は他に配置候補を保持していないため，'cc-pb'の配置も棄却され，別の格子に配置され，続いて，'cc-pc'が配置されるが，その'cc-pc'の配置も通路の問題により棄却される．



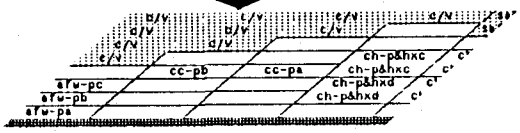
⑤: 'cc-pc'が別の格子に配置されるが，その配置も通路の問題により棄却され，続いて，'cc-pb'の配置も棄却される．



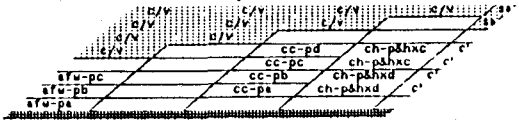
⑥: さらに，もうひとつの'cc-pb'の配置が棄却されて，結局'cc-pa'の配置が棄却され，各要素が配置されていく．しかし，その配置も'cc-pd'が通路をふさぐため，棄却される．



⑦: 'cc-pd'が別の格子に配置されるが，これも通路をふさいでしまうため，棄却される．



⑧: ⑦に続いて，'cc-pc'，'cc-pd'を他の格子に配置しても問題が生じるため，'cc-pb'の配置が棄却されることになり，再配置が行われる．



⑨: ⑧の配置も棄却され，'cc-pa'を再配置して配置を進めると，'cc-pb'，'cc-pc'，'cc-pd'のすべてが配置できる．

図5･17　図形レベルの配置トレース機能

(第２階層の一部領域の配置状況の時間的推移)

こともあり得る．このような場合に対応するためには，ユーザが配置処理に積極的に介入できるようにする機能が必要となる．本システムにおいては，前項で示したトレース機能により配置処理過程を容易に把握することができるため，ユーザにとって真に介入が必要な状況を知ることは比較的容易である．そこで，システムがある一連の要素群（5･5･2項における要素のグループ）の配置を終了した時点で，介入が必要である場合には任意の要素の配置を棄却して強制的に探索におけるバックトラックを行わせることができるようにする．これによって，配置制約を満足しないような配置結果に陥ることを避けつつ，ユーザがシステムの処理に介入できるようになる．

5･9　事 例 ── 原子力発電所の配置設計への適用 ──

最後に，本システムを原子力発電所の基本配置設計に適用した事例を示す．この配置問題は，100 台前後の機器を３次元的に配置する問題であり，その設計においては経済性に加えて安全性や信頼性に対する要求が重要となる．しかるに，従来は熟練経験者の勘や経験によって設計が行われていたため，合理性に富む解を得るためには多大な設計時間を要するなどの困難が伴っていた．これに対して，本システムを適用することにより，有効に配置過程を支援しつつ，配置制約に裏付けされた合理的な配置が容易に行えるようになる．ある発電所の配置設計を行った事例を以下に示す．設計対象は４階層から成り，その各階層は 12×16 個の単位格子から構成されている．扱った配置要素には表5･1に示すようなものがあり，あらかじめその位置が決定されていた格納容器などの７ 個の要素を除いて計 73 個の要素の配置を行った．また，考慮した配置制約はおよそ 200 個である．図5･18は配置の結果であり，図中(ａ)は全体の透視図，(ｂ)はある階層の平面図，(ｃ)と(ｄ)はそれぞれ東西・南北方向のある位置での断面図である．本結果における配置データは熟練設計者による実設計の結果を参照しつつ作成したものであり，システムによる配置結果と設計者に

