

コンカレント・エンジニアリングにおける  
知的設計支援システムに関する研究

1993

村 山 長

## 論文要旨

最近、設計・生産の分野でコンカレント・エンジニアリングが注目されている。これは、製品設計と同時に製品のライフサイクルと関連する様々な特性(製造性、安全性など)を考慮に入れるものである。

コンカレント・エンジニアリングでは、普通、この特性を考慮に入れるため、設計者が設計対象のモデルを構築した後、そのモデルを用いて専門技術者(例えば、製造性を考慮に入れる場合は生産設計の技術者)が特性を調べる作業(以下、統合化作業)を実行する。その結果、設計対象に不具合が存在することが分かれば、設計者と専門技術者との相談によりモデルを修正し、再び統合化作業を実行する。一方、熟練した設計者になると、信頼性などの特性を頭の中で予測しながら、モデリング作業を行っている場合がある。つまり、熟練設計者の頭の中ではモデリング作業と並行して、統合化作業も行っていると考えられる。熟練設計者は、このような並行処理によって、モデリング作業の初期段階から、設計対象の不具合を発見し、それに対する適切な対策を施している。

本論文は、このような並行処理に基づく設計支援システムの実現を目的とする。これは、CADシステムを用いたモデリング作業と並行して、統合化作業用ツールを実行することにより、モデリング作業中の設計者に設計対象の不具合の存在等を指摘するものである。これにより、設計者はモデリング作業の段階で特性を考慮に入れることが可能となる。

本論文は6つの章からなり、その内容は以下の通りである。

第1章では、本論文の目的と背景を述べ、各章の内容を概説する。

第2章では、モデリング作業の段階で特性を考慮に入れるための新しい設計支援の概念を示す。まず、モデリング作業中の設計対象モデルを絶えず監視し、不具合を発見すれば即座にその情報を設計者に与える知的設計オブザーバの概念を提案する。次に、これを実現するために不可欠な並行処理の方法を提案する。これは、統合化作業を部分問題に分解し、モデリング作業と並行して処理

可能となった部分問題を順次処理するものである。また、この設計支援方法において、設計者にフィードバックすべき情報を検討する。

第3章では、挙動により構造などが変化する機械をモデリングできるCADシステムの開発について述べる。まず、この変化する機械をオブジェクト指向とペトリネットに基づいてコンピュータ内で表現する方法を示す。次に、このような機械のモデリングを行うために開発したCADシステムの構成・機能について述べる。そして、このシステムにおけるモデリング方法と挙動のシミュレーション方法を示す。

第4章では、故障シミュレーションに対して第2章で提案した概念を適用する。これは、第3章で示したCADシステムによるモデリング作業と並行して故障シミュレーションを実行するものである。このため、まず、故障シミュレーションを部分問題に分解し、CADシステムのモデルに対する操作と連動して部分問題を処理する方法を提案する。そして、開発したシステムの実行例を示し、モデリング作業中の設計対象の不具合の発見や、設計変更による故障伝播の変化の把握ができることを示す。

第5章では、モデリング作業との並行処理が不可能な統合化作業として、組立順序の生成を取り上げる。まず、並行処理が不可能であっても、修正設計を行った場合などには、第2章で提案した概念の適用が有効であることを示す。次に、組立順序生成の部分問題への分解がAND/ORグラフで表せ、これを探索することにより組立順序が得られることを示す。さらに、第2章の概念に基づくAND/ORグラフの探索アルゴリズムを提案する。このアルゴリズムにより、修正設計などで設計変更された設計対象に対しては、設計変更前の組立順序情報を用いて効率よく設計変更後の組立順序が得られる。

第6章では、結論として各章で得られた主要結果をまとめる。

# 目 次

第1章 緒論	1
第2章 並行処理による設計支援の基本概念	8
2.1 緒言	8
2.2 知的設計オブザーバ	9
2.3 統合化作業の分解による並行処理	12
2.3.1 統合化作業の分解	12
2.3.2 統合化作業のIs-a・Instance-of関係	15
2.3.3 部分問題のリアルタイム処理	17
2.3.4 解の統合のリアルタイム処理	18
2.4 設計者へのフィードバック情報	20
2.5 オブジェクト指向によるシステムの実現例	21
2.6 結言	25
第3章 オブジェクト指向に基づくCADシステムの開発	26
3.1 緒言	26
3.2 オブジェクト指向による設計対象の表現	27
3.2.1 機械の構造の記述	27
3.2.2 機械の状態変化の記述	30
3.2.3 ペトリネットによる挙動の表現	32
3.2.4 挙動のオブジェクト表現	35
3.2.5 設計対象モデル	36
3.3 CADシステム	38

3.3.1	システム構成	38
3.3.2	モデリング方法	39
3.3.3	モデリングの視点	43
3.3.4	挙動シミュレーション	44
3.4	システムの実行例	44
3.5	結言	50

## 第4章 製品モデリングと故障シミュレーションの並行処理

4.1	緒言	51
4.2	故障シミュレーション	52
4.2.1	入出力情報	52
4.2.2	故障シミュレーションの方法	55
4.3	リアルタイム故障シミュレーション	57
4.3.1	故障シミュレーションの分解	57
4.3.2	故障シミュレーションのリアルタイム処理	59
4.4	リアルタイム故障シミュレータの開発	63
4.5	システムの実行例	66
4.6	結言	71

## 第5章 組立順序生成の効率化

5.1	緒言	72
5.2	組立順序生成問題	73
5.2.1	限定条件	73
5.2.2	組立順序生成問題の分割	74
5.2.3	分解可能性の検証方法	76
5.3	AND/ORグラフ探索による組立順序の生成	81

5.3.1	探索空間の縮小化	81
5.3.2	組立順序の探索アルゴリズム	85
5.3.3	設計変更前のAND/ORグラフの利用	89
5.4	計算例	90
5.5	結言	98
第6章 結論		99
謝辞		103
参考文献		104
学術研究論文一覧		115

## 第1章 緒論

近年、製品の多様化、高度化、複雑化により設計作業量が増大する一方、企業間競争の激化により設計に要するリードタイムの短縮が求められるようになった。これにともない、設計現場に様々なシステムが導入され、設計作業の効率化が図られるようになった。

現在、設計現場で普及しているシステムは、設計の様々な作業と対応して、種々のものに分類できる<sup>(1)</sup>。その中で、中心的な役割を果たすものが、CAD (Computer Aided Design) システム、CAE (Computer Aided Engineering) システム、CAPP (Computer Aided Process Planning) システム、CAM (Computer Aided Manufacturing) システムである。CADシステムは設計対象のモデリングや設計図面の作成を支援するシステムである。CAEシステムはモデルを用いて設計対象の特性（例えば、信頼性、機能性など）を解析・シミュレーションにより調べ、モデルの評価・検証を行うものである。また、CAPPシステム、CAMシステムはモデルを基に製造準備を行うシステムである。なお、CAEシステムは広義には設計から生産に至る全ての活動を支援するシステムであり、CAD、CAPP、CAMの機能を包含するものであるが、ここでは狭義の解析・シミュレーションによる特性評価ツールとして捉える<sup>(2)</sup>。また、CAMシステムも本来は製造支援システムであるが、通常は製造のための作業設計を支援するシステムとして位置づけられており、ここでもこの意味で用いる。

これらのシステムはコンピュータのハードウェアとソフトウェアの両技術の進歩により飛躍的に普及したが、それに伴い、現状のシステムのもつ問題点も次第に明らかになった<sup>(3)</sup>。以下に主な問題点を示す。

#### (1) 支援範囲の問題

設計作業には、一般に、概念設計、基本設計、詳細設計、工程設計、作業設計の段階があるが、現在のシステムは概念設計、基本設計といった設計の上流作業を支援できない。この原因には、これらの上流作業が知的作業であることや、これらの段階では設計対象の非形状情報の操作が中心であるが現在のシステムではこれを扱えないこと、などがある。

#### (2) 問題解決能力の問題

現在のシステムが支援可能な段階に対しても、十分に設計作業をサポートしているとはいえない。すなわち、現在のシステムは計算、作図、データ検索などの単純作業のみを処理しており、問題解決能力に著しく欠ける。設計者への適切なアドバイス機能などの高度な知的機能の実現が望まれる。

#### (3) 自動化の島の問題

設計現場には多種のシステムが存在するが、これらのシステムはそれぞれ様々なデータを扱う。これらのデータの表現形式はシステムによって異なるため、システム間でデータの共有や交換ができないことが少なくない。これは自動化の島<sup>(4)</sup>の問題と呼ばれ、設計作業の効率化のため、その解決が急務とされている。

これらの問題を解決するため、次世代システムの研究が盛んに進められている。この研究には2つの方向がある。

一つは、システムの知的化である。この知的化には以下の事項が含まれる。

- (a) これまで人間が行っていた知的な作業を処理できる高度な問題解決能力をシステムにもたせる。
- (b) 設計対象の不具合の存在の指摘や、設計改善策の提案など、設計者に対してアドバイスを行う機能をシステムにもたせる。

(c) 自然言語の理解や設計者の意図の把握など、設計者との対話を良好にする能力をシステムにもたせる。

これらを実現するため、人工知能やその応用であるエキスパート・システムの手法などの導入が盛んに試みられている<sup>(3)-(24)</sup>。これは、二つのアプローチに大別できる。一つは、特定の機械を対象とした設計エキスパート・システムの構築のようなアドホックなアプローチ<sup>(12)-(20)</sup>である。もう一つは、人間の設計行為を解明し、それを基に人工知能を用いて知的設計支援の方法論を構築する理論的アプローチ<sup>(21)-(24)</sup>である。後者は、特に最近、設計の科学的解明なしに知的システムを構築できないという立場から、活発に研究されている<sup>(3)</sup>。

もう一つの方向はシステムの統合化である。システム間の物理的な統合は LAN (Local Area Network) などのネットワークによって既に実現されており、また、論理的な統合も設計対象の2次元幾何情報のシステム間交換に対しては、IGES (Initial Graphics Exchange Specifications)<sup>(25)</sup>などの規格により実現されている。最近、さらに3次元幾何情報および非幾何情報の交換も可能にするため、CAD\*I<sup>(26)</sup>やSTEP (Standard for the Exchange of Product model data)<sup>(27)</sup>などの国際規格が検討されている。また、複数のシステムのデータベースをオブジェクト指向に基づいて統合的に管理する研究<sup>(28)</sup>や、複数のモデル間でデータの整合性を管理する研究もなされている<sup>(29)</sup>。

最近、この統合化と関連する概念としてコンカレント・エンジニアリング<sup>(30)-(34)</sup>が注目されている。これは、製品のライフサイクルと関わる全ての要素(品質、コスト、製造、保全、廃棄など)を、製品設計と同時に考慮させるものである<sup>(30)</sup>。サイマルテニアス・エンジニアリング<sup>(35)-(37)</sup>や木村らのフレキシブル・インテグレーション<sup>(38)</sup>はこれと同じことを意味する。現在は特に、製品設計の初期段階から製造性を考慮に入れる研究が盛んに行われている。この場合、このアプローチは製造(組立)のための設計<sup>(39)-(46)</sup>と呼ばれる。

コンカレント・エンジニアリングでは、製品の製造や保全などを考慮に入れるため、製品設計の段階で、設計対象の製造性、保全性などの特性を調べる作

業（以下、統合化作業と呼ぶ）が実行される。このため、設計対象モデルを構築するCADシステムと統合化作業用ツールの統合化が要求され、また、実践されている。この統合化作業用ツールとして、既存のCAEシステム、CAPPシステム、CAMシステムを利用する<sup>(47)</sup>ほか、組立性評価ツールなどのツールを新たに開発する場合<sup>(42),(48)-(60)</sup>もある。また、このようなツールを設計者自身が使う場合もあるが、多くの場合、その特性評価の専門技術者（たとえば、製造性評価の場合は生産設計の技術者）が用いる。後者の場合は、設計者と専門技術者間での協調作業が必要となる。この協調をいかに実現するかがコンカレント・エンジニアリングの研究テーマの一つをなしている<sup>(61),(62)</sup>。

なお、コンカレント・エンジニアリングの研究では、このようなCADシステムと統合化作業用ツールとの統合化に基づく研究以外に、製品設計と工程設計を統合的に最適化する研究<sup>(63)</sup>がある。しかし、この方法では、設計者が製品設計と工程設計の両知識に精通していなければ、最適化のための問題の記述が困難である。また、設計エキスパート・システムの研究と同様に、ヒューリスティックな特性評価用知識を設計支援システムに組み込む研究<sup>(64)</sup>もなされているが、特性評価用知識が特定の対象に特殊化されてしまい、汎用性に欠けるという問題がある。また、評価用知識がヒューリスティックであるため、解析・シミュレーションなどによる場合に比べ、評価結果の信頼性が劣るという問題もある。これらのことから、製品設計段階で様々な特性を考慮に入れるためには、CADシステムと統合化作業用ツールの統合化が必要である。

以上、設計用システムには知的化と統合化の2つの方向があることを示したが、コンカレント・エンジニアリングに関連する研究の中には、これらの方向を融合し、システムの統合化による知的化を実現するものがある。E. Molloy et al.<sup>(42)</sup>やC. C. Hayes et al.<sup>(47)</sup>は、CADシステムと統合化されたCAPPシステムを用いることにより、設計対象の製造上の問題点を見つけ出し、製造コスト削減のためのアドバイスを行う方法を提案している。また、M. Supossnek et al.<sup>(36)</sup>はCASE ( Computer Aided Simultaneous Engineering ) プロジェクトにおいて、統

合化による知的支援の枠組を提案し、ウインド・レギュレータの設計に応用している。これらの研究は、CADシステムと統合化作業用ツールを統合することにより、CADデータを用いて設計対象の様々な特性を調べ、その結果に基づいて設計者を知的に支援するものである。この知的支援は、上述のシステムの知的化の事項のうち(b)の一部を実現するものであり、非熟練設計者のサポートという面で特に重要である。

このようなコンカレント・エンジニアリングにおける統合化による知的化の研究では、設計者がCADシステムを用いてモデルを構築した後、専門技術者または設計者自身が統合化作業用ツールを実行し特性を評価する。その結果、設計対象に不具合が存在することが分かれば、システムから与えられたアドバイス情報を基に設計者と専門技術者との相談によりモデルを修正し、再び統合化作業を実行する。このような反復処理は不具合がなくなるまで行われる。一方、熟練した設計者になると、信頼性などの統合化作業の結果を頭の中で予測しながら、モデリング作業を行っている場合がある。つまり、熟練設計者の頭の中ではモデリング作業と並行して、統合化作業もある程度実行していると考えられる。熟練設計者は、このような並行処理によって、モデリング作業の初期段階から、設計対象の不具合を発見し、それに対する適切な対策を施している。

本論文は、このような並行処理に基づく設計支援システムの実現を目的とする。これは、CADシステムを用いたモデリング作業と並行して、統合化作業用ツールを自動的に実行することにより、モデリング作業中の設計者に設計対象の不具合の存在等を指摘するものである。これにより、設計者はモデリング作業の段階で特性を考慮に入れることが可能となる。このため、従来のコンカレント・エンジニアリングで行われていた設計者と専門技術者との協調が不要になる可能性がある。また、協調が必要であっても、設計者が構築したモデルに不具合が内在する可能性が低くなるため、上述のようなモデリング作業と総合化作業との反復処理が行われる可能性が低くなり、設計に要するリードタイムが短縮される。

本論文の構成を次に示す。

第2章では、モデリング作業の段階で特性を考慮に入れるための新しい設計支援の概念を示す。まず、モデリング作業中の設計対象モデルを絶えず監視し、不具合を発見すれば即座にその情報を設計者に与える知的設計オブザーバの概念を提案する。次に、これを実現するために不可欠な並行処理の方法を提案する。これは、統合化作業を部分問題に分解し、モデリング作業と並行して処理可能となった部分問題を順次処理するものである。これにより、モデリング作業の任意の時点で設計対象の特性を知ることが可能となり、また、モデリング作業時のモデルの変更に対応する特性の変化を知ることが可能となる。そして、この設計支援方法において、設計者にフィードバックすべき情報を検討する。最後に、簡単なモデリング作業と統合化作業を対象に、オブジェクト指向に基づいてシステムを実現する方法を示す。

第3章では、挙動により構造などが変化する機械をモデリングできるCADシステムの開発について述べる。まず、この変化する機械をオブジェクト指向とベトリネットに基づいてコンピュータ内で表現する方法を提案する。次に、このような機械のモデリングを行うために開発したCADシステムの構成・機能について述べる。そして、このシステムにおけるモデリング方法と挙動のシミュレーション方法を示す。最後に、このシステムを用いたモデル構築例、および、挙動のシミュレーション例を示す。

第4章では、故障シミュレーションに対して第2章で提案した概念を適用する。これは、第3章で示したCADシステムによるモデリング作業と並行して故障シミュレーションを実行することにより、モデリング作業中の設計者を信頼性の立場から支援するものである。まず、制約伝播に基づく故障シミュレーションの方法を提案する。次に、故障シミュレーションを部分問題に分解し、CADシステムのモデルに対する操作と連動して部分問題を処理する方法を提案する。そして、CADシステムと統合化されたリアルタイム故障シミュレータについて述べる。最後にシステムの実行例を示し、モデリング作業中に設計対象

の不具合の発見や、設計変更による故障伝播の変化の把握が可能になることを示す。

第5章では、モデリング作業との並行処理が不可能な統合化作業として、組立順序の生成を取り上げる。まず、モデリング作業と組立順序生成との並行処理が不可能であるが、修正設計を行った場合などには、第2章で提案した概念の適用が有効であることを示す。次に、組立順序生成の部分問題への分解がAND/ORグラフで表せ、これを探索することにより組立順序が得られることを示す。そして、第3章で示したCADシステムで作成されたデータを用いた各部分問題の処理方法を提案する。さらに、組立順序探索の効率化のためにヒューリスティクスによるAND/ORグラフの縮小方法を示し、第2章の概念に基づくAND/ORグラフの探索アルゴリズムを提案する。このアルゴリズムにより、設計変更された設計対象に対しては、設計変更前の組立順序情報を用いて効率よく設計変更後の組立順序が得られる。最後に、CADシステムと統合化された組立順序生成システムについて述べ、実行例により組立順序が効率よく得られることを示す。

第6章では、結論として各章で得られた主要結果をまとめる。

## 第2章 並行処理による設計支援の基本概念

### 2.1 緒言

設計支援システムの将来像が盛んに議論されている。畑村<sup>(65)</sup>は、将来、システムが具備すべき機能として設計対象の監視機能を挙げている。これは、設計者がシステムを用いて設計作業を行う場合に、システムが能動的に作動して設計対象の機能性、製造性などの特性上の不具合などを検出し、設計者に適切なアドバイスを与えるものである。この機能は、特に非熟練設計者のサポートに有用である。この機能を実現する研究として、特定の機械を対象とした設計エキスパート・システムの研究<sup>(19),(20),(48)</sup>があるが、不具合検出などのための知識を対象の種類に特化した形式で予め記述しておかなければならないという問題がある。