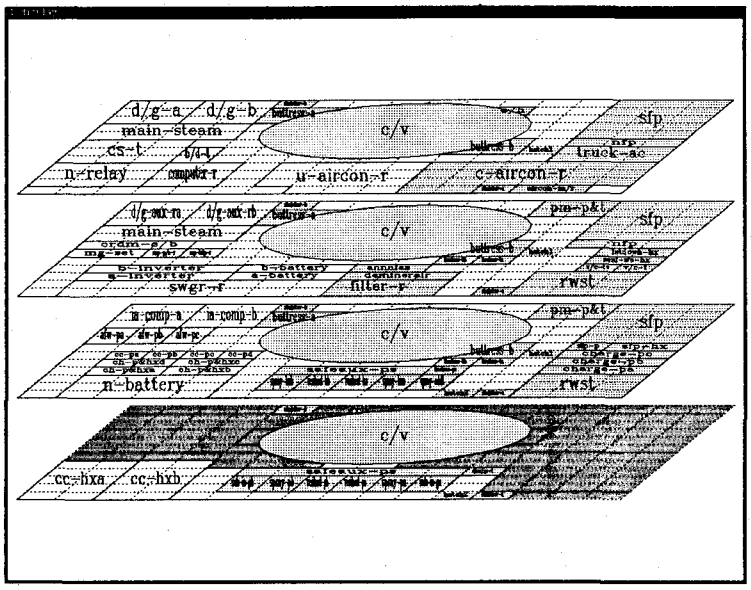
表5・1　原子力発電所に配置される機器

機器／区域名称	区域区分	形　状 (単位格子の個数)	備考
格納容器	管理区域		既配置とする
充塡ポンプA	管理区域	3×1×1	
充塡ポンプB	管理区域	3×1×1	
充塡ポンプC	管理区域	3×1×1	
ほう酸ポンプ	管理区域	1×1×1	
ほう酸タンクA	管理区域	1×1×2	
ほう酸タンクB	管理区域	1×1×2	
非再生冷却器	管理区域	2×1×1	
封水冷却器	管理区域	2×1×1	
・・・	・・	・・・	
・・・	・・	・・・	
燃料取替え用水ピット	管理区域	3×3×2	
トラックアクセス	管理区域	3×2×1	
・・・	・・	・・・	
・・・	・・	・・・	
主蒸気管室	非管理区域		既配置とする
復水タンク	非管理区域	3×3×1	
制御用空気圧縮器A	非管理区域	3×2×1	
制御用空気圧縮器B	非管理区域	3×2×1	
MGセット	非管理区域	2×1×1	
・・・	・・	・・・	
・・・	・・	・・・	

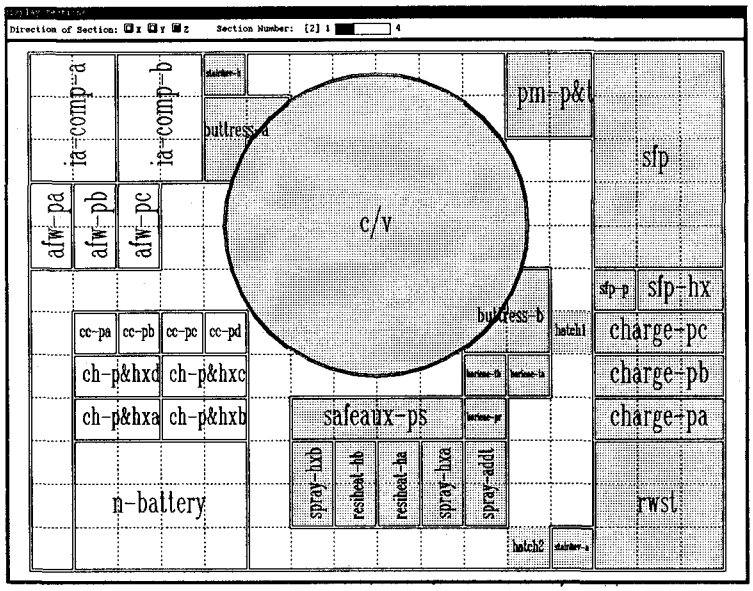
よるそれとを比較した結果，本配置が極めて満足すべきものであることを確認した．

5・10　結 言

本章では，基本配置設計に対して，その設計タスクの理解に基づき，設計支援システムの構成方法として制約指向の概念に基礎を置く設計支援手法を提案した．さらに，その手法をもとにオブジェクト指向プログラミングの手法を用

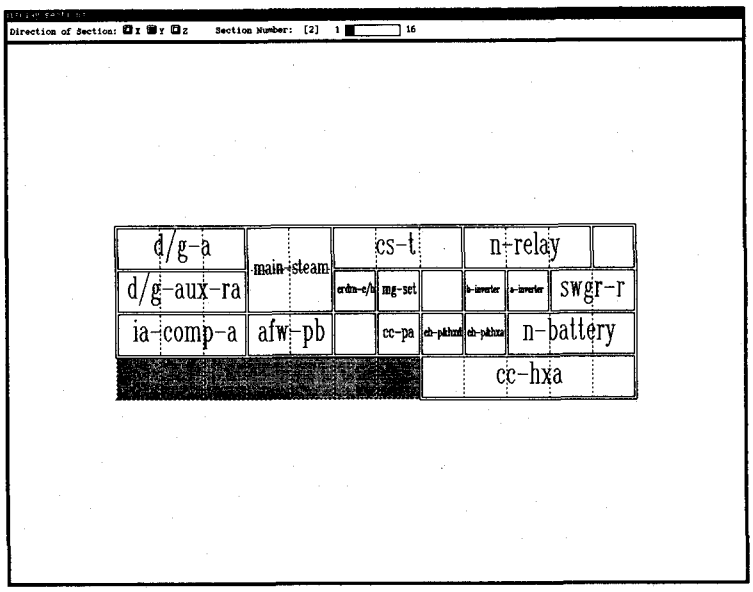


（ａ） 全体透視図

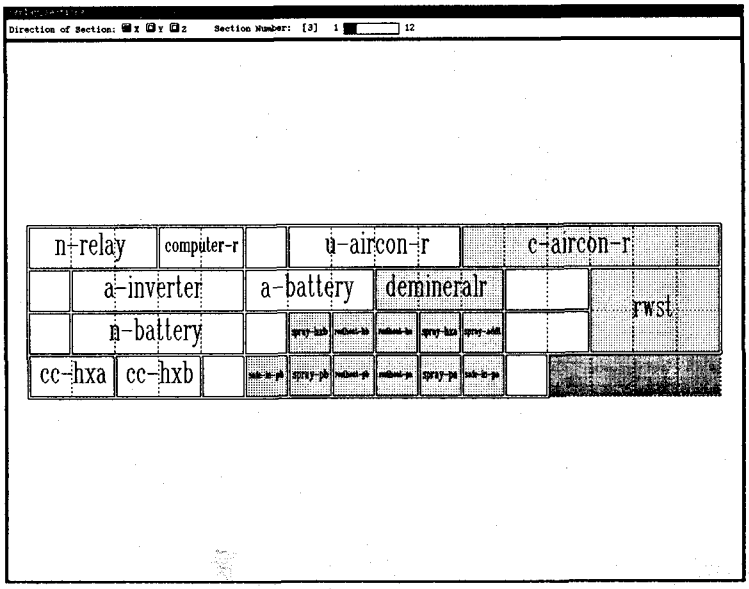


（ｂ） ある階層の平面図

図5･18 原子力発電所の配置結果の一例（１）



（ｃ） 東西方向の断面図



（ｄ） 南北方向の断面図

図5･18 原子力発電所の配置結果の一例（2）

いて構築した配置設計支援システムについて示した．配置設計は従来，熟練設計者の経験や勘に基づいて行われており，支援システムにおいてもそのような知識を直接記述したものが多く，配置知識は一般化されて整理されているとは言えなかった．これに対して，ここで提案した手法，ならびに構築したシステムは，以下の特徴を持つ．