本章では、この監視機能をモデリング作業と設計対象の特性を調べる統合化作業との並行処理に基づいて実現する方法を示す。これは、並行処理によって、モデリング作業の任意の時点で得られる統合化作業の結果から不具合などを検出し、それを基に適切な支援を設計者に行うものである。まず、2.2節ではモデリング作業中の設計対象モデルを絶えず監視する知的設計オブザーバの概念を示す。次に、2.3節では、これを実現するために必要な並行処理の方法を示す。そして、2.4節では、モデリング作業中の設計者にフィードバック

すべき情報を検討する。最後に、2.5節では、簡単なモデリング作業と統合化作業を対象に、オブジェクト指向に基づいてシステムを実現する方法を示す。

## 2.2 知的設計オブザーバ

設計者は、設計対象の信頼性、製造性などの特性を考慮して設計する必要がある。しかし、非熟練設計者にとってこれは困難である。そのため、設計支援システムにこの特性の考慮を可能にさせる機能が必要となる。コンピュータを用いたモデリング作業を特性の観点から知的に支援するため、知的設計オブザーバの概念を提案する。これは、以下の要件を満足するシステムである。

- (1) システムは、モデリング作業の任意の時点で、その段階の設計対象に対応する統合化作業の結果を設計者に与えることができる。
- (2) システムは、モデリング作業の途中であっても、統合化作業の結果から設計対象の不具合を発見すれば、即座にその情報を設計者にフィードバックする。
- (3) システムは、設計者のモデリング作業を中断することなく、統合化作業を実時間で処理できる。
- (4) 設計者が設計変更を行なった場合、システムはその変更が設計対象の特性に及ぼす影響を推定し、設計者に知らせることができる。

この設計支援システムの概念では、統合化作業の実行や設計対象の不具合の検出など、特性を考慮に入れるために必要な処理はシステムが行うため、設計者の負担が軽減される。なお、不具合が存在する場合、適切な設計改善案を導出することが理想的であるが、このための汎用的に有効な方法を確立することは現段階では不可能に近く、本研究の対象外とする。

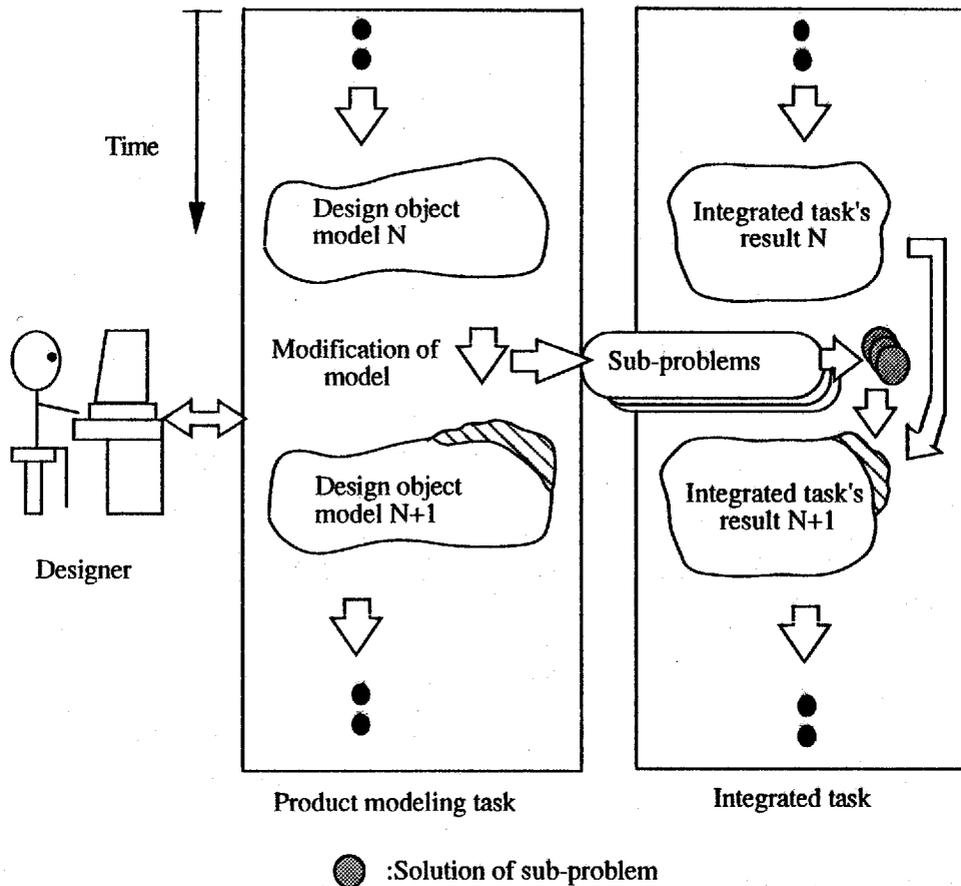


Fig. 2.1 Concurrent processing of product modeling task and integrated task

(1)から(4)の要件を満足させるための方法を以下に示す。これは、統合化作業を部分問題に分解し、モデリング作業と並行して処理可能になった部分問題を次々と処理する方法である。この方法を以下ではリアルタイム設計処理と呼ぶ。Fig. 2.1にこの方法の概念図を示す。この図に示すように、リアルタイム設計処理では、モデリング作業中に設計対象モデルが部分的に変更され、それによって新たに処理可能となった部分問題が発生すると、即座にそれが処理される。この処理によって得られた部分問題の解は、それ以前に存在していた統合化作業の結果との統合が可能であれば統合される。これにより、変更された設計対象モデルに対応する統合化作業の結果が得られる。なお、この並行処理

の方法については次節で詳細に検討する。

このような並行処理により、モデリング作業の途中であっても、部分問題の解や統合化作業の結果から、その時点の設計対象に内在する不具合を発見できる。また、設計対象モデルが部分的に変更された場合、変更されたモデル全体に対して統合化作業を再実行するのではなく、新たに発生する部分問題のみを処理する。このため、モデリング作業を中断することなく、実時間で処理結果が得られる。また、モデルの変更によって新たに処理された部分問題の解や統合化作業の結果の変化から、その変更による影響を推定できる。以上のことから、リアルタイム設計処理は、要件(1)~(4)を満足できる方法であることが分かる。

このようなリアルタイム設計処理の方法が適用可能であるためには、以下の条件を満足する必要がある。

- (a) 統合化作業をコンピュータによって自動的に処理できる。
  - (b) 統合化作業を部分問題に分解できる。
  - (c) 分解された部分問題がモデリング作業の進行に伴って、順次に処理可能となる。
  - (d) モデリング作業中に処理可能となった部分問題は、実時間で処理できる。
- 統合化作業の中には、条件(a),(b)を満足するが、(c),(d)を満足しないものがある。例えば、有限要素法を用いた解析では、設計対象の幾何情報以外に解析を行うための種々の属性情報を必要とするが、この属性情報はモデリング作業の最終段階に決定されるため、その全ての部分問題はその段階まで処理できない。このような場合、モデリング作業と統合化作業の並行処理は不可能である。しかし、修正設計などによって設計対象のモデルを得るような場合には、リアルタイム設計処理の適用は有効である。この場合、修正前の統合化作業の処理結果が存在すれば、修正によって新たに発生する部分問題のみを処理するだけで、効率よく修正後の統合化作業の処理結果が得られる。

なお、ここでは統合化作業として、設計対象の特性を調べる順問題を対象と

し、最適化手法等により設計パラメータを決定する逆問題を扱わない。

また、統合化作業の中には、モデリング作業によって作成した設計対象モデルを統合化作業用のモデルに変換する必要があるものがある。この場合は、モデルの変換作業を一つの統合化作業と考え、これに対してもリアルタイム設計処理の概念を適用する。これにより、モデリング作業、モデル変換作業、統合化作業の三つの作業の並行処理が可能になる。

## 2. 3 統合化作業の分解による並行処理

### 2. 3. 1 統合化作業の分解

統合化作業の部分問題への分解は Fig. 2.2 に示すように進められる。この図のように統合化作業Aは設計対象モデルDの部分情報DAを入力として実行される。また、統合化作業Aの実行により情報集合Rが生成される。ここでは、この統合化作業Aの実行を次のように表わす。

$$A = (DA, F, R) \dots \dots \dots (2.1)$$

ただし、Fはこの統合化作業の実行手続きを表わす。

リアルタイム設計処理が適用可能な統合化作業は、複数の部分問題に分解される。この分解は次式で表わせる。

$$\left. \begin{array}{l} P1 = (d1, f1, r1) \\ P2 = (d2, f2, r2) \\ \vdots \\ Pi = (di, fi, ri) \\ \vdots \\ PN = (dN, fN, rN) \end{array} \right\} \dots \dots \dots (2.2)$$

$$C = ((r1, r2, \dots, rN), fc, R) \dots \dots \dots (2.3)$$

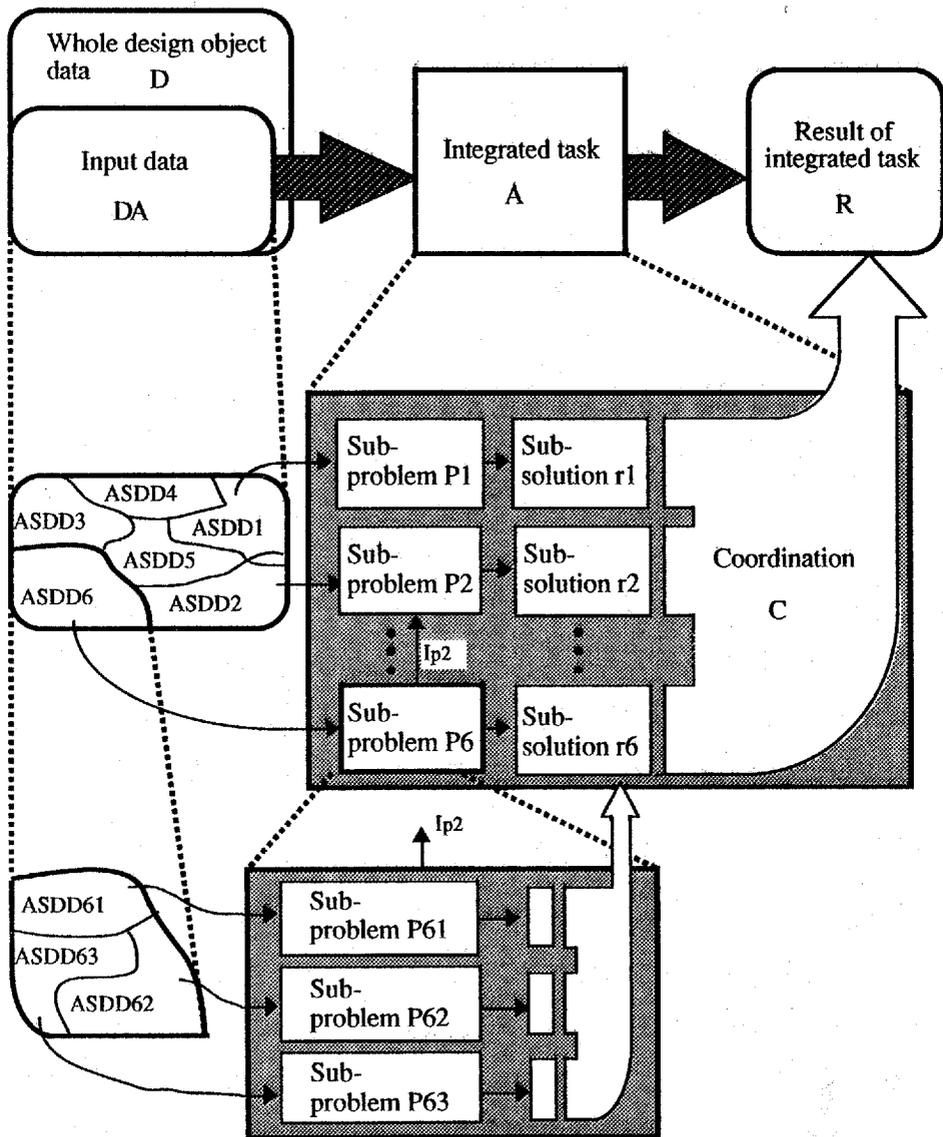


Fig. 2.2 Decomposition of integrated task

ここで、 $P_i (i = 1, 2, \dots, N)$  は分解された部分問題を表わす。  $f_i$  は部分問題  $P_i$  の処理手続きであり、  $d_i$  は部分問題  $P_i$  を処理するための入力情報の集合である。  $r_i$  は部分問題  $P_i$  を処理した結果、生成される出力情報の集合であり、部分解と呼ぶ。  $C$  は部分解の統合を表し、手続き  $fc$  により、統合化作業の結果  $R$  (以下、統合解と呼ぶ) が得られることを示している。この部分解の統合には、部分解とは別種の情報を統合解として生成する場合や、部分解の直和を統合解とする場合など様々である。統合化作業の分解に従い、統合化作業の入力情報集合  $DA$  もこの図のように分解される。ここで、

$$DA = \cup_i ASDD_i \dots \dots \dots (2.4)$$

である。ただし、  $\cup_i ASDD_i$  は全ての  $ASDD_i$  ( Application Subset of Design object Data ) の和集合を表わす。このように設計対象に関する情報は、統合化作業の実行という立場から分解することができる。このことは小野里ら<sup>(66)</sup>によっても指摘されている。部分問題  $P_i$  は、このような  $ASDD_i$  と他の部分問題からの干渉入力  $I_{pi}$  の集合  $I_{pi}$  を入力とする。すなわち、

$$d_i = ASDD_i \cup I_{pi} \dots \dots \dots (2.5)$$

分解された部分問題は、この図の  $P_6$  のように、さらに小さい部分問題に分解できる場合がある。このように分解された部分問題と統合化作業とは、オブジェクト指向<sup>(67)</sup>の概念における Part-of 関係にある。これは、統合化作業  $A$  が部分問題  $P_1 \sim P_6$  から構成され、  $P_6$  はさらに  $P_{61} \sim P_{63}$  から構成されることを意味する関係である。

以上のように階層的に分解される部分問題は、式(2.5)のように設計対象モデルの一部の情報を入力とするため、モデル全体が作成される前でも処理できる可能性がある。

### 2. 3. 2 統合化作業の Is-a・Instance-of 関係

前節で、統合化作業とその部分問題とが Part-of 関係にあることを示したが、同様に Instance-of 関係および Is-a 関係も存在する場合がある。Instance-of 関係は、実際に処理される統合化作業やその部分問題（インスタンス）とそれらの種類（クラス）との関係である。Fig.2.3の例では、部分問題インスタンス1は部分問題クラス1に属する具体問題となっている。図に示されているように、このような関係は、その入出力情報にも存在する。すなわち、部分問題インスタンス1への入力ASDDインスタンス1は、部分問題クラス1への入力ASDDクラス1とInstance-of関係にある。また、出力である部分解インスタンス1と部分解クラス1との間にも Instance-of関係が存在する。

次に、Is-a関係の例をFig.2.4に示す。図において、統合化作業クラスxとクラスyの間にIs-a関係が存在する。これは、統合化作業クラスxが、作業の抽象度においてクラスyより上位にあることを表す。すなわち、両作業クラスは同じ特性を調べる方法を与えるが、上位クラスxの方法が下位クラスyの方法より総合的で簡略化された手続きとなっている。その入出力情報においても、上位のものは下位より抽象的になっている場合がある。また、上位と下位の統合化作

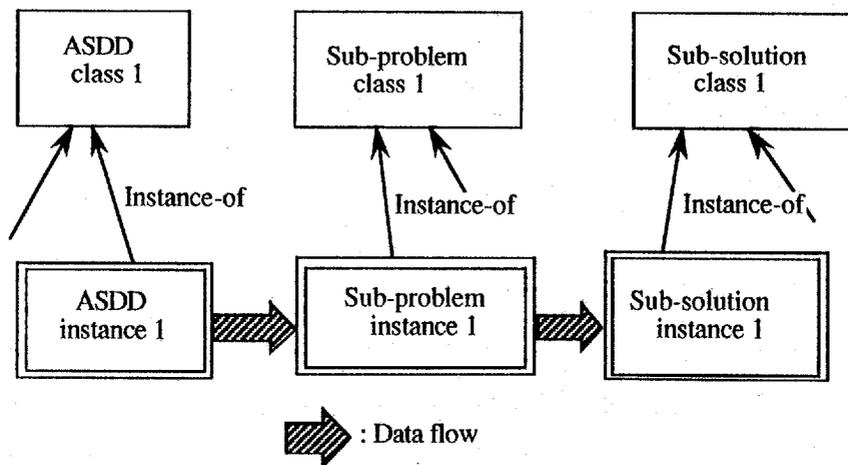


Fig. 2.3 Instance-of relationship for sub-problem, input data and output data

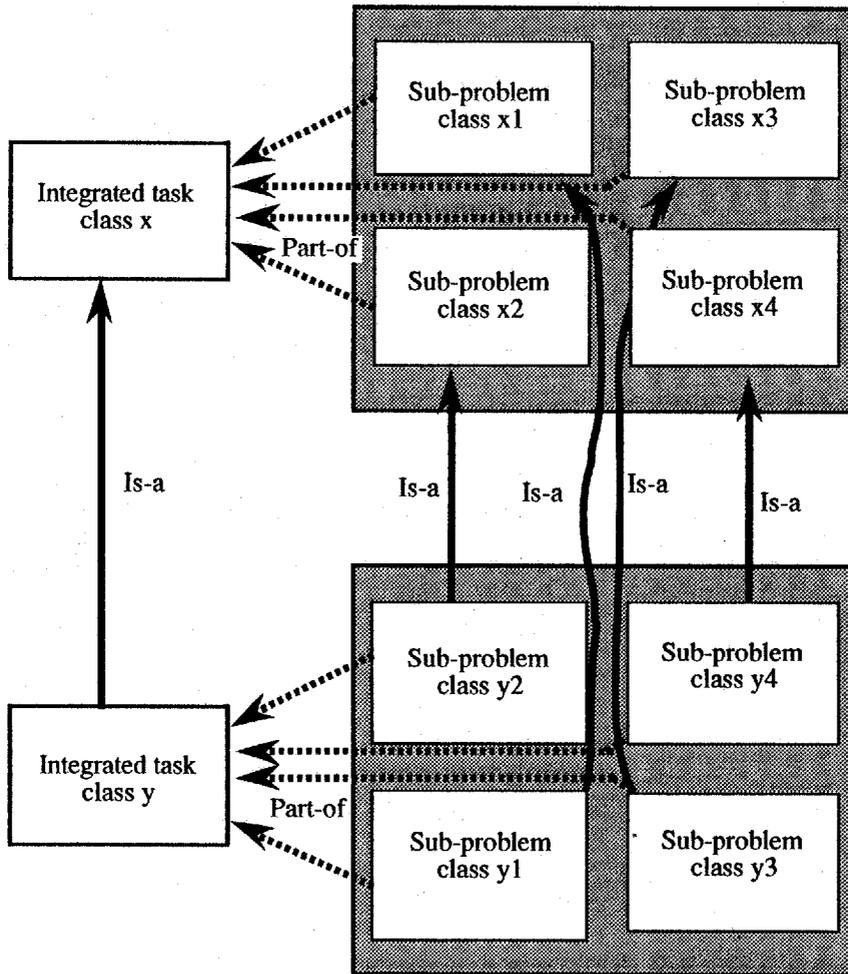


Fig. 2.4 Is-a relationship among integrated tasks and these sub-problems

業で部分問題への分解方法が同じであれば、分解される部分問題間にも、上位と下位で図に示されるようなIs-a関係が存在する。このようなIs-a関係にある統合化作業およびその部分問題間には、その処理に用いられる知識において、“深い・浅い”という関係<sup>(68)</sup>があると考えられる。なお、このような上位の統合化作業とその部分問題は、次の場合に処理される。

(1) 設計初期段階で、下位の統合化作業やその部分問題を処理するのに十分な情報が揃っていないが、上位のものであれば処理できる。

(2) 下位の統合化作業の厳密な手続きで処理する必要がない。

このうち、(1)の場合、設計対象の情報が十分詳細化されると下位の統合化作業やその部分問題も処理可能となる。この時、上位のものがヒューリスティクスを用いて処理される場合は、上位の解と下位の解との間で矛盾が起こる可能性がある。これは、ヒューリスティクスが必ずしも正しい解を生成するとは限らないためである。このため、この場合は下位の解を生成した時に上位の解を削除する必要がある。

### 2. 3. 3 部分問題のリアルタイム処理

知的設計オブザーバでは、部分問題がモデリング作業中に処理可能な状況になれば、即座に処理される。ここで、処理可能な状況とは、モデリング作業中に、部分問題の処理に必要な入力情報集合が揃った時である。今、部分問題  $P_i$  の入力情報集合  $d_i$  が以下のような情報  $x_{ij}$  の集合として表せるとする。

$$d_i = \{ x_{ij} : x_{ij} \in ASDD_i \text{ または } x_{ij} \in I_{pi} \} \dots \dots \dots (2.6)$$

ここで、情報  $x_{ij}$  の存在を表す 2 値変数を次式で定義する。

$$Q(x_{ij}) = \begin{cases} 1; \text{情報 } x_{ij} \text{ が存在する} \\ 0; \text{情報 } x_{ij} \text{ が存在しない} \end{cases} \dots \dots \dots (2.7)$$

部分問題  $P_i$  の処理可能条件  $QC(P_i)$  は、このような 2 値変数を元とする以下の論理式で表せる。

$$QC(P_i) = \bigwedge_j Q(x_{ij}) \dots \dots \dots (2.8)$$

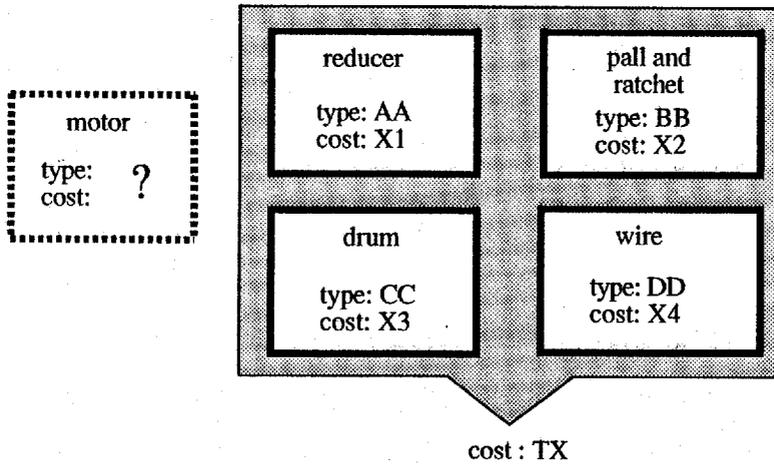
ただし、 $\bigwedge_j Q(x_{ij})$  は全ての  $j$  についての  $Q(x_{ij})$  の論理積を表すものとする。 $QC(P_i)$

が 1 であれば、部分問題は処理可能である。モデリング作業中に、ある情報  $x_p$  が生成されると、 $Q(x_p)$  を処理可能条件の元としてもつ部分問題に対し、処理可能条件を満足するかどうかチェックされる。条件を満足すれば、即座にその部分問題を処理する。その結果、得られた部分解がさらに他の部分問題の入力情報となり、部分問題が連鎖的に処理される場合もある。また、モデリング作業中に情報が削除された場合は、それを入力情報としていた部分問題の部分解を削除する。また、既に処理された部分問題の入力情報に変更があった場合は、その部分解を削除し、部分問題を再処理することになる。

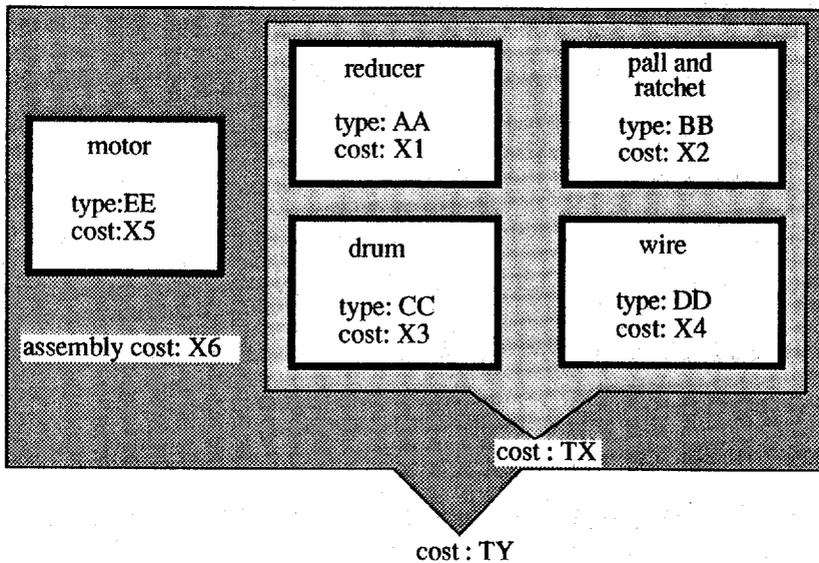
#### 2. 3. 4 解の統合のリアルタイム処理

部分解の統合は、モデリング作業中に、式 (2.3) で示した部分解の組 ( $r_1, r_2, \dots, r_N$ ) が揃えば即座に実行される。ただし、Fig. 2.5 に示すように、部分解の組が揃っていない場合でも、解の統合が可能な場合がある。この図は、編集設計によって得られる設計対象の製造コストを求める統合化作業の例を示している。この作業は、各構成要素の製造コストを求める部分問題と、組立コストを求める部分問題に分解でき、部分解の統合により設計対象全体の製造コストが求まる。この例における編集設計は、設計対象の基本構成を決めた後、各構成要素を選定し、それぞれの型式を決定することにより進められる。Fig. 2.5 (a) は設計途中の設計対象を表している。この設計対象では、モータの型式が未決定であるため、その製造コストは分からないが、他の構成要素の型式は決定済みでそれらの製造コストが分かるため、それらの合計値を算出できる。これは決定済み部分の部分解を統合し、その部分の統合解を求めたことになる。得られた製造コストの合計値が要求条件を満足していなければ、その時点で設計対象モデルを修正できる。Fig. 2.5 (b) のようにモータの型式が決定され、その製造コストと組立コストが求まると、それらを Fig. 2.5 (a) で求めた製造コストの合計値に加算することになる。これは、統合解の修正を行ったことになる。同様に、モデリング作業中に部分解が削除された場合や、部分解が再計算された場

合も、それ以前に存在していた統合解を修正することによって、その段階の設計対象に対応する統合解が得られる。なお、部分解の統合方法や統合解の修正方法は、統合化作業の種類によって異なる。



(a) Coordination of sub-solutions (manufacturing costs of components)



(b) Modification of coordinated cost (manufacturing cost of whole design object)

Fig. 2.5 Real-time coordination of sub-solutions

## 2. 4 設計者へのフィードバック情報

知的設計オブザーバでは、モデリング作業と統合化作業とを並行処理するが、この時、どのような状況に、どのような情報を設計者にフィードバックするかが重要である。設計者にフィードバックすべき情報を以下に示す。

### (1) 設計対象の不具合に関する情報

モデリング作業中に、統合化作業の結果から設計対象の不具合が発見されれば、その時点で即座に、その情報を設計者に知らせる必要がある。これにより、モデリング作業の初期段階から、不具合を発見でき、それに対する対策を早く施すことが可能となる。

不具合が存在すると判断されるのは、設計対象内に明らかに誤った部分が存在する場合や、信頼性や製造性などの特性が要求条件を満足しなくなった場合などである。この情報は、設計者に警告としてフィードバックされる必要がある。しかし、情報の欠落によって不具合が存在すると判断される場合には、次のような問題がある。それは、モデリング作業の初期段階で、設計対象に関する情報が十分に決定されていないために、情報の欠落による不具合があると判断されてしまうというものである。このため、情報の欠落による不具合は、設計者には警告としてではなく、参照情報として提示する必要がある。

### (2) 設計変更による影響に関する情報

部品やユニットの置換のように、設計者が明らかに設計変更を意図したと考えられる場合は、その設計変更が設計対象の特性に及ぼす影響を設計者にフィードバックする。この影響は、設計変更によって新たに処理された部分問題の解や統合化作業の結果（特性）の変化から推定できる。設計者が信頼性の改善のように、ある特性の観点から設計変更をした場合は、設計変更がその特性に及ぼす影響をフィードバックすることにより、設計者は意図通りの結果が得られたかどうかを確認できる。また、それ以外の特性に及ぼす影響もフィードバックすることにより、設計者は設計変更の副作用として起こりうる状況を把握

できる。

## 2. 5 オブジェクト指向によるシステムの実現例

本方法の有効性を示すため、簡単なモデリング作業と統合化作業を対象としたシステムの実現例を示す。対象とするのは、設計対象の基本構成のモデリング作業とモデルを用いて設計対象の故障率を調べる作業である。ただし、簡単のため、設計対象を直列システムで、設計対象の構成要素と信頼性ブロック図の節点が一対一に対応するものに限定する。これにより、構成要素の故障率の和を求めることにより、設計対象の故障率を算出できる。

オブジェクト指向に基づいて実現したシステムの構成を Fig.2.6 に示す。この図で、Sub-problem solver (以下、SPS) は設計対象の各構成要素の故障率を求める部分問題を処理するオブジェクトであり、Solution coordinator (以下、SC) は部分問題の解（構成要素の故障率）を統合するオブジェクトである。各SPSには、部分問題の処理可能条件をチェックする手続き、および、故障率モデルを用いて故障率を求める手続きがそれぞれメソッドとして記述されている。なお、故障率モデルは構成要素の種類によって異なるため、構成要素の種類毎にSPSが構成されている。

モデリング作業中に設計対象モデルへの構成要素の追加、または構成要素に関する情報の変更があると、その構成要素と対応するSPSにメッセージが送信される。受信したSPSは、部分問題の処理可能条件をチェックし、条件を満足すれば部分問題を処理するメソッドを実行し、その構成要素の故障率を計算する。そして、SPSは、計算結果をSCに送信する。SCは、求めた構成要素の故障率とモデル変更前に存在していた設計対象全体に対する故障率とを統合し、変更後の設計対象全体に対する故障率を求める。また、このシステムでは、このSPSとIs-a関係の上位にあるSPSも構成できる。この上位のSPSは、ヒューリスティクスを用いて、各構成要素の信頼性を環境条件を考慮に入れて定性的に

調べるものである。この上位のSPSにより、構成要素が十分に選定されておらず故障率を求めることができない場合でも、定性的な信頼性を知ることができる。ここで、定性的な信頼性とは、例えば、ある種の部品をある環境条件下で使用すると設計対象の信頼性が非常に低くなる、といったものである。

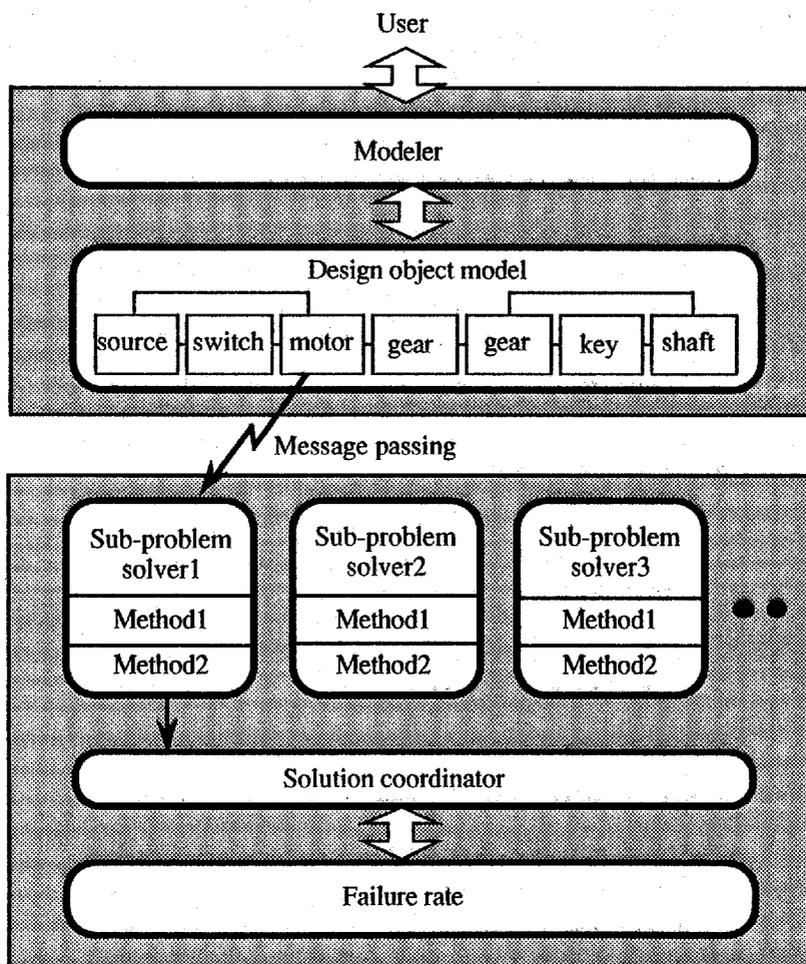


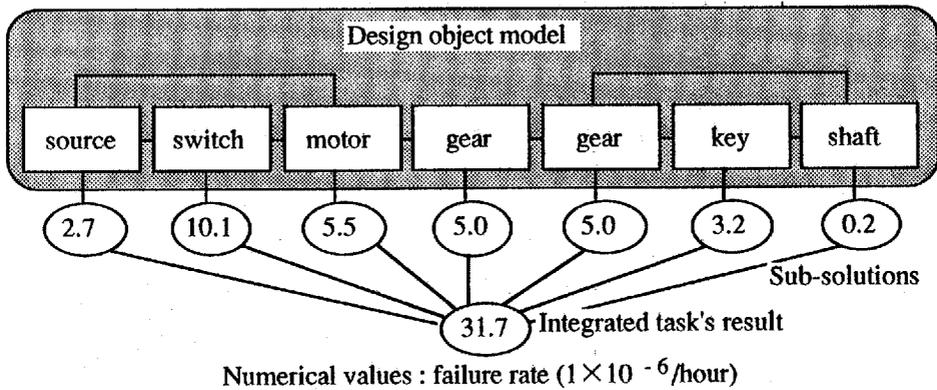
Fig. 2.6 System construction

このシステムによるモデリング途中の対話例を Fig. 2.7 に示す。この図では、設計者がモータを誘導モータに具象化したところ、設計対象全体の故障率が要求条件を満足しなくなり、設計者に警告が与えられたことが示されている。また、Fig.2.8にこの具象化による設計対象モデルと統合化作業の結果の変化を示す。この図で斜線部は再計算された部分解と統合解を表す。この図では、モータを誘導モータに具象化したため、誘導モータの故障率を求める部分問題が新たに処理され、その部分解 ( $10.2 \times 10^{-6}/\text{hour}$ )を用いて統合解が修正されたことが示されている。このように設計者は、設計対象モデルを完成させる前に、要求条件が満足されないことを知ることができる。

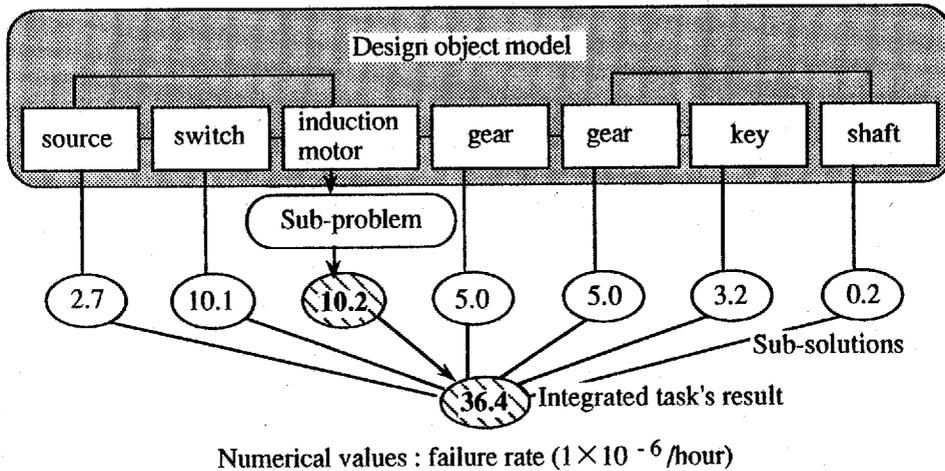
```
?- specialization(p(3)).
  I am retrieving candidates to specialize p(3) : motor.
  Select one from following parts :
  1 ; dc_motor
  2 ; synchronous_motor
  3 ; ac_commutator_motor
  4 ; induction_motor
    : 4.

----- warning -----
The failure rate of design object is 36.4 (10-6/hour ).
This can not satisfy the required condition ( failure rate < 35.0 (10-6/hour)).
You should modify your design object.
```

Fig. 2.7 Dialogue example



(a) Design object model and integrated task's result in product modeling



(b) Calculation when the design object is modified

Fig. 2.8 Example of real-time operation of failure rate calculation in product modeling

## 2. 6 結 言

本章では、設計対象のモデリング作業と統合化作業とを並行処理することにより、モデリング作業中の設計者に設計対象の不具合の存在や設計変更による影響を知らせる方法を示した。得られた結論を以下に示す。

- (1) 設計者のモデリング作業を設計対象の特性の観点から知的に支援するためにシステムが満足すべき要件を明かにした。
- (2) この要件を満足させるため、統合化作業を部分問題に分解し、モデリング作業と並行して処理可能となった部分問題を順次処理する方法を示した。
- (3) 統合化作業とその部分問題には、抽象化の関係、すなわち、Part-of 関係、Instance-of 関係、Is-a 関係が存在することを示した。
- (4) モデリング作業中の設計者にフィードバックすべき情報として、設計対象の不具合に関する情報と設計変更による影響に関する情報があることを示した。
- (5) 設計対象の故障率を調べる問題に本方法を適用し、その有効性を示した。

## 第3章 オブジェクト指向に基づくCADシステムの開発

### 3.1 緒言

CADシステムは設計対象のモデリングを支援するシステムとして位置づけられているが、現在のシステムの多くは設計対象の形状情報しかモデル化できない。これがシステムの支援範囲を限定する原因となっている。

最近、非形状情報を含む多様な設計対象情報を記述するプロダクト・モデルの概念<sup>(10)</sup>が提案され、それに関する研究が活発化している。鈴木<sup>(69)</sup>は、このモデルを述語論理に基づいて記述する方法を提案している。しかし、この方法は、設計対象情報の構造化が困難であるため、複雑な設計対象のモデル化には不向きである。