(1) 配置設計における知識を，制約という概念のもとに一般化して整理し，その制約を配置処理における取扱いによって階層的に分類した．

(2) 制約の分類に従って，制約指向プログラミングの手法と生成検査法を組み合わせることによって，制約の宣言的な記述をもとに，配置処理を行う一般的なアルゴリズムを確立した．

(3) 制約として取り扱えないような大域的な配置知識に対して，配置ポテンシャルの概念を導入し，適切な配置処理の実現を可能にした．

(4) システム構築の具体的な手段として，オブジェクト指向プログラミングを用いたことにより，一般性と拡張性に優れたシステムとなっている．

(5) ワークステーションの各種機能に基づいた各種の有効なインターフェース機能を有している．

このような手法は，配置設計に対する汎用的な手法の基礎となるものであり，原子力発電所の基本配置設計のみならず，各種の配置問題に適用することができる．また，システム構築の基礎になっている問題構成要素間のネットワーク状の関係に基づいたオブジェクト指向による問題表現は，各種の設計問題においても有効であり，設計支援システムの構築の一形態を与えるものでもある．

文　献

(1) McDermott, J., "R1: A Rule-Based Configurer of Computer Systems", Artificial Intelligence, no.19, (1982), pp.39-88.

(2) 渡辺ほか，計算機室レイアウト用エクスパート・システムの開発，情報

処理学会論文誌, vol.26, no.5 (1985), pp.926-935.
(3) Kumara, S. R. T. et al., "Application of expert systems and pattern recognition methodologies to facilities layout planning", International Journal of Production Research, vol.26, no.5, (1988), pp.905-930.
(4) 例えば, Leler, W., Constraint Programming Languages － Their Specification and Generation, (1988), Addison-Wesley.
(5) 鈴木, 制約指向プログラミング, 知識プログラミング (淵 監修) 第2章, (1988), 共立出版, pp.27-39.
(6) 文献(5), pp.34-36.
(7) 本論文, 第2章の文献(18)~(20).
(8) 例えば, Liggett, R. S., "Optimal Space Planning in Practice", CAD, vol.13, no.5, (1981), pp.277-288.
(9) 例えば, Chan, W. J. and Paulson Jr., B. C., "Exploratory design using constraints", AI EDAM, vol.1, no.1, (1987), pp.59-71.
(10) Sussman, G. L. and Steele Jr., G. L., "CONSTRAINTS － A Language for Expressing Almost Hierarchical Descriptions", Artificial Intelligence, no.14, (1980), pp.1-39.
(11) Borning, A. H., "ThingLab － A Computer-Oriented Simulation Laboratory", (1979), Stanford Univ.
(12) 木村, オブジェクト指向によるＣＡＤ／ＣＡＭのためのモデリングとデータベース, 情報処理, vol.29, no.4, (1988), pp.368-373.
(13) Abelson, H. and Sussman, M. K., Structure and Interpretation of Computer Programs, (1985), MIT Press, pp.230-242.
(14) Winston, P. H. and Horn, B. K. P., LISP (3rd Edition), (1989), Addison-Wesley, pp.335-357.
(15) Steele Jr., G. L., Common Lisp: The Language, (1984), Digital Press, (邦訳: 後藤・井田, (1985), 共立出版).
(16) SunView 1 Programmer's Guide, (1988), Sun Microsystems.
(17) SunCore Reference Manual, (1988), Sun Microsystems.

第6章　総括

本論文では，知識情報処理の技術に基づいて，基本設計におけるシンセシスの過程を支援しようとする立場から，いくつかの具体的な設計問題に対する設計支援手法を提案し，それをもとにした設計支援システムを構築して，その有効性を検証した．

基本設計における問題は，いわゆる悪構造問題（ill-structured problem）であるといわれるように，一般には，その方法論を明らかにすることが容易ではない．本研究では，設計過程に対する分析に基づいて，設計問題のタイプを「創成形」・「知的検索形」・「試行錯誤形」・「最適形」に分類した上で，なかでも，設計シンセシスの面では最も比重が大きい基本設計における問題，すなわち，知的検索形と試行錯誤形の設計に対して支援手法を提案した．各設計手法の提案にあたっては，これらの設計における処理の内容が非定型的であり，また数値的な処理とは馴染まないこと，設計者との協調した処理が不可欠であることなどの点を考慮し，そのような処理を記述する手段として知識情報処理の技術を援用することを基本とした．さらに，設計支援システムの構築においては，実際の問題を取り上げてその有効性を具体的に検証することが重要であるとの立場から，基本設計における具体的な問題として，機能設計，プラント機器構成設計，基本配置設計の問題を取り上げ，それぞれに対して各手法による設計支援システムを構築した．以下に，各章で得られた成果を総括する．

第１章では，本研究の目的や意義について述べた．

第２章では，まず，設計問題とコンピュータ利用技術との関連について論じた上で，設計問題のタイプを類別し，基本設計におけるシンセシスの過程を有効に支援するための手法や，その実現方法として知識情報処理技術を導入しようとする立場を明らかにした．

第３章では，試行錯誤形の問題として機能設計を取り上げ，設計項目間の依存関係に着目した「ネットワークモデル」を提案し，それに基づいて機能設計における「設計・評価・再設計」の過程を柔軟に支援するための設計支援システムを構築した．本システムは，オブジェクト指向に基づいた設計対象の表現やネットワークモデルによる基本的な処理機能をはじめとして，感度解析や最適化手法との融合による設計目標間の総合化に対する支援，形状モデリング手法との融合による機能評価に対する支援機能により，機能設計の過程を柔軟に支援することができる汎用的なシステムとなっている．さらに，構築したシステムを船舶の基本設計に適用して，その有効性を検証した．

第４章では，知的検索形の問題としてエネルギープラントにおける機器構成設計を取り上げ，オブジェクト指向に基づいた設計データの表現を基本として，機器の検索とプラント候補の合成，プラント候補に対する評価を段階的に進めていくことにより，機器構成設計の過程を有効に支援するための手法を提案した．また，このような手法に基づいて，舶用動力プラントならびにコージェネレーションプラントの設計を対象にした設計支援システムを構築し，知的検索形の問題を有効に支援できることを検証した．

第５章では，第３章で取り上げた機能設計とは異なった性質を有する試行錯誤形の問題として基本配置設計を取り上げ，「制約指向」の考え方に従った配置設計手法を構成し，オブジェクト指向プログラミングのもとで，汎用的な配置アルゴリズムと個別問題における制約の記述とからなる設計支援システムを構築した．このような構成により，本システムは，高い拡張性と一般性を有しており，また，その適用により配置設計における条件を明確化することができるなどの点で，配置設計に対する合理的な支援手法となっている．さらに，構築したシステムを原子力発電所の配置設計に適用して，その有効性を検証した．

以上のように，本論文では，第３章から第５章にわたって，基本設計において特徴的となる具体的な問題を取り上げて，知識情報処理の技術を援用した設計支援システムを構築した．基本設計における問題は，一般には，その方法論を明らかにすることが容易ではなく，個々の設計問題の特徴をとらえた上で，

その構成に対応して，設計者によるシンセシスの操作を有効に支援していく必要がある．本論文で取り上げた各設計問題に対する支援手法は，それぞれの問題において特徴となる設計処理に着目して構成したものであり，基本設計の過程を柔軟に支援することができる．また，各支援手法は，それぞれのタイプの設計問題に対する一般的な設計支援手法を与えるものであり，広く類似した問題に適用することができるものと考えられる．さらに，各支援システムの構築に際して用いたオブジェクト指向によるシステムの構成方法は，設計支援システムの構築におけるプログラミング手法として広く有効なものである．