一方、複雑な設計対象情報を自然な形でコンピュータ内で扱うため、フレームやオブジェクト指向による設計対象の表現方法<sup>(11)-(18),(70)-(72)</sup>が提案されている。また、これに基づく商用CADシステム<sup>(72)</sup>も開発されている。しかし、これまでの方法では、変速歯車装置のような部品間の接続関係などが変化する機械のモデル化が不可能であった。

本章はこのような変化する機械のモデリングが可能なCADシステムの開発について述べる。まず、3.2節では、変化する機械を形式的に記述し、その記述に基づいたオブジェクト指向による設計対象の表現方法を提案する。次に、

3. 3節でこの設計対象のモデリングが可能なCADシステムの構成、および、このシステムにおけるモデリング方法、挙動のシミュレーション方法を示す。最後に、3. 4節で、このシステムを用いたモデル構築例、および、挙動のシミュレーション例を示し、システムの有効性を明かにする。

### 3. 2 オブジェクト指向による設計対象の表現

#### 3. 2. 1 機械の構造の記述

機械は、一般に複数の実体から構成され、これら実体間でなんらかの関連を持つことにより機能を発揮する。ただし、ここでは実体を機械部品および、それらから構成されるもの（機械、ユニットなど）に限定する。このような機械はFig. 3.1のような実体関連ダイアグラム<sup>(73)</sup>で表すことができる。この機械をコンピュータを用いて設計するためには、この実体、関連を形式的に記述し、その記述に基づいたコンピュータ内での機械の表現方法を確立する必要がある。

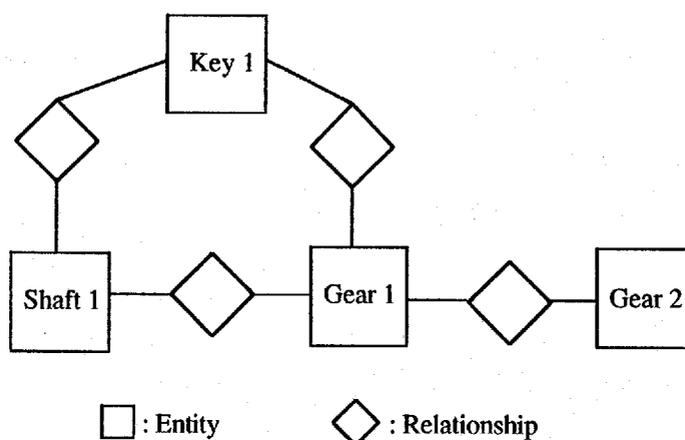


Fig. 3.1 An example of entity-relationship diagram

吉川<sup>(23)</sup>は一般設計学において、実体が属性の集合として記述できることを示している。また、長澤の属性モデリング<sup>(74)</sup>やオブジェクト指向はこれと同様な概念に基づくと考えられる。ただし、属性とは実体を特徴づける性質であり、例えば、歯車は、ピッチ円直径、歯数などの属性をもつ。一般にこの属性間には様々な制約関係が存在する。設計計算や各種シミュレーションなどはこのような制約関係を用いて実行される。このような属性間の制約関係も実体を特徴づける情報と考えられるため、ここでは、実体 $e_i$ を属性の集合 $Ae_i$ と属性間の制約関係の集合 $Re_i$ の組として記述する。すなわち、

$$e_i=(Ae_i,Re_i) \dots \dots \dots (3.1)$$

また、属性 $a \in Ae_i$ および属性間の制約関係 $re \in Re_i$ は以下のように記述できる。

$$a=(ia,va) \dots \dots \dots (3.2)$$

$$re=Rc(ia_1,ia_2,\dots,ia_k) \dots \dots \dots (3.3)$$

ここで、 $ia$ は属性項目、 $va$ はそれに対する値であり、この値には数値、記号だけでなく数値の集合なども取り得るものとする。また、 $Rc$ は関係名であり、 $ia_1,ia_2,\dots,ia_k$ はそれぞれ $Rc$ により関係づけられる属性項目である。

このように記述される実体は、オブジェクト指向におけるオブジェクトとして捉えることができる。ただし、オブジェクト指向では、属性間の制約関係を、設計計算などの作業に依存したメソッドとして記述するが、ここでは様々な作業に利用可能な制約関係として記述する。

属性の中には、摩擦係数のように複数の実体と関わるものがある。このような属性は、一つの実体の属性ではなく、実体間の関連の属性と考えられる。また、属性間の制約関係も複数の実体と関わるものがある。これらを扱うため、実体 $e_1,e_2,\dots,e_n$ 間の関連 $r_k$ についても以下のように表す。

$$r_k=(Ar_k,Rr_k) \dots \dots \dots (3.4)$$

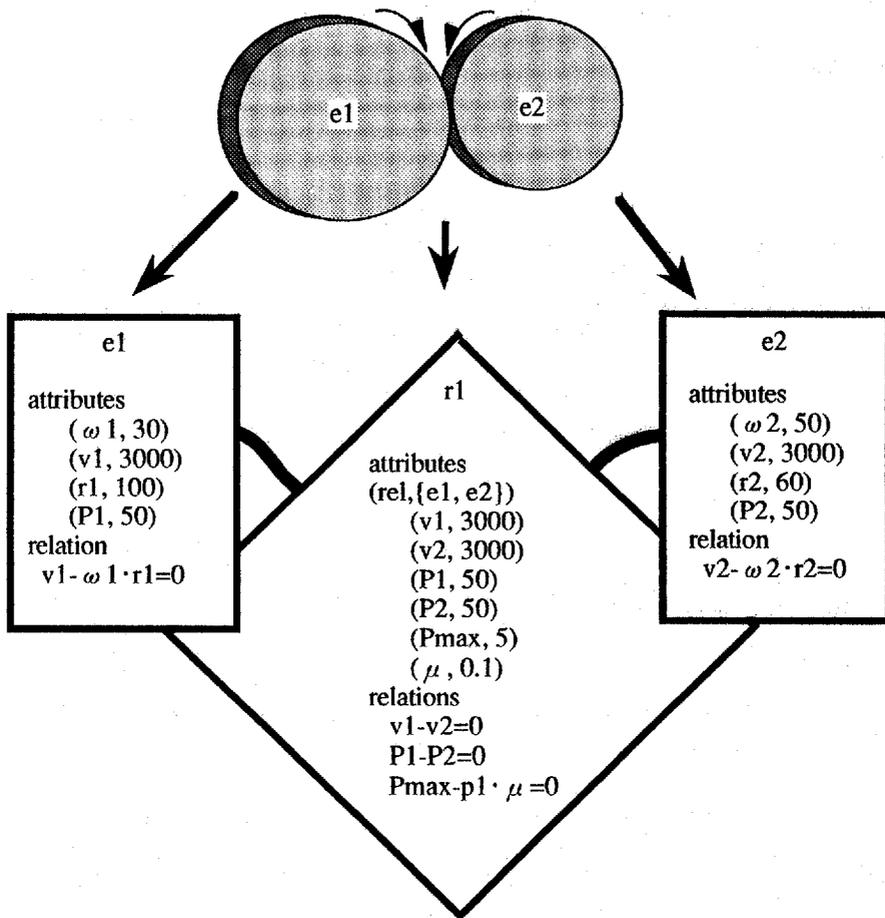


Fig. 3.2 An example of objects' description

ただし,  $Rr_k$ は実体 $e_1, e_2, \dots, e_n$ 間で成立する属性間の制約関係の集合である. また,  $Ar_k$ には以下のものが含まれる.

- (1) 実体 $e_1, e_2, \dots, e_n$ 間の関連であることを表す情報
- (2) 属性間の制約関係集合 $Rr_k$ で関係づけられる属性

ここでは, このような関連もオブジェクト指向におけるオブジェクトとして扱う. Fig. 3.2 に実体と関連の記述例を示す. この図では, 回転運動を伝達する二つの実体 $e_1, e_2$ とそれらの間の関連 $r_1$ が示されており, 実体 $e_1, e_2$ にはそれぞれ

れ属性とその実体内で成立する属性間の制約関係が、関連r1には実体e1,e2の両方と関わる属性および属性間の制約関係が記述されている。なお、以下では実体、関連を総称して構造オブジェクトと呼ぶ。

### 3. 2. 2 機械の状態変化の記述

機械の多くは動的に状態を変化させることにより機能を発現する。この変化には、属性値の変化だけでなく、以下のものもある。

- (1) 構造オブジェクトを記述する属性項目の変化
- (2) 構造オブジェクトを記述する属性項目間の制約関係の変化

これらの変化例をFig. 3.3 に示す。なお、この図で、(a)はある種の化学物質のように、他からの影響により実体の種類が変化する（その実体を記述する属性項目が変化する）場合を、また(b)は実体間の接続関係の変化による属性項目間の制約関係の変化を示している。このように構造オブジェクトを記述する情報が変化する場合、式(3.1)、式(3.4)はある状態における構造オブジェクトを表していると考えられる。

このように変化する構造オブジェクトは、様々な状態を取りうるが、普通、機械の設計段階では設計者は有限個の状態を考える。ただし、ここでの状態は瞬間的な状態ではなく、ある時間区間の状態とし、その時間区間内では(1),(2)の変化は起こらないものとする。

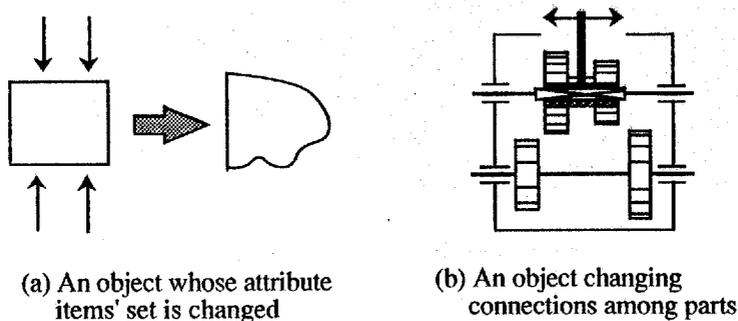


Fig. 3.3 Examples of transformable objects

今、ある構造オブジェクト $O_k$ が $m_{ok}$ 個の状態 $st_u^{ok}$  ( $u=1\sim m_{ok}$ )をとるものとする。状態 $st_u^{ok}$ も、構造オブジェクトと同様に、それを特徴づける属性集合 $A_{oku}$ と属性間の制約関係の集合 $R_{oku}$ によって記述できる。すなわち、

$$st_u^{ok}=(A_{oku}, R_{oku}) \dots \dots \dots (3.5)$$

たとえば、回転しているという状態は、角速度、角加速度などの属性の集合とそれらの中で成立する制約関係の集合で記述できる。状態 $st_u^{ok}$ は、多くの場合、他の構造オブジェクトから影響を受けて他の状態へ遷移する。また、この遷移が他の構造オブジェクトに影響を与えることもある。Fig. 3.4 に他の構造オブジェクトからの影響と状態の遷移との関係を示す。ただし、ここではこの影響 $ef_{c1c2}^{ok}$ を以下のように構造オブジェクト $O_k$ に入力されるエネルギーや情報などの変化として定義する。

$$ef_{c1c2}^{ok}=(c1^{ok}, c2^{ok}) \dots \dots \dots (3.6)$$

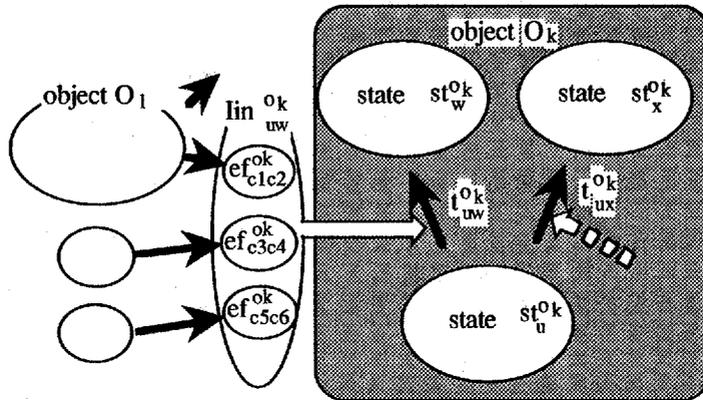


Fig. 3.4 State transition of object

ここで、 $c1^{ok}$ は変化前のエネルギーや情報などの状態であり、 $c2^{ok}$ は変化後の状態である。これらの状態もそのエネルギーや情報などのもつ属性の集合で記述できる。このとき、 $st^{ok}$ から $st^{kw}$ への遷移 $t^{ok}$ は以下のように表せる。

$$t^{ok} = (st_{ok}^{ok}, st_{kw}^{ok}, lin_{ok}^{ok}, Iout_{ok}^{ok}) \dots \dots \dots (3.7)$$

ここで、 $lin_{ok}^{ok}$ は他の構造オブジェクトからの影響の集合を、 $Iout_{ok}^{ok}$ はこの遷移が他の構造オブジェクトに与える影響の集合を表す。

このように構造オブジェクト $Ok$ が $st^{ok}$ から $st^{kw}$ に遷移すると、Fig. 3.3の例のように構造オブジェクト $Ok$ の状態を記述する属性項目や属性項目間の制約関係が変化する場合がある。すなわち、状態 $st^{ok}$ 、 $st^{kw}$ における属性項目集合をそれぞれ $IA_{oku}$ 、 $IA_{okw}$ とすると、

$$IA_{oku} \neq IA_{okw} \dots \dots \dots (3.8)$$

$$R_{oku} \neq R_{okw} \dots \dots \dots (3.9)$$

となる場合がある。

このような $m_{ok}$ 個の状態間の遷移関係は、構造オブジェクトの挙動として認識できる。ここで、構造オブジェクト $Ok$ の挙動 $b^{ok}$ を以下のように記述する。

$$b^{ok} = (St^{ok}, T^{ok}, lin^{ok}, Iout^{ok}) \dots \dots \dots (3.10)$$

ただし、 $St^{ok}$ は構造オブジェクト $Ok$ のとりうる状態 $st^{ok}$ の集合であり、 $T^{ok}$ は構造オブジェクト $Ok$ の遷移 $t^{ok}$ の集合である。また、 $lin^{ok}$ は全ての $lin_{ok}^{ok}$ の和集合であり、 $Iout^{ok}$ は全ての $Iout_{ok}^{ok}$ の和集合である。

### 3. 2. 3 ペトリネットによる挙動の表現

挙動をここでは、ペトリネット<sup>(75)</sup>を用いて表現する。これにより、ペトリネットの実行規則に従った挙動のシミュレーションが可能となる。また、ペトリネットの様々な特性の解析手段を適用できる可能性がある。

一般にペトリネットPNは以下のように記述される.

$$PN = (P, T, I, O) \dots \dots \dots (3.11)$$

ここで, Pはプレースの集合, Tはトランジションの集合, Iはプレースからトランジションへの枝の集合, Oはトランジションからプレースへの枝の集合である. 構造オブジェクトの挙動は次の全単射によってペトリネットに変換できる.

$$\left. \begin{aligned} f1 : St^{ok} \cup Iin^{ok} \cup Iout^{ok} &\rightarrow P \\ f2 : T^{ok} &\rightarrow T \\ f3 : \bigcup_{t_{uw}^{ok} \in T^{ok}} \left( (st_u^{ok}, t_{uw}^{ok}) \cup (Iin_{uw}^{ok} \times \{t_{uw}^{ok}\}) \right) &\rightarrow I \\ f4 : \bigcup_{t_{uw}^{ok} \in T^{ok}} \left( (t_{uw}^{ok}, st_w^{ok}) \cup (\{t_{uw}^{ok}\} \times Iout_{uw}^{ok}) \right) &\rightarrow O \end{aligned} \right\} \dots \dots \dots (3.12)$$

ここで,  $\bigcup_{a \in A} X$  はAに含まれる全てのaに対してXの和集合をとることを表し,  $A \times B$ はAとBの直積を表す. これにより一つの構造オブジェクト $O_k$ の挙動はペトリネットにより表現できる. また, 複数の構造オブジェクトが結合されている場合, 各構造オブジェクトのペトリネットを複合せせる必要がある. ここで複合とは, 各構造オブジェクトの $Iin^{ok}$ ,  $Iout^{ok}$ を表すプレースのうち, 対応するプレース群を一つのプレースにすることである.

構造オブジェクトの挙動をペトリネットで表現した例をFig. 3.5に示す. この図で, 状態を表すプレース (以下, 状態プレース) にあるトークンは, 構造オブジェクトがその状態にあることを表し, 影響を表すプレース (以下, 影響プレース) にあるトークンはその影響が与えられたことを表す.

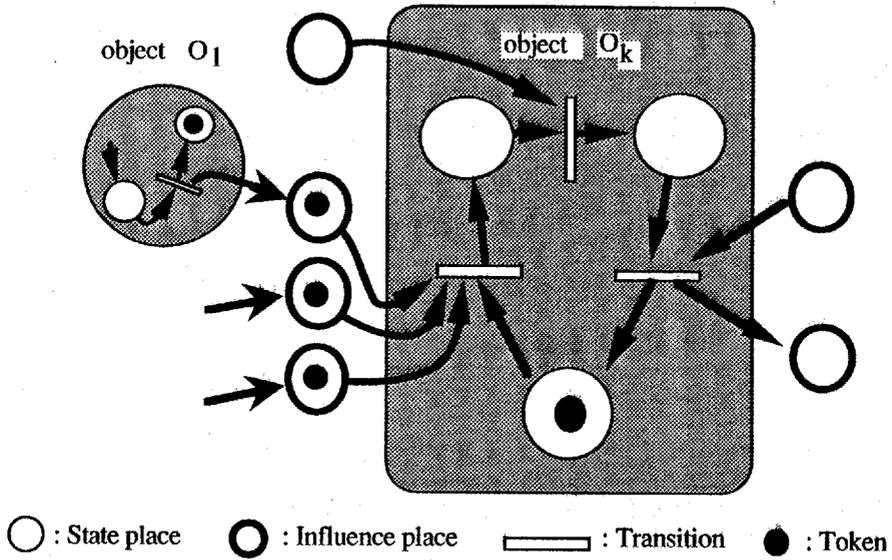


Fig. 3.5 Petri net representation of objects' behavior

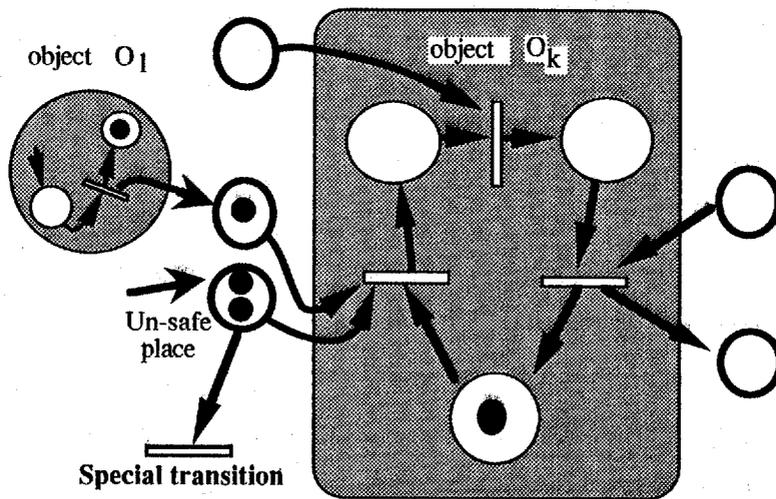


Fig. 3.6 Addition of special transition

このペトリネット表現の特徴は、Fig. 3.5 で色をつけた部分が状態機械<sup>(75)</sup>になることである。すなわち、各トランジションの入力となる状態プレースおよび出力となる状態プレースはそれぞれ一つである。これは式(3.7)の遷移の定義

から明かである。このため、この色をつけた部分の状態プレース集合は影響プレース集合のマーキングに関わらず、P-インバリエントとなることが保証される。ただし、P-インバリエントとは、プレース集合に存在するトークンの合計数が絶えず一定であることを意味する。P-インバリエントであるため、色をつけた部分の初期トークンの数が一つであれば、トランジションが発火しても、その部分のある一つの状態プレースのみトークンをもち、そのトークンの数は一つである。すなわち、色をつけた部分は安全である。これは、構造オブジェクトがいかなる場合も一つの状態しかとりえないことを保証するものである。また、影響プレースも安全である必要がある。しかし、このペトリネット表現では構造的にこのことを保証できない。このため、各影響プレースの安全性については、ペトリネットを複合する時にチェックする必要がある。もし、安全でない影響プレースがあれば、Fig. 3.6のようにそのプレースを入力とする特殊なトランジションをペトリネットに追加することになる。このトランジションは影響プレースに二つ以上のトークンが入るとそれを取り除き、トークンを一つにする。

### 3. 2. 4 挙動のオブジェクト表現

オブジェクト指向に基づくと、実際に機械を構成する構造オブジェクトは、それらの種類を表すクラスの下におかれたインスタンスとして表される。構造オブジェクトの挙動に関する情報は、一般のオブジェクト指向では、その構造オブジェクトのクラスに内包されたメソッドとして記述される。しかし、ここでは以下の理由から、挙動に関する情報を構造オブジェクトから分離し、挙動もオブジェクトとして扱う。

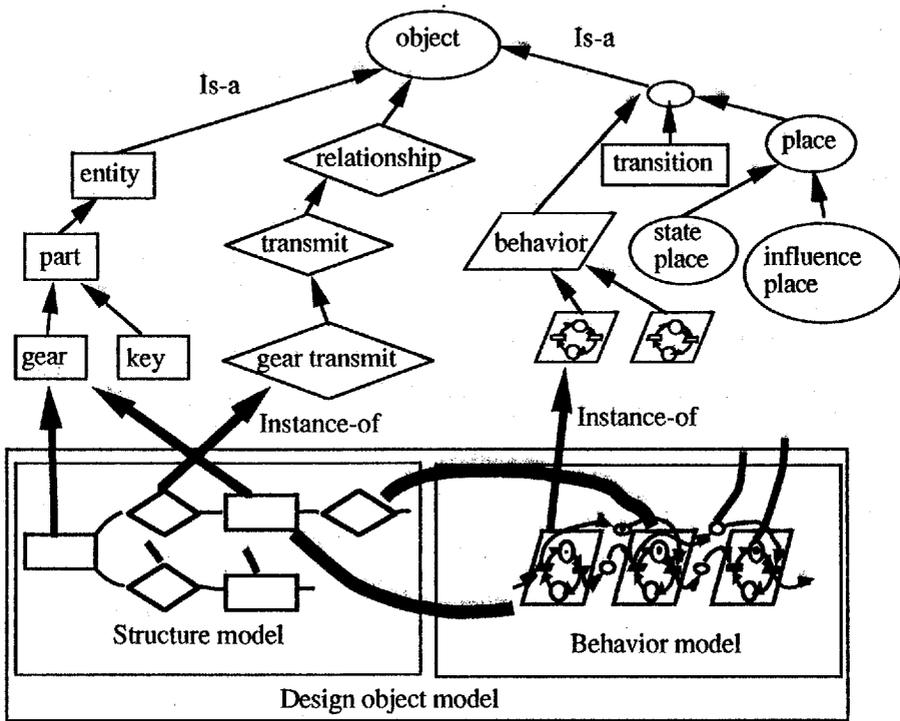
(1) 設計の初期段階では、まず機能を実現するための挙動を考え、次にその挙動を発現する構造オブジェクトを考える場合がある。この場合、挙動のモデリングをした後、挙動のモデルから構造オブジェクトのモデルを作り出すことになり、挙動と構造オブジェクトを独立したモデルとして記述する必要がある。

(2) 構造オブジェクトの種類（クラス）が挙動により変化する場合がある。  
クラスに挙動情報を内包する方法ではこのようなクラスの変化を記述できない。  
このため、ここでは挙動、および、それを構成する状態プレース、トランジション、影響プレースを全てオブジェクトとして扱う。

### 3. 2. 5 設計対象モデル

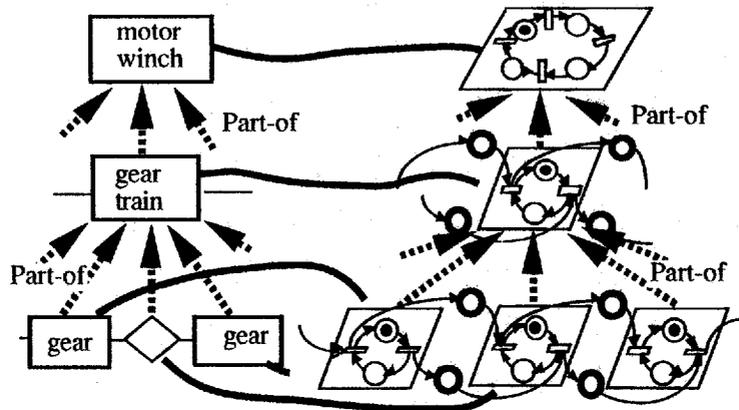
機械は、実体、関連、挙動などを全てコンピュータ内でオブジェクトとして表すことによりモデル化できる。各オブジェクトは、式(3.1)~(3.7), (3.10)で表される情報を含む。ただし、状態プレースは、ある構造オブジェクトのとりうる状態を表し、式(3.5)から明かなように、構造オブジェクトと同様な情報をもつが、ここではこの状態プレースの情報には、対応する構造オブジェクトの情報の中で、その状態の記述に特に必要のないものを含まないものとする。例えば、歯車に対応する状態プレースには、角速度などの属性を記述するが、歯数やモジュールなどの属性を記述しない。これにより、状態プレースおよびそれから構成される挙動を、一般性のあるものにできる。

Fig. 3.7 にモデルの構成例を示す。この図のように、設計対象モデルは、構造モデルと挙動モデルから構成される。構造モデルは、実体クラスと関連クラスの下に生成されるインスタンスのネットワークであり、機械の構造を表す。挙動モデルは、挙動、状態プレース、トランジション、影響プレースのインスタンスのネットワークであり、機械の挙動を表す。この図のように、構造モデルを構成する各構造オブジェクトと挙動モデルを構成する挙動とは一対一で対応している。構造モデルは、挙動モデルのマーキングに対応した設計対象のある状態における機械の構造を表す。このため、マーキングされた状態プレースと構造オブジェクトは、記述されている情報において無矛盾である。



— : Correspondence between structure object and behavior

Fig. 3.7 Construction of design object model



(a) Structure model

(b) Behavior model

— : Correspondence between structure object and behavior

Fig. 3.8 Part-of hierarchy of design object model

設計対象モデルには、オブジェクト指向に従って、Instance-of, Is-a, Part-ofの関係が導入されている。Instance-of関係とIs-a関係はFig. 3.7に、Part-of関係はFig. 3.8に示されている。これらの図に示すように、これらの関係により階層構造が構成される。Fig.3.8のように、挙動は構造オブジェクトと一対一に対応しているため、挙動モデルは構造モデルと同じPart-of階層を構成している。なお、挙動モデルでは、Part-of階層の上位の挙動とそれより下位の挙動群との間で、整合性が保証される必要がある。すなわち、下位の挙動群によって上位の挙動が引き起こされることが保証されなければならない。しかし、後述するシステムでは、この整合性をチェックする機能を用意していないため、設計者が後述の挙動シミュレーションによりそれをチェックするものとする。

### 3. 3 CADシステム

#### 3. 3. 1 システム構成

上述の設計対象モデルの構築が可能なCADシステムを開発した。このシステムの構成をFig. 3.9に示す。クラス・ライブラリは実体、関連、挙動などのクラスの集合である。これには、設計対象のモデリングを容易化するため、様々なクラスが予め入れられている。クラスには、インスタンスの生成や削除などモデルに対する基本的な操作がメソッドとして記述されている。このメソッドを用いることにより、構造モデルと挙動モデルを構築することができる。

モデラーは設計者に、モデリング作業をサポートするための様々な機能を提供する。この機能には、例えば、設計対象を構成する部品を選定するといったものがある。この機能を用いてモデルを変更する場合は、モデラーからクラスにメッセージを送信することにより、クラスに記述されたメソッドを実行することになる。

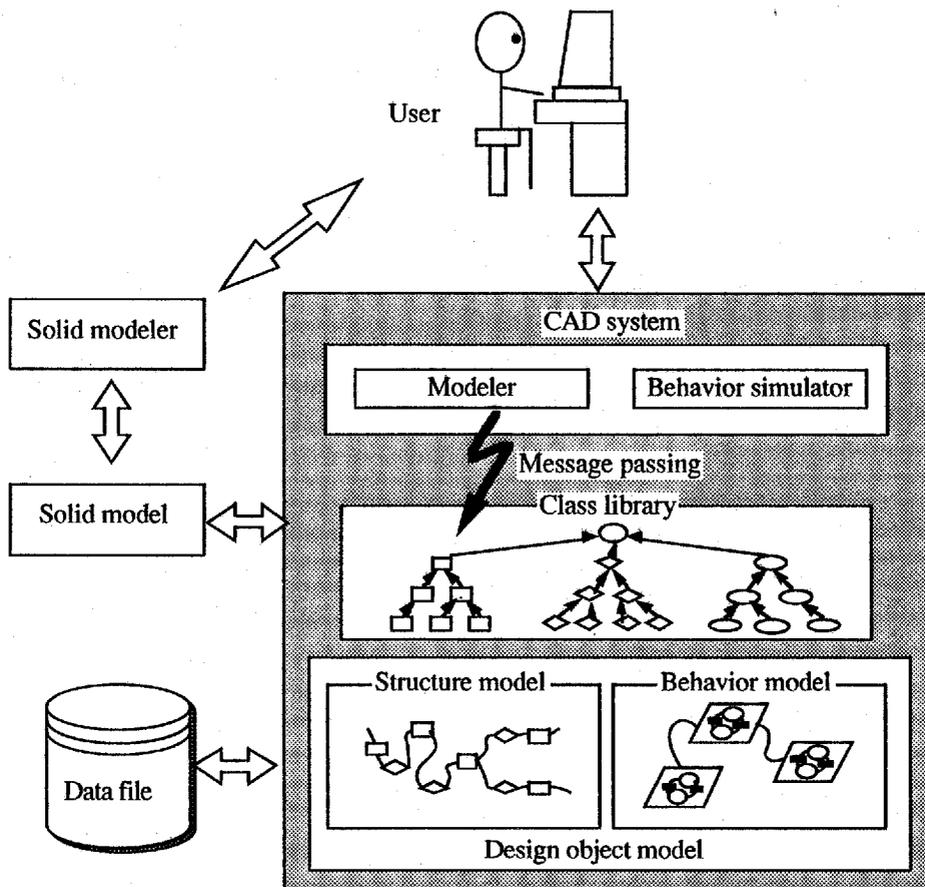


Fig. 3.9 System construction

挙動シミュレータは、構造モデルと挙動モデルを用いて、対象の挙動を調べるものである。また、設計対象の幾何情報の扱いを可能にするため、市販のソリッドモデラーと統合されている。ソリッドモデラーにより構築された幾何モデルのデータは構造モデルの実体オブジェクトおよびその属性と対応づけられる。なお、現在、この対応づけの自動化はなされていない。

### 3. 3. 2 モデリング方法

設計者は、予め用意されたクラスの下にインスタンスを生成することにより、

構造モデルと挙動モデルを構築することができる。モデリング時に実行可能なモデルに対する操作の中で主なものを以下に示す。

#### (1) インスタンスの生成

クラスにメッセージを送信し、クラスに記述されたメソッドを起動することにより、そのクラスのインスタンスを作成できる。実体間に関連インスタンスを生成する場合は、関係づけられる複数の実体と関わる属性間の制約関係がその属性の種類などを基に半自動的に求められる。ここで、半自動的とは、システムが制約関係をもつ可能性のある属性の組を全て見つけ出し、それらの中から設計者が実際に制約関係をもつものを選択するものである。推定された関係は、生成される関連インスタンスに記述される。また、挙動インスタンスを生成する場合も、他の挙動インスタンスとの複合が影響プレースの情報と構造モデルの情報を基に行われる。

#### (2) インスタンスの削除

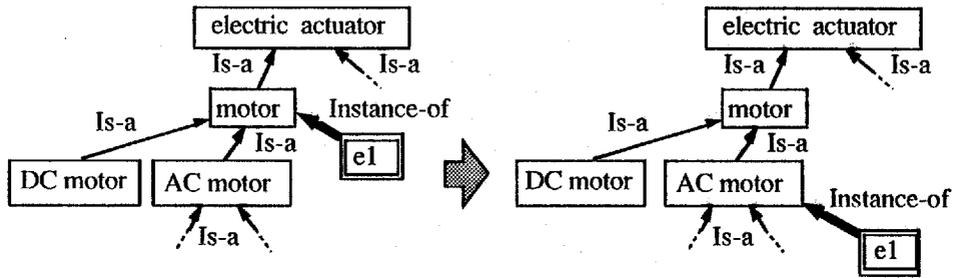
インスタンスの削除も、クラスにメッセージを送信し、クラスに記述されたメソッドを起動することにより実行される。

#### (3) 具象化

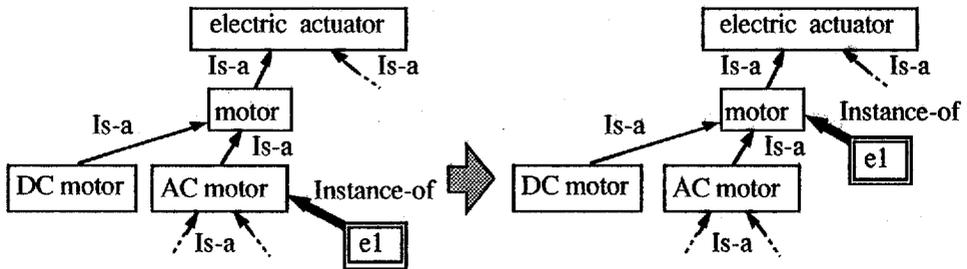
モデルを構成するインスタンスをより具象化したい場合、モデラーはクラス・ライブラリのIs-a階層を利用して具象化の候補を検索し、その中から設計者に選定させる。例えば、Fig. 3.10 (a)のように、モータのインスタンスe1を具象化する場合、候補としてDCモータ、ACモータなどが検索され、その中から設計者が選定することになる。設計者が選定すると、e1が一旦削除され、選択されたクラスの下にe1が再び生成される。このようなモデルの変更は、モデラーからクラスにインスタンスの削除、生成のメッセージを送信することにより実現される。

#### (4) 抽象化

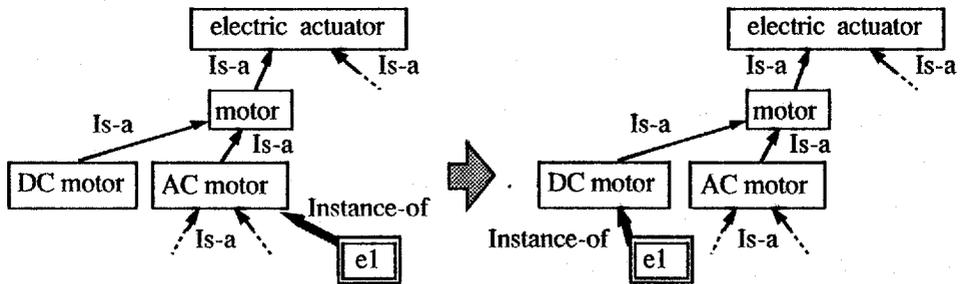
これは、Fig. 3.10 (b)のように、具象化とは逆の操作を行うものであり、あるインスタンスをIs-a階層の上位のクラスのインスタンスにするものである。



(a) Specialization of an entity



(b) Abstraction of an entity



(c) Exchange of an entity

Fig. 3.10 An example of specialization, abstraction and exchange of an entity

(5) 置換

これは、Fig. 3.10 (c)のようにインスタンスをIs-a階層において同じ層の異なるクラスのインスタンスにするものである。

#### (6) 構造・挙動変換

これは、構造オブジェクトのとりうる挙動の決定や、挙動を実現する構造オブジェクトの決定を行うものである。この機能を実現するため、モデラーには構造オブジェクトのクラスと挙動のクラスとの対応関係が予めデータベースに蓄えられている。構造オブジェクトのインスタンスに発現させたい挙動を決定する場合、このデータベースを用いて挙動の候補が検索され、設計者がこの中から選定する。設計者が選定すると、その挙動のインスタンスが生成され、構造オブジェクトのインスタンスとの対応関係が挙動インスタンスに記述される。候補の中に設計者が求めるものがない場合は、設計者自身が状態プレース、影響プレース、トランジションのインスタンスを生成し、それらを組み合わせて挙動インスタンスを作成する。また、逆に、挙動インスタンスに対応する構造オブジェクトを求める場合も同様であり、適切な候補がない場合は、設計者がその挙動をさせたい構造オブジェクトのクラス名を入力し、構造オブジェクトのインスタンスを生成することになる。

#### (7) 属性値の決定

本システムでは、設計者が属性値の決定や変更を自由にできる。しかし、属性は属性間の制約関係により、そのとりうる値が拘束されるため、将来的には、属性値の決定時の制約条件のチェックや、制約関係を基にした属性値の自動決定を行う機能が必要である。なお、トークンをもつ状態プレースとそれに対応する構造オブジェクト間で、記述される情報における整合性がモデラーによって管理されており、一方の属性値の決定や変更がなされると、もう一方の属性値の決定や変更も自動的になされる。

#### (8) モデルの保存と呼び出し

本システムを用いて構築されたモデルをファイルに保存することができる。また、保存されたモデルを呼び出し、過去の設計例として有効利用できる。この呼び出し時は、モデルを構成する全てのオブジェクトが、(1)のインスタンスの生成を実行することにより、設計対象モデル内に作成される。

### 3.3.3 モデリングの視点

設計者が製品設計をする場合、たえず設計対象全体を見ているのではなく、空間的に限定された設計対象の構成部分に注目している場合がある。また、時間的にも、ある時点の対象の状態に注目している場合がある。そこで、設計者が注目している部分を表す視点という概念を導入する。設計者は、構造モデルまたは挙動モデルの注目している部分に視点を置き、その部分を具体化する情報を作成する。例えば、Fig. 3.11のように視点を置くと、設計者は歯車列のインスタンスの属性値の決定、およびその下位構造の決定を行うことになる。このような視点をモデルのPart-of階層の上位から下位へ徐々に下げることによって、設計対象の詳細な情報が作成されていく。また、挙動モデルのマーキングは、注目している状態を表す。異なる状態における属性値などを決定したい場合は、トランジションを発火させ、トークンを移動させることになる。