謝辞

本研究を遂行するにあたり，終始御厚意あふれる御指導および御鞭撻を賜わりました大阪大学工学部産業機械工学科 赤木新介教授に心より深く感謝の意を表します．

本論文を作成するにあたり，御懇切な御校閲と御助言をいただきました大阪大学工学部電子制御機械工学科 大川善邦教授，白井良明教授，木村英紀教授，岩田一明教授に深く感謝いたします．

本研究を進めるにあたり，適切な御助言をいただきました大阪大学工学部産業機械工学科 伊東弘一助教授，さらに，適切な御意見をいただきました大阪大学工学部産業機械工学科 横山良平助手に深く感謝いたします．

また，事例として取り上げた各設計問題に関して，御教示いただきました大阪ガス(株)技術部 長谷川 洸 氏，山本俊夫氏，ならびに三菱原子力工業(株)プラント設計部 仲戸川哲人氏，井上具大氏 ほかの各位に感謝の意を表します．

さらに，各設計支援システムを構築するにあたり，御協力いただきました大阪大学大学院工学研究科産業機械工学専攻前期課程 卒業生 窪西英俊君，ならびに大学院生 中橋昭久君，間渕暢浩君，岡島 豊 君，長谷宏明君をはじめ，システム設計工学講座の各位に感謝いたします．

関連発表論文

(1) 第3章に関する発表論文

(1-1) 論 文

・赤木・藤田，船舶基本設計に関するエキスパートシステムの構築，関西造船協会誌，no.206，(1987)，pp.65-75.

・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，日本機械学会論文集 C編，vol.54，no.500，(1988)，pp.1017-1025.

・赤木・藤田，ネットワークモデルによる設計過程の支援 ── エキスパートCADシステムにおける設計処理の機能 ── ，日本機械学会論文集 C編，vol.54，no.505，(1988)，pp.2300-2306.

・Akagi，S. and Fujita，K.，"Building an expert system for the preliminary design of ships"，AI EDAM，vol.1，no.3，(1987)，pp.191-205.

・Akagi，S. and Fujita，K.，"Building an Expert System for Engineering Design based on the Object-Oriented Knowledge Representation Concept"，Transactions of the ASME，Journal of Mechanisms，Transmissions and Automation in Design，(To be published).

(1-2) 国際会議

・Akagi，S. and Fujita，K.，"A Knowledge based Geometric Modeling System using Object-Oriented Approach"，Proceedings of the 1989 ASME Design Automation Conference，DE-19-1，(1989-9)，pp.129-134.

(1-3) 口頭発表

・赤木・藤田，オブジェクト指向にもとづく設計エキスパートシステムの研究 ── 船の基本設計システム ── ，日本機械学会 関西支部第62期定時総会講演会 講演概要集，No.874-1，(1987-3)，pp.140-141.

・赤木・藤田，オブジェクト指向形汎用設計エキスパートシステムの構築，日本機械学会・精密工学会 第5回設計自動化工学講演会 講演論文集，(1987-7)，pp.64-66.

・赤木・藤田，設計エキスパートシステムのためのインテリジェントターミナル，日本機械学会 関西支部第63期定時総会講演会 講演概要集，

No.884-1, (1988-3), pp.191-192.

・赤木・藤田，船の基本設計に関するエキスパートシステム，精密工学会・日本機械学会 第６回設計自動化工学講演会講演論文集，(1988-7)，pp.16-18.

・赤木・藤田，ＡＩ手法に基づく概念設計のためのモデリング手法 ── ＡＩ手法と最適化手法のハイブリッド化 ── ，日本機械学会 第66期全国大会講演会 講演概要集，No.880-6，(1988-10)，pp.388-389.

・赤木・藤田，オブジェクト指向による機能設計のための形状モデリングシステムの研究，日本機械学会 関西支部第64期定時総会講演会 講演概要集，No.894-1，(1989-3)，pp.97-98.

・赤木・藤田，オブジェクト指向による機能設計支援システムの研究 ── 形状モデリングシステムとの融合 ── ，日本機械学会・精密工学会 第７回設計シンポジウム 講演論文集，(1989-7)，pp.43-45.