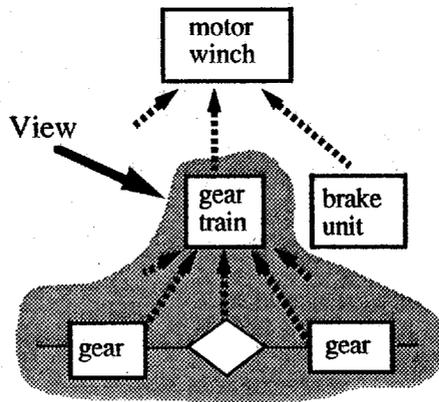


Fig. 3.11 View in design object model

### 3. 3. 4 挙動シミュレーション

構造モデルと挙動モデルを用いて挙動シミュレーションを実行することにより、設計対象の変化を確認することができる。挙動モデルを用いたシミュレーションは、ペトリネットの発火規則に基づいて実行される。このシミュレーションによってトランジションが発火し、トークンが移動すると、トークンをもつ状態プレースの情報と構造モデルの情報との整合性を保つために、構造モデルを構成する構造オブジェクトの情報が自動的に変更される。また、この時、構造オブジェクトの種類が変化する場合は、構造オブジェクトのinstance\_of関係を変更させることになる。

挙動シミュレーションでは、まず、視点のおかれている挙動インスタンスの発火可能なトランジションを発火させ、次に、その挙動インスタンスよりpart\_of関係において下位にある挙動インスタンス群に対して、上位の発火したトランジションに対応するトランジションを発火させる。このような操作を繰り返し、発火可能なトランジションが存在しなくなった時、シミュレーションが終了する。このような挙動モデルのシミュレーションにより、構造モデルもトランジションの発火と連動して変更される。また、このように上下階層を同時にシミュレーションすることにより、上位の構造オブジェクトの挙動が下位の構造オブジェクトのどのような挙動によって引き起こされるかを確認できる。なお、このシミュレーションは、モデリング途中の段階でも、詳細な情報が未決定の設計対象モデルを用いて実行できる。

### 3. 4 システムの実行例

本システムを用いて変速が可能な運動伝達装置のモデリングを行った。Fig. 3.12に構築途中の設計対象モデルを示す。これは、まず”変速運動伝達”という挙動クラスのインスタンスb0を生成し、次にそのインスタンスの具象化によ

ってその挙動を実現する下位の挙動のインスタンスb1~b5を生成することにより構築された。このように挙動のインスタンスが生成されると自動的に対応する実体インスタンスe0~e3, 関連インスタンスr1, r2も生成される。ただし, この段階では各挙動を実現する実体や関連が未決定であるため, 実体インスタンスe0~e3は“実体”というクラスに属し, 関連インスタンスr1, r2は“関連”というクラスに属している。なお, 挙動インスタンスb0, b1, b3, b5は実体インスタンスe0, e1, e2, e3に, b2, b4は関連インスタンスr1, r2にそれぞれ対応している。このようにして構築された挙動モデルを基に, 構造・挙動変換により構造モデルを構成する実体, 関連の種類を決定した。この時のユーザとの対話例をFig. 3. 13に示す。この例では, 挙動インスタンスb3に対応する実体として歯車を決定したことが示されている。このような構造・挙動変換により, 実体インスタンスe2のクラスが“実体”から“歯車”に変更される。構造・挙動変換を各挙動インスタンスに対して行った結果, 構造モデルはFig. 3. 14のようになった。この図でnilは接続関係が存在しないことを表す。このモデルは, 歯車e1, e3間で運動の伝達が行われており, 歯車e1, e2間では運動が伝達されていないことを表している。このモデルにさらに軸, 軸受などのインスタンスを追加し, それらに対応する挙動のインスタンスを決定した。その結果, 設計対象モデルはFig. 3. 15のようになった。ただし, この図では, モデルの一部だけが示されている。この図の(a), (b)は, 同図(c)のマーキングと対応した状態を表している。このような複雑なモデルでも, 予め用意されたクラスの下にインスタンスを生成することにより, 容易に構築できる。このモデルを用いて挙動シミュレーションを行い, 異なる状態を生成した結果をFig. 3. 16に示す。この図から, e1, r1, e2に対応する挙動オブジェクトb1, b2, b3のマーキングが変化し, それに伴って構造モデルのe1, e2間の関連r1がgear-transmitに変更されたことが分かる。この関連の変更に伴い, e1, e2間の制約関係が推定され, この関連r1に新たに記述された。このような挙動シミュレーションにより設計者は挙動による対象の変化を把握できる。

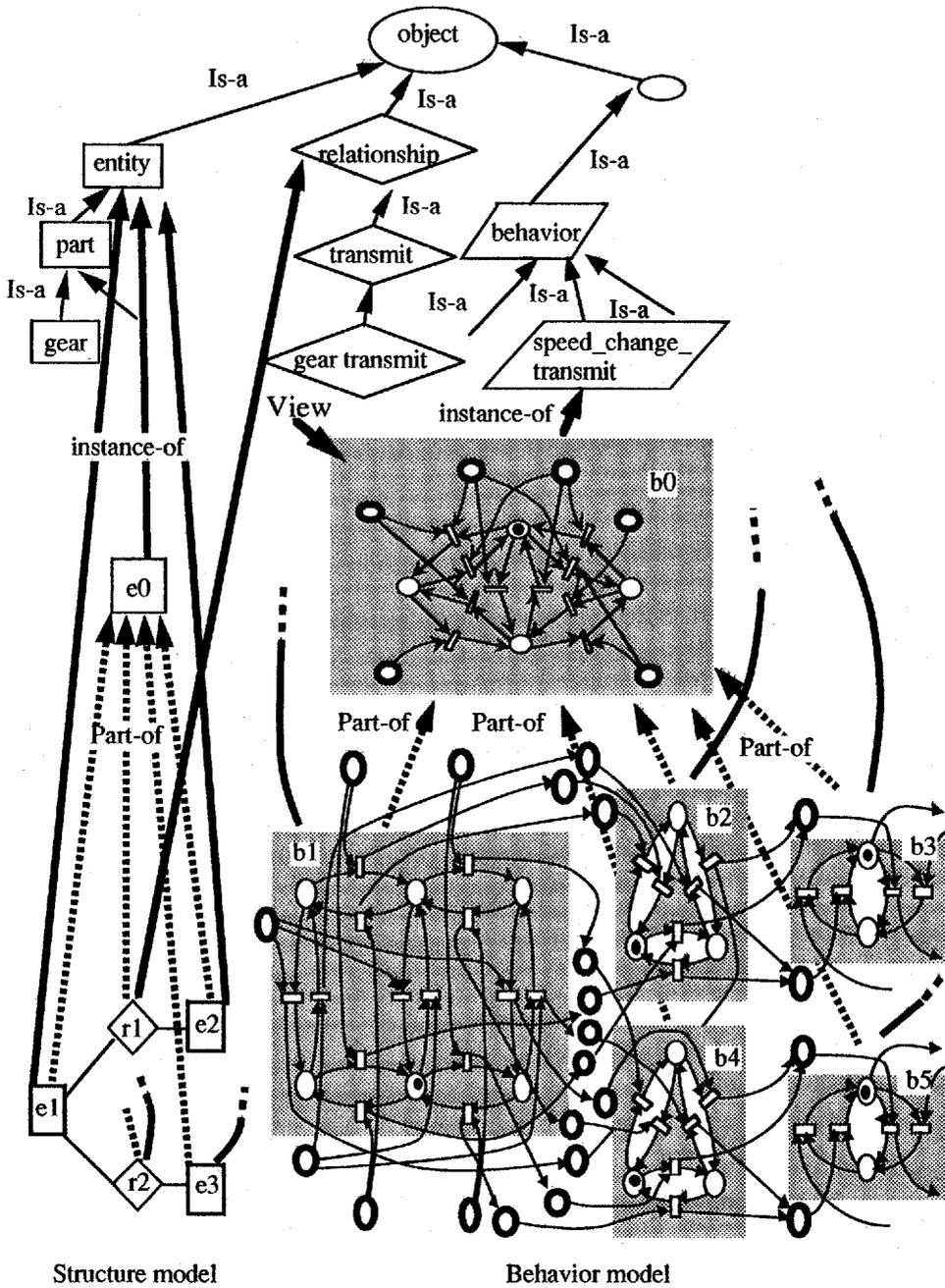


Fig. 3. 12 A design object model in product modeling

30 : ?- ms(op, b3, make\_object).  
 1 : friction wheel  
 2 : gear  
 Which do you want to use for realizing behavior b3 ?  
 2.  
 The entity e2 corresponding to b3 is gear.

Fig. 3.13 Dialogue example

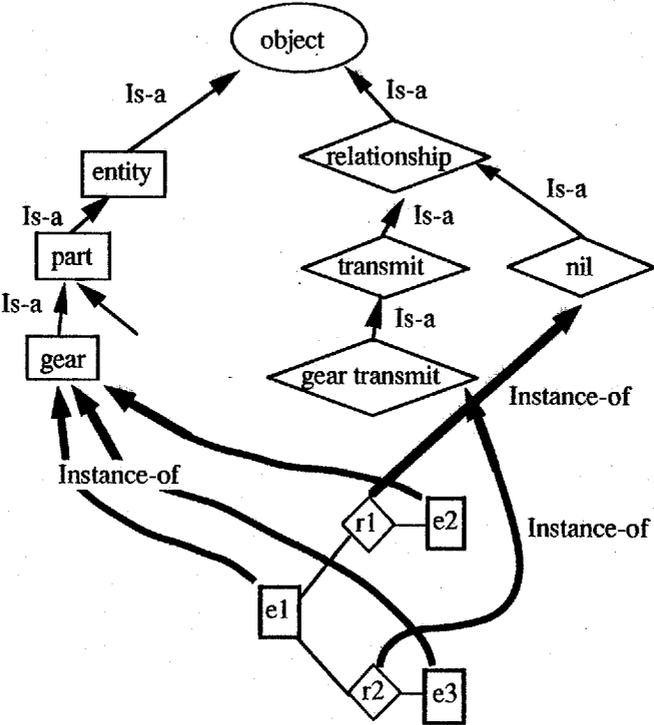
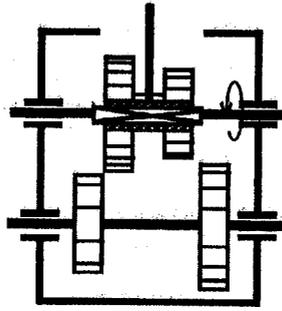
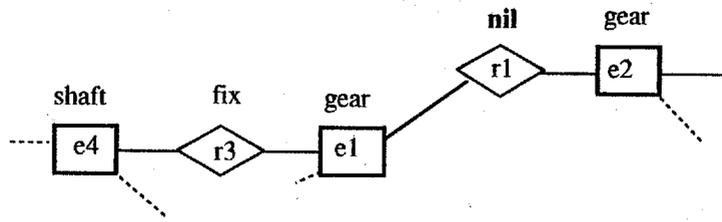


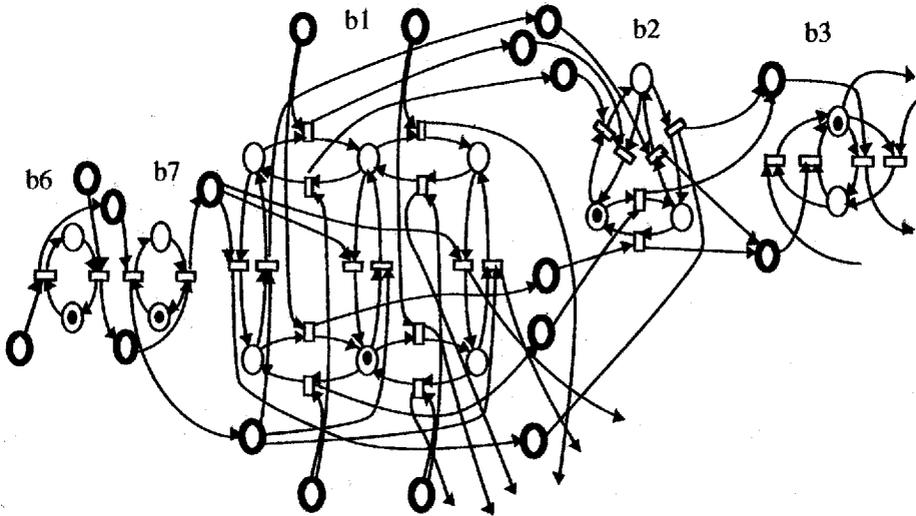
Fig. 3.14 A structure model in product modeling



(a) Design object

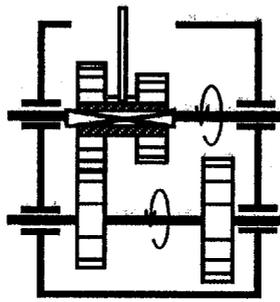


(b) Structure model

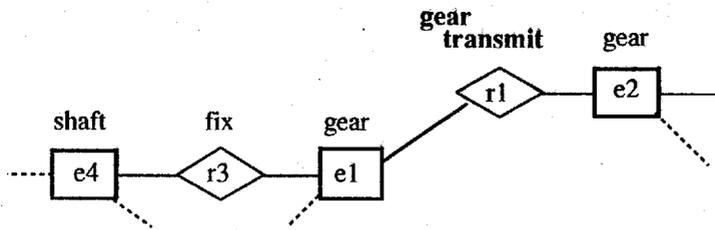


(c) Behavior model

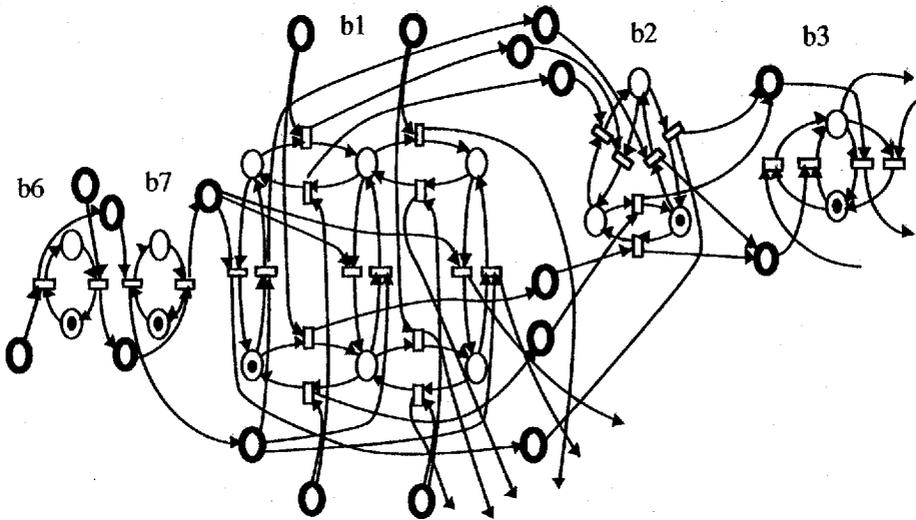
Fig. 3.15 An example of design object model



(a) Design object



(b) Structure model



(c) Behavior model

Fig.3.16 An example of design object model after state transition

### 3.5 結 言

本章では、オブジェクト指向に基づく設計対象の新しい表現方法を提案し、その方法に基づいて開発したCADシステムについて述べた。本章で得られた結論を以下に示す。

- (1) 機械の構造を実体と関連からなるグラフとして捉え、実体、関連をそれぞれオブジェクトとして表すことにより、複数の実体と関わる属性や属性間の制約関係を記述することができる。
- (2) 機械の挙動をペトリネットで表すことにより、ペトリネットの実行規則に基づく挙動シミュレーションが可能になる。
- (3) 挙動を実体や関連と分離したオブジェクトとして扱うため、挙動を先にモデル化し、そのモデルを基に、その挙動を発現する機械構造をモデル化することができる。また、実体や関連の種類の変化を引き起こす挙動も表現することが可能となる。
- (4) 開発したCADシステムを用いることにより、構造などが変化する設計対象のモデリングが容易にできる。また、モデルを用いた挙動シミュレーションにより、設計対象の挙動を確認できる。

## 第4章 製品モデリングと故障シミュレーションの並行処理

### 4.1 緒言

製品設計段階で重要な統合化作業として信頼性解析がある。これは、製品の高信頼度化を実現するため、設計対象で起こりうる故障や信頼度を予測するものである。この作業のコンピュータ支援が、特に最近、製品の複雑化、大規模化に伴い、要求されるようになった。このコンピュータ支援に関する研究として、コンピュータ・システムを対象とした信頼性解析ツールの研究<sup>(76)</sup>、FTA (Fault Tree Analysis)やFMEA (Failure Mode and Effects Analysis)の自動化に関する研究<sup>(77)-(90)</sup>、故障シミュレーションの研究<sup>(91)-(94)</sup>、設計診断の研究<sup>(95)</sup>がある。しかし、これらの研究ではCADシステムとの統合すらなされておらず、作業効率や設計者のサポートという面で問題がある。本章では、この信頼性解析方法の一つである故障シミュレーションに対して、第2章で提案した概念を適用する。すなわち、第3章で示したCADシステムによるモデリング作業と並行して故障シミュレーションを実行することにより、モデリング作業中の設計者を信頼性の立場から支援するものである。

本章では、まず、4.2節で制約伝播<sup>(92),(93)</sup>に基づく故障シミュレーションの方法と、故障シミュレーションの入出力情報を示す。次に、4.3節ではリアルタイム設計処理に基づいて故障シミュレーションを部分問題に分解し、

CADシステムのモデルに対する操作と連動して部分問題を処理する方法を示す。そして、4.4節では、CADシステムと統合されたリアルタイム故障シミュレータについて述べ、最後に4.5節でこのシステムの実行例を示す。

## 4.2 故障シミュレーション

### 4.2.1 入出力情報

故障シミュレーションは、物理量間の制約関係や機械の構成要素間の接続関係を用いて、構成要素で起こりうる故障が機械全体に及ぼす影響を調べるものである。

これまで、様々な故障シミュレーションの方法が提案されているが、ここでは、以下の特徴をもった方法を提案する。

- (1) 部品間の固着などの故障を扱うため、実体だけでなく関連に対しても起こりうる故障とその影響を調べる。
- (2) 機械は様々な状態をとるが、その状態によって起こりうる故障やそれによる影響が異なる。これを考慮に入れるため、機械のとりうる各状態に対して、起こりうる故障とその影響を調べる。

なお、冗長系については、ここでは、考慮に入れない。

提案する故障シミュレーションは、第3章で示したCADシステムにより構築されるモデルを用いて実行される。ただし、機械のある状態を表す構造モデルを入力とし、挙動シミュレーションにより構造モデルの表す状態が変更されると、その異なる状態を表す実体や関連に対して再び故障シミュレーションを実行する。これにより、機械のとりうる各状態に対する故障による影響を調べることができる。今、構造モデルを表す情報集合DSの中で故障シミュレーションと関わる情報集合をDFとする。設計対象が $N_e$ 個の実体 $e_i$  ( $i=1\sim N_e$ )と $N_r$ 個の関連 $r_j$  ( $j=1\sim N_r$ )から構成されているものとする、情報集合DFは以下のよう

に分解できる。

$$DF = \left( \bigcup_{i=1}^{Ne} e_{Fi} \right) \cup \left( \bigcup_{j=1}^{Nr} r_{Fj} \right) \dots \dots \dots (4.1)$$

ただし、 $e_{Fi}$ 、 $r_{Fj}$ は、それぞれDFの中の、実体  $e_i$ に関する情報集合、および、関連  $r_j$ に関する情報集合である。

故障シミュレーションは、実体や関連で成立する物理量間の制約式をもとに、構造モデルが表す状態において故障が発生した時の異常状況を推定する。この故障シミュレーションFSの実行をここでは次のように表現する。

$$FS = (DF, FF, G) \dots \dots \dots (4.2)$$

ここで、FFは故障シミュレーションの実行手続きを表し、Gはシミュレーション結果を表す。

シミュレーション結果Gは、Fig. 4.1 に示すように、有向グラフ（以下、故障伝播グラフと呼ぶ）で表される。このグラフの始点は、短絡、固着などの故障の種類（故障モード）を表し、その他の節点は異常状態を表す。ただし、ここでは異常状態を物理量の異常と定義する。例えば、ベルト伝達機構においてベルトの緩みによってトルクの伝達が十分にできなくなる場合は、“ベルトの緩み”が故障モードであり、“トルクの低下”が異常状態である。この図のように、故障モードと異常状態は、実体、関連毎に記述される。このグラフの枝はこれらの間の因果関係を表す。なお、この図で実体2の異常状態1に複数の枝が入力されているが、これは関連1の異常状態1と実体2の故障モードが同じ異常状態を引き起こすことを表している。

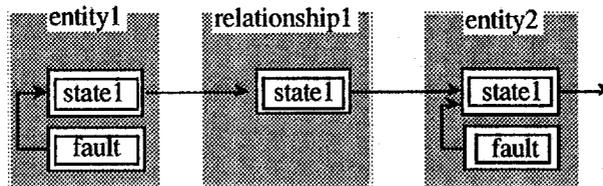


Fig. 4.1 Fault propagation graph

このグラフの故障モードを表す節点には、故障モード名とその発生率が記述される。ただし、故障モードの発生率は“故障率に故障モード比を乗じた値”であり、故障モード比はある実体や関連で起こり得る全ての故障モードの中でその故障モードの起こる比率である。異常状態を表す節点には、物理量の定性値<sup>(92)</sup>が記述される。ただし、定性値として、+（正常範囲以上）、-（正常範囲以下）、0（正常範囲内）が割り当てられる。ここで、実体  $e_i$  の故障モードと異常状態に関する情報集合を  $S_{e_i}$ 、関連  $r_j$  の故障モードと異常状態に関する情報集合を  $S_{r_j}$  とする。また、実体  $e_i$  と実体  $e_k$  の間に関連  $r_j$  がある場合、実体  $e_i$  から関連  $r_j$  への異常状態の伝播（枝）の集合を  $A_{r_j}^{e_i}$ 、関連  $r_j$  から実体  $e_i$  への異常状態の伝播の集合を  $A_{e_i}^{r_j}$  とすると、故障伝播グラフ  $G$  は、以下のような情報の和集合として表せる。

$$G = \left( \bigcup_{i=1}^{N_e} S_{e_i} \right) \cup \left( \bigcup_{j=1}^{N_r} S_{r_j} \right) \cup \left( \bigcup_{j=1}^{N_r} \left( \bigcup_{e_i \in E_{r_j}} \left( A_{r_j}^{e_i} \cup A_{e_i}^{r_j} \right) \right) \right) \dots \dots \dots (4.3)$$

ただし、 $E_{r_j}$  は関連  $r_j$  と接続する実体  $e_i$  の集合である。実体  $e_i$  の  $S_{e_i}$  は、さらに、その実体の故障モードの集合  $S_{e_i}^m$ 、それに起因する異常状態の集合  $S_{e_i}^s$ 、故障モードからその異常状態への枝の集合  $A_{e_i}^{e_i}$ 、その実体と接続する関連  $r_j$  からの異常状態の伝播に起因する異常状態の集合  $S_{e_i}^{r_j}$  に分かれる。すなわち、

$$S_{e_i} = S_{e_i}^m \cup S_{e_i}^s \cup A_{e_i}^{e_i} \cup \left( \bigcup_{r_j \in R_{e_i}} S_{e_i}^{r_j} \right) \dots \dots \dots (4.4)$$

ただし、 $R_{e_i}$  は実体  $e_i$  と接続する関連の集合である。同様に  $S_{r_j}$  も、その関連における故障モードの集合  $S_{r_j}^m$ 、それに起因する異常状態の集合  $S_{r_j}^s$ 、故障モードからその異常状態への枝の集合  $A_{r_j}^{r_j}$ 、接続する実体  $e_i$  からの異常状態の伝播に起因する異常状態の集合  $S_{r_j}^{e_i}$  に分解できる。すなわち、

$$S_{ij} = S_{ij}^m \cup S_{ij}^s \cup A_{ij}^j \cup \left( \bigcup_{e_i \in E_{ij}} S_{ij}^{e_i} \right) \quad \dots \dots \dots (4.5)$$

このように、故障シミュレーションの入力情報 DF および出力情報 G は、ともに、部分情報に分解できる。ここで、モデリング作業中に DF の部分情報が生成された時に、部分問題の処理により対応する G の部分情報を得ることができれば、故障シミュレーションをモデリング作業と並行して処理することが可能になる。

#### 4. 2. 2 故障シミュレーションの方法

提案する故障シミュレーションでは故障による影響が各実体および各関連毎に制約伝播に基づいて推定される。この故障シミュレーションのために、以下の設計対象に関する情報が利用される。

- (1) 実体や関連で起こりうる故障モードに関する情報
- (2) 実体や関連で成立する物理量間の制約式
- (3) 実体と関連から構成されるネットワーク

情報(1)には、実体や関連で起こりうる故障モードと、それに対応する変数の定性値、およびその故障モードが起こるための条件が含まれる。故障シミュレーション時は、この条件を満足する故障モードに対してのみ、その影響が推定される。情報(2)は、 $f(x_1, x_2, \dots, x_n) = 0$  という陰関数で表現される。その引数である各変数  $x_i$  には、正常状態で変化しうるか否か、および、故障状態で変化しうるか否か、という情報が付加されている。また、変数は、他の実体などとの入出力関係から、入力変数、内部変数、出力変数に分類される。また、情報(3)は、実体間の接続関係を表すもので、異常状態を伝播する時に用いられる。これらの情報は構造モデルとクラス・ライブラリに記述される。

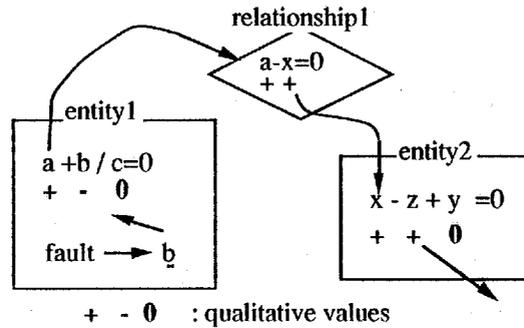


Fig. 4.2 Constraint propagation in fault simulation

これらの情報をもとにした異常状態の推定方法を以下に示す。なお、この方法は基本的には山口ら<sup>(92), (93)</sup>の制約伝播と同様であるが、実体だけでなく関連に対しても異常状態の推定を行っている点が異なる。これにより、関連で起こる故障とその影響を調べることが可能となる。

STEP1: 実体または関連で起こりうる故障モードに対応する変数の定性値、または、他の実体や関連からの入出力異常に対応する入出力変数の定性値を求める。求めた変数を  $x_0$  とする。

STEP2: 変数  $x_0$  を引数にもつ制約式  $f$  を情報(2)より求める。

STEP3: 制約式  $f$  において、 $x_0$  以外の引数の中で変化しうる変数  $x_e$  (複数あればその中の任意の一つ) を情報(2)より求める。

STEP4: 制約式を次のような陽関数に変形する。

$$x_e = f(x_1, x_2, \dots, x_0, \dots, x_n)$$

STEP5: 変形した制約式の右辺の引数の中で、 $x_0$  以外の変数の定性値を0と見なし、 $x_0$  の定性値によって  $x_e$  が定性的にどう変化するか調べる。

STEP6: 制約式  $f$  の変数の中で、 $x_e$  以外に変化しうる変数が存在すれば、それを新たに  $x_e$  としSTEP4に移る。

このような処理を各実体および各関連毎に行い、求めた定性的な変化を実体や関連の異常状態とみなす。求めた変数の定性値を情報(3)を用いて次々と Fig. 4.2 のように伝播させることにより、故障による影響を調べる。

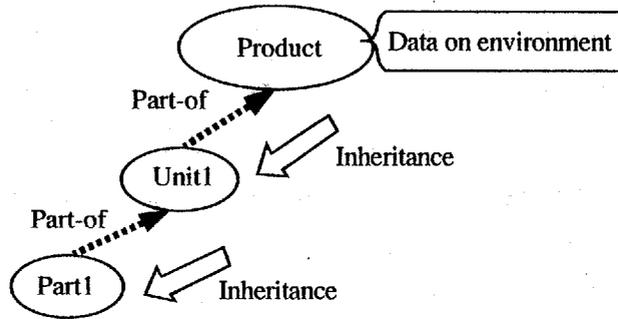


Fig.4.3 Inheritance of data on environment

また、故障シミュレーションでは、故障モードに対しては、発生率が算出される。これは、公表されているMIL-HDBK-217B<sup>(96)</sup>とGreen<sup>(97)</sup>の故障率モデルを基に故障率を計算し、これと故障モード比との積をとることによって、算出される。この時、製品の環境や動作条件を考慮に入れて計算される。この環境や動作条件は構造モデルに属性として記述される。Fig. 4. 3 のようにある部品1の環境や動作条件を求める場合、部品1に記述されたものだけでなく、Part-of階層の上位のユニット1と製品に記述されたものも継承される。なお、求めた故障モードの発生率が予め設定された値（以下、下閾値）より小さい場合は、その故障モードによる影響は調べないものとする。この値は設計者が設計を始める時に設定する。

#### 4. 3 リアルタイム故障シミュレーション

##### 4. 3. 1 故障シミュレーションの分解

モデリング作業と並行して故障シミュレーションを実行するため、リアルタイム設計処理の概念を適用する。すなわち、故障シミュレーションを複数の部分問題に分割し、モデリング作業中にDFの部分情報  $e_{Fi}$ ,  $f_{Fj}$  が生成されると、即座に、この部分問題の処理によって対応する故障伝播グラフの部分情報を得る。このようなモデリング作業と連動した故障シミュレーションをリアルタイム

故障シミュレーションと呼ぶことにする。このような処理を行うため、ここでは、故障シミュレーションを、以下のような4種類の部分問題に分解する。

(1) 部分問題P1

これは、実体  $e_i$  で起こりうる全ての故障モードとその発生率を求め、発生率が設定値以上のものに対しては、それに起因する異常状態集合を求める問題である。問題の処理手続きを  $f_1$  とすると、この問題は次式で表現できる。

$$P1 = (e_{Fi}, f_1, S_{ei}^m \cup S_{ei}^s \cup A_{ei}^{ej}) \dots \dots \dots (4.6)$$

(2) 部分問題P2

これは、関連  $r_j$  で起こりうる全ての故障モードとその発生率を求め、発生率が設定値以上のものに対しては、それに起因する異常状態集合を、手続き  $f_2$  に従って求めるものであり、次式で表せる。

$$P2 = (r_{Fj}, f_2, S_{rj}^m \cup S_{rj}^s \cup A_{rj}^{rj}) \dots \dots \dots (4.7)$$

(3) 部分問題P3

これは、ある実体  $e_i$  の異常状態集合が、それと接続する関連  $r_j$  に及ぼす影響を手続き  $f_3$  に従って調べるものであり、次式で表現する。

$$P3 = (r_{Fj} \cup S_{ei}^s \cup (\bigcup_{\substack{rk \in Rei \\ rk \neq rj}} S_{ei}^{rk}), f_3, A_{rj}^{ej} \cup S_{rj}^{ej}) \dots \dots \dots (4.8)$$

(4) 部分問題P4

これは、ある実体間の関連  $r_j$  の異常状態集合が、それと接続されている実体  $e_i$  に及ぼす影響を、手続き  $f_4$  に従って調べるものであり、次式で表す。

$$P4 = (e_{Fi} \cup S_{rj}^s \cup (\bigcup_{\substack{ek \in Erj \\ ek \neq ei}} S_{rj}^{ek}), f_4, S_{ei}^{rj} \cup A_{ei}^{rj}) \dots \dots \dots (4.9)$$

このような部分問題の処理によって、故障伝播グラフの部分情報が得られる。

これらを統合することにより，設計対象全体に対するシミュレーション結果Gを得ることができる．なお，上述の手続きf1～f4は，各部分問題に対する故障シミュレーションの実行を意味する．

#### 4. 3. 2 故障シミュレーションのリアルタイム処理

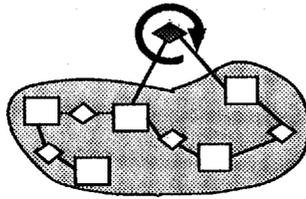
リアルタイム故障シミュレーションでは，上述の部分問題の処理を，モデリング作業中の構造モデルに対する操作と連動して実行する．

##### (a) インスタンスの生成

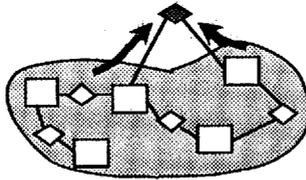
モデリング作業中に実体インスタンスが生成されると，部分問題 P1 の処理により，その実体内で起こりうる故障モードとその発生率，およびその故障モードによってその実体内で引き起こされる異常状態が求められる．また，実体間に関連インスタンスが生成された場合は，Fig.4.4のように3通りのシミュレーションが部分問題の処理によって行われ，その影響が故障の観点から調べられる．以下にこの3通りのシミュレーションと部分問題との関係を示す．

- (1) 追加された関連インスタンスで起こりうる故障モードとその発生率を調べ，その故障モードによってその関連インスタンス自身のとりうる異常状態を部分問題 P2 の処理によって求める (Fig.4.4(a)) ．
- (2) 既存の設計対象の異常状態が，追加された関連インスタンスに及ぼす影響を部分問題 P3 の処理によって調べる (Fig.4.4(b)) ．
- (3) 追加された関連インスタンスで起こりうる異常状態が，既存の設計対象に及ぼす影響を部分問題P3とP4の処理によって調べる (Fig.4.4(c)) ．

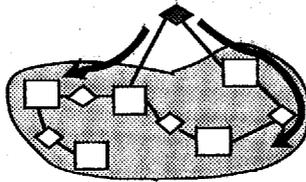
これらの中で，(2),(3)で異常状態の伝播を行うが，この伝播中に推定された異常状態が他からの伝播によって引き起こされる異常状態と同じになれば，伝播を停止する．



(a) Examining what faults and abnormal states arise in the newly added relationship (sub-problem P2)



(b) Propagating the abnormal states arising in the existing design object to the newly added relationship (sub-problem P3)



(c) Propagating the abnormal states arising in the newly added relationship to the existing design object (sub-problem P4, P3,..)

◆ : relationship added newly      ○ : existing design object

Fig. 4.4 Real-time fault simulation when a relationship is added newly to the design object

今、モデリング作業中に構造モデルに対するN回目の操作で新たに実体インスタンスまたは関連インスタンスが追加されたものとする。この時、上述の処理によって新たに生成された故障モードや異常状態に関する情報集合を  $ga_N$  とすると、その時点の構造モデルに対する故障伝播グラフを次のようにして得ることができる。

$$C1 = ((G_{N-1}, ga_N), fa, G_N) \dots \dots \dots (4.10)$$

ただし、 $G_N$ はN回目の操作実行後の故障伝播グラフを示し、 $fa$ は故障伝播グラフの統合手続きを示す。この式は、新たにインスタンスの追加がモデルに対して行われた時、故障シミュレーションを更新されたモデル全体に対してやり直すのではなく、その操作実行前の故障伝播グラフを修正することによって $G_N$ を得ることを意味するものである。

(b) インスタンスの削除

モデリング作業中に実体インスタンスまたは関連インスタンスが構造モデルから削除されると、それに対応して故障伝播グラフを修正する必要がある。この修正は、故障伝播グラフにおいて、削除される部分から他の部分への異常状態の伝播が存在する場合、その伝播を Fig. 4.5 (a) のように削除するものである。ただし、Fig. 4.5 (b) のように伝播の途中で他から伝播された異常状態と同じ異常状態になる場合は、そこまでの伝播を削除する。

ここで、N回目のモデルに対する操作で実体や関連のインスタンスが削除されたとすると、この時の故障伝播グラフの修正も式(4.10)と同様な形式で表せる。すなわち、

$$C2 = ((G_{N-1}, gdN), fd, G_N) \dots \dots \dots (4.11)$$

ただし、この $gdN$ は、 $G_{N-1}$ から削除される情報集合であり、 $fd$ は故障伝播グラフの修正手続きを示す。

(c) インスタンスの具象化・抽象化・置換

モデリング作業中に実体や関連のインスタンスの具象化、抽象化、置換が行われると、第3章で示したように、まずインスタンスの削除が行われ、次にインスタンスの生成が行われる。このため、これらの操作が実行されると、故障シミュレーションでは(b)の処理の後、(a)の処理が実行される。

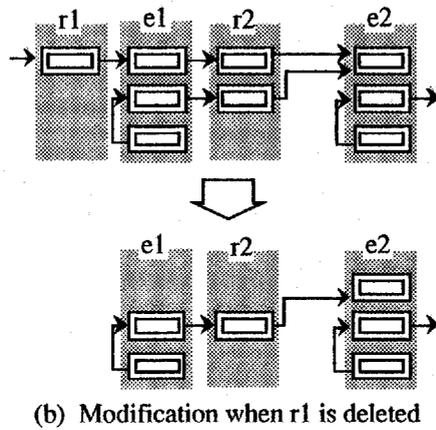
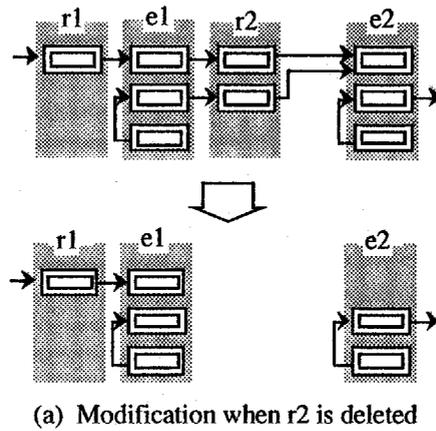


Fig.4.5 Modification of fault propagation graph when an entity or a relationship is deleted

(d) 環境・動作条件の変更

ある実体インスタンスの環境や動作条件が変更されると、Part-of 階層においてその実体インスタンスより下位の実体や関連のインスタンスで起こりうる故障モードや異常状態が変化する可能性がある。このため、それより下位の全ての実体や関連のインスタンスに対して、(b)の処理を行った後、(a)の処理を行う。

(e) 挙動シミュレーション

挙動シミュレーションを実行すると、実体や関連のインスタンスの表す状態

が変化する。この場合、状態が変化した実体や関連のインスタンスに対して、部分問題の処理により故障シミュレーションを行う。

#### (f) 過去の設計例の利用

過去の設計例を利用する場合、CADシステムはファイルに保存された過去の設計例を読み込み、設計対象モデルにこの設計例を構成するインスタンスを生成する。このため、故障シミュレーションでは全ての生成されたインスタンスに対して(a)の処理を行うことになる。