・赤木・間渕・藤田，航空機の基本設計に対する統合化設計支援システムの構築 ── 要目決定と形状処理との融合 ── ，日本機械学会 関西支部第65期定時総会講演会 講演概要集，(1990-3)，(講演予定).

(２) 第４章に関する発表論文

(2･1) 論 文

・赤木・藤田・窪西，プラント設計におけるエキスパートＣＡＤシステムの研究，日本機械学会論文集 Ｃ編，vol.54，no.497，(1988)，pp.228-233.

(2･2) 国際会議

・Akagi, S., Fujita, K. and Kubonishi, H., "Building an Expert System for Power Plant Design", Proceedings of the 1988 ASME Design Automation Conference, DE-14, (1988-9), pp.297-302.

・Akagi, S. and Fujita, K., "Application of Artificial Intelligence Technology to Power Plant Design - Case Studies for a Marine Power Plant and a Land-use Cogeneration Plant -", Proceedings of the 4th International Symposium on Marine Engineering Kobe '90, (1990-10), (To be published).

(2･3) 口頭発表

・赤木・窪西・藤田，プラント設計支援エキスパートシステムの開発研究，日本機械学会・精密工学会 第５回設計自動化工学講演会 講演論文集，(1987-7)，pp.67-69.

・赤木・窪西・藤田・長谷川・山本，プラント設計に関するエキスパートＣＡＤシステムの研究（コージェネレーションプラントへの適用)，日本機械学会 関西支部第63期定時総会講演会 講演概要集，No.884-1，(1988-3)，pp.193-194.

・赤木・中橋・藤田・長谷川・山本，コージェネレーションプラントに関する設計エキスパートシステムの研究，日本機械学会 関西支部第64期定時総会講演会 講演概要集，No.894-1，(1989-3)，pp.99-100.

・赤木・中橋・藤田，コージェネレーションプラントに関する設計エキスパートシステムの研究（続報)，日本機械学会 関西支部第65期定時総会講演会講演概要集，(1990-3)，(講演予定)．

(３) 第５章に関する発表論文

(3･1) 論 文

・赤木・藤田，制約指向に基づく基本配置設計支援システムの研究（第１報：制約指向による基本アルゴリズム)，日本機械学会論文集 Ｃ編，(投稿中)．

・赤木・藤田・仲戸川・井上，制約指向に基づく基本配置設計支援システムの研究（第２報：オブジェクト指向によるシステム構築)，日本機械学会論文集 Ｃ編，(投稿中)．

(3･2) 国際会議

・Akagi, S. and Fujita, K., "A Constraint-directed Methodology for Plant Layout Design", Proceedings of the International Conference on Manufacturing Systems and Environment - Looking Toward the 21st Century, (1990-5), (To be published).

(3･3) 口頭発表

・赤木・藤田・仲戸川・井上，プラントの基本配置設計に対する汎用支援手法の構築，日本機械学会 第67期全国大会講演会 講演概要集，No.890-50，(1989-10)，pp.417-418.

・赤木・藤田・仲戸川・井上，制約指向による原子力発電所の基本配置設計支援システム，計測自動制御学会 第15回システムシンポジウム第10回知識工学シンポジウム合同シンポジウム 講演論文集，(1989-10)，pp.345-350.
・赤木・藤田・長谷・仲戸川・井上，プラントの基本配置設計に対する汎用支援手法の構築（続報：原子力発電所の基本配置設計への適用)，日本機械学会 関西支部第65期定時総会講演会 講演概要集，(1990-3)，(講演予定)．

(4) その他の業績

・藤田・川谷・増淵，二浮体型波浪エネルギー変換システムにおける浮体形状及びその他のパラメータとエネルギー変換効率との関係，海洋科学技術センター 第2回波浪エネルギー利用シンポジウム，(1987-6)，pp.27-34.