このようなモデリング作業時のモデルに対する操作と連動した故障シミュレーションは、Part-of 階層の各階層の実体および関連のインスタンスに対して可能であれば行われる。このため、モデリングの初期段階で上位の実体や関連のインスタンスが生成されると、それに対して故障シミュレーションが実行され、モデリング作業の進行に伴ってそのインスタンスを構成する下位の実体や関連のインスタンスが明らかになってくると、それらに対しても故障シミュレーションが実行される。つまり、モデルの下位構造が詳細化されると、故障の伝播も詳細化される。ただし、ここでは、上位の実体や関連に記述されている制約式などの情報と、それより下位の実体や関連の情報とが無矛盾であることを仮定している。

### 4. 4 リアルタイム故障シミュレータの開発

第3章で示したCADシステムと統合化されたリアルタイム故障シミュレータを開発した。Fig. 4. 6 に示すように、リアルタイム故障シミュレータは、第2章で示したシステムと同様な構成になっている。この図でSPS1~SPS4は部分問題の種類P1~P4に対応している。また、故障率データベースはSPS1,SPS2からの要求により故障モードの発生率を計算するものである。これは、実体や関連の種類により発生率の計算手続きが異なるため、Fig. 4. 7 のようなオブジェクト指向の概念に基づいた構成となっている。すなわち、発生率を求めるため

のデータと手続きが、この図のソレノイドやモータのように実体や関連の種類毎にカプセル化されている。ポリモアフィズム<sup>(67)</sup>の考え方により異なる種類のものでも、同じメッセージ(r\_f\_rate)を送信することにより、発生率が求まる。

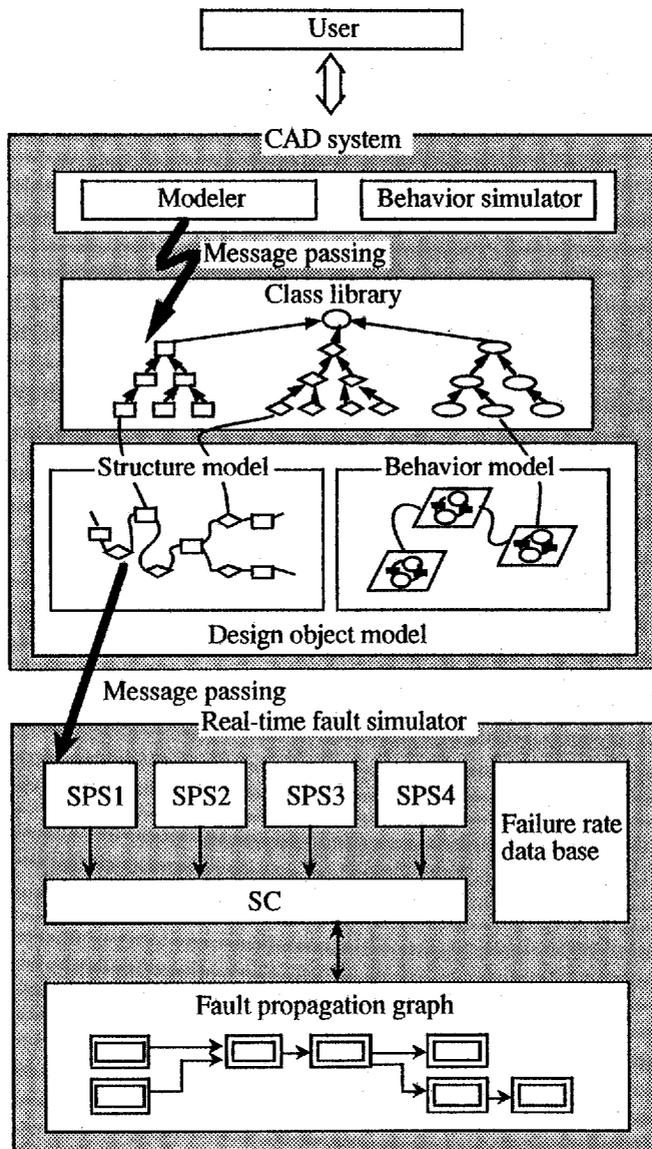


Fig.4.6 System construction

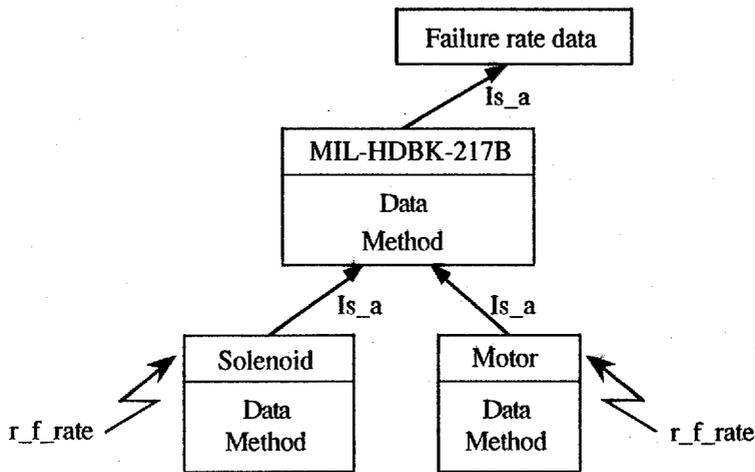


Fig.4.7 Failure rate data base based on object oriental concept

CADシステムによるモデリング作業中に、実体や関連のインスタンスが生成されると、対応するSPSにメッセージと必要な情報が送信される。受信したSPSは直ちに部分問題を処理し、その出力結果をSCに受けわたす。SCは受信した出力結果をもとに故障伝播グラフを更新する。部分問題の処理により生成された異常状態が他の実体や関連のインスタンスに影響を与える可能性があれば、さらに、SPSにメッセージを送信し、別の実体や関連のインスタンスに対してもその影響を調べることになる。異常状態の伝播はこのようなSPS間のメッセージの送信を次々に行うことにより実行される。また、実体や関連のインスタンスが削除された場合は、直接SCにメッセージを送信し、故障伝播グラフを修正することになる。なお、故障シミュレータには、故障モードの発生率がある閾値（以下、上閾値）より大きい場合に、設計者に対して警告メッセージを送る機能がある。この値は、モデリング作業を開始する時に設計者が設定する。

#### 4.5 システムの実行例

システムを用いて、変速歯車装置のモデリング作業と故障シミュレーションとの並行処理を試みた。なお、予め下閾値を $0.2 \times 10^{-6}/\text{hour}$ に、上閾値を $28 \times 10^{-6}/\text{hour}$ に設定した。モデリング途中の構造モデルの一部を Fig. 4.8 に、それに対応する故障シミュレーション結果の一部を Fig. 4.9 に示す。なお、Fig. 4.9 で節点に書かれた数値は故障モードの発生率を表し、単位は $10^{-6}/\text{hour}$ である。この図では、キーe16の破壊が歯車の角速度の低下 (e17とe18のstate1) を引き起こし、歯車e17,e18の破損も同様の現象を起こすことが示されている。

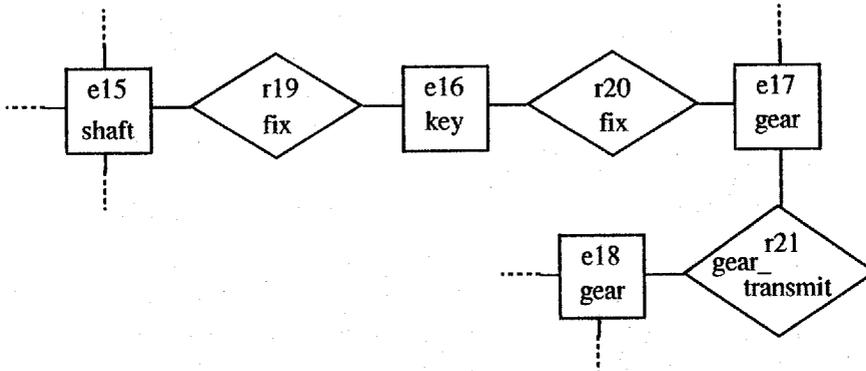


Fig.4.8 A part of structure model at a point of time in product modeling

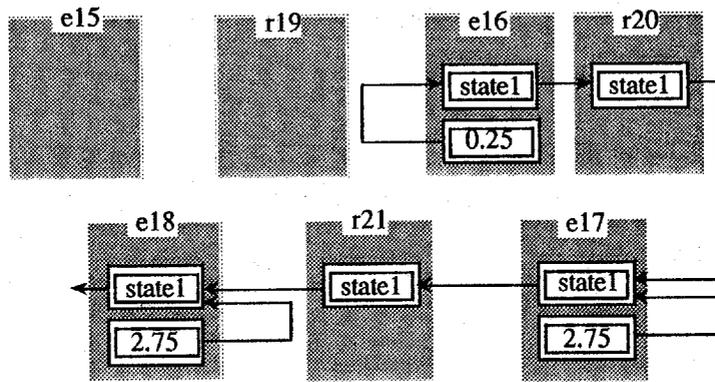


Fig. 4.9 A part of fault propagation graph

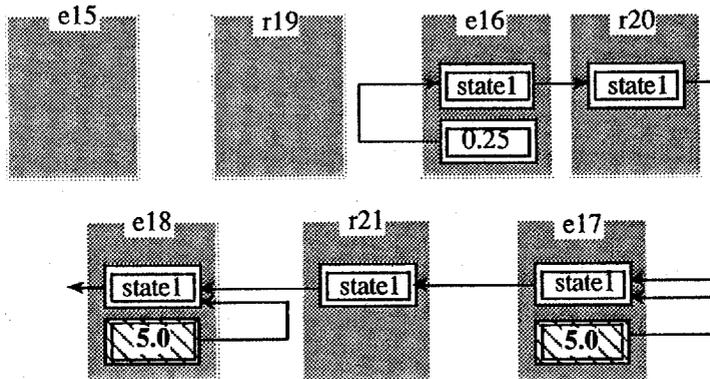


Fig.4.10 A part of fault propagation graph when e17 and e18 are specialized to spur gears

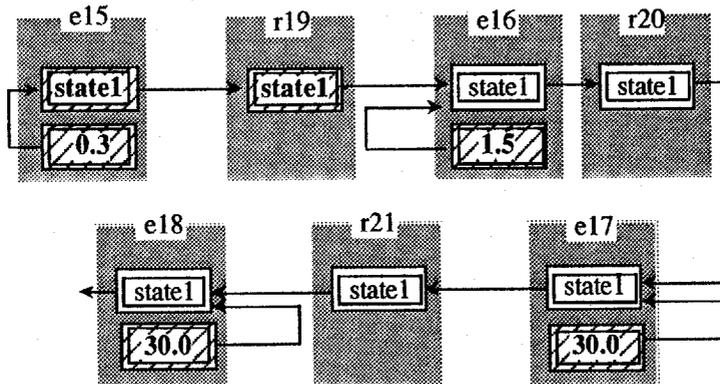


Fig.4.11 A part of fault propagation graph when the data on environment are changed

ここで、CADシステムの“インスタンスの具象化”機能を実行することによって、歯車e17とe18を平歯車に具象化した。すると、即座に具象化に対応する部分問題が処理され、その結果、故障伝播グラフは Fig. 4. 10 のようになった。この図で斜線部が変更された部分を表す。この図から、具象化によって、歯車e17,e18で起こりうる故障モード（歯車の破損）の発生率が増加したことがわかる。これは、歯車の中でも平歯車は他種のものに比べ故障モードの発生率が高いことを意味する。

\*\*\* WARNING \*\*\*

Failure\_rate of "gear\_is\_broken" ( e17 ) is  $30.0 \times 10^{-6}$  /hour

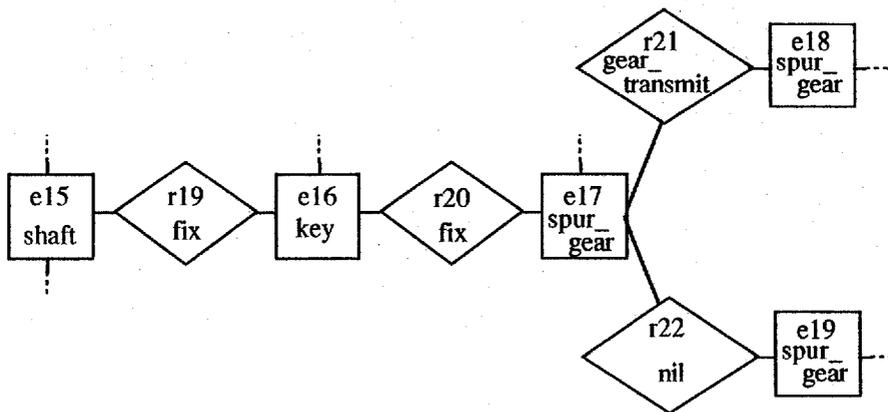
This can not satisfy the required condition.

Fig.4.12 An example of warning

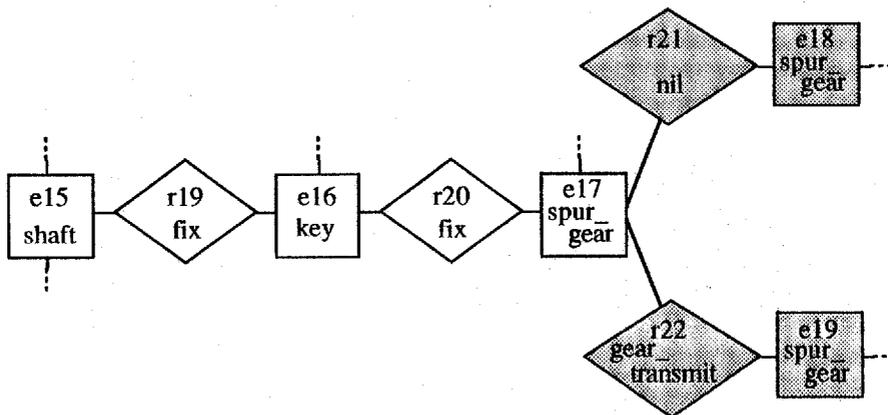
次に、製品の環境情報を変更した時の故障シミュレーション結果の一部を Fig. 4. 11 に示す。この環境情報の変更により、軸e15で起こる故障モードの発生率が下閾値 ( $0.2 \times 10^{-6}$ /hour) を越えたため、故障伝播グラフにその節点が追加され、その影響も推定された。また、平歯車 e17, e18 で起こる故障モードの発生率が上閾値 ( $28 \times 10^{-6}$ /hour) を越えたため、 Fig. 4. 12 に示すようなメッセージが設計者に示された。このような警告により、設計者は設計対象の不具合をモデルが完成する前に知ることが可能となる。

CADシステムの挙動シミュレーションによって、設計対象の構造が変化した場合の実行例を示す。Fig. 4. 13 に状態遷移前後の状態を表す構造モデルの一部を、Fig. 4. 14 にそれぞれの状態に対応する故障シミュレーション結果の一部を示す。Fig. 4. 13 で、色を付けた部分が挙動シミュレーションによって状態変化したオブジェクトを表しており、平歯車e17とe18間の接続関係がなくなり、新たにe17とe19間に歯車伝達という関係が生成されたことが示されている。Fig. 4. 14 では、設計対象がFig.4.13(a)の状態のときは平歯車e17の異常状態1が r21,e18に伝播され、Fig.4.13(b)の状態のときは、これがr22,e19に伝播されることが示されている。

このように、モデリング途中での設計対象の不具合の発見、設計変更や挙動シミュレーションによる故障伝播の変化の把握、を行うことができた。ただし、この図では構造モデルおよび故障伝播グラフの一部だけが示されている。設計者には、設計対象の変更に対応する故障伝播グラフの変化の様子をフィードバックするが、必要に応じて故障伝播グラフの任意の部分を提示することもできる。

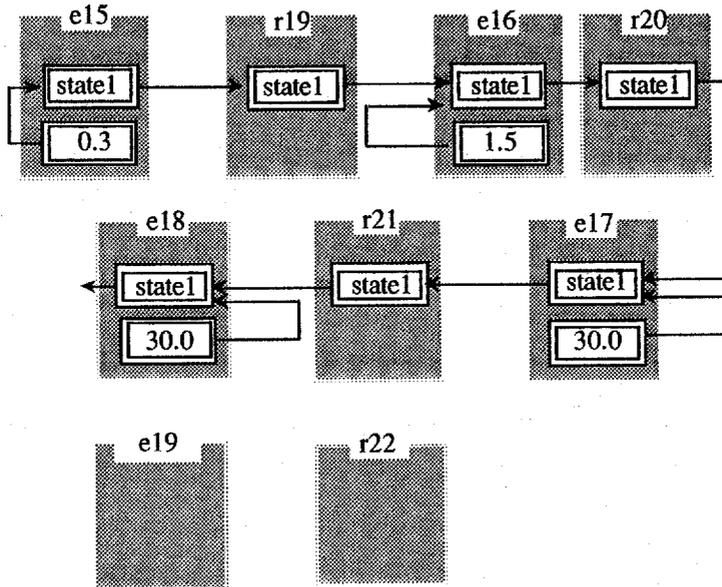


(a) A part of structure model before state transition

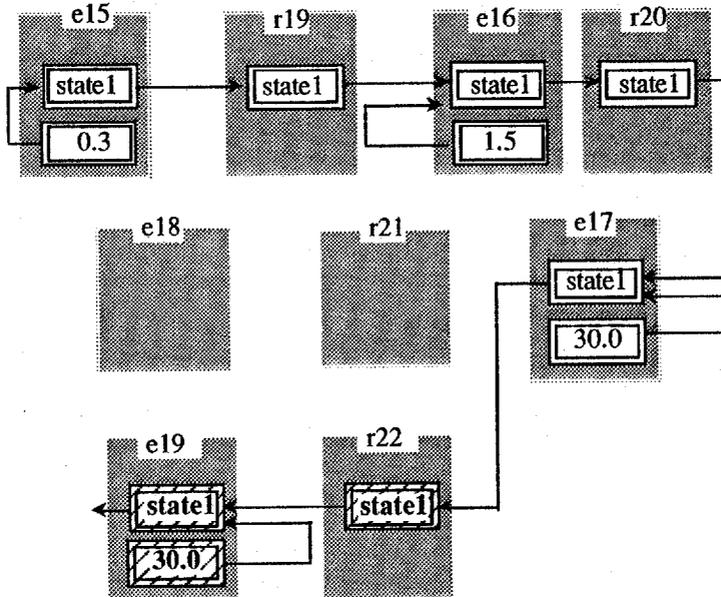


(b) A part of structure model after state transition

Fig.4.13 The change of structure model by state transition



(a) A part of fault propagation graph before state transition



(b) A part of fault propagation graph after state transition

Fig.4.14 The change of fault propagation graph by state transition

#### 4. 6 結 言

本研究で得られた結論を以下に示す。

- (1) モデリング作業中の設計者を信頼性の観点から支援するためには、モデリング作業と信頼性解析との並行処理が必要であり、これを実現するため、故障シミュレーションを部分問題に分解し、モデリング作業中のモデルに対する操作と連動して部分問題を処理する方法を示した。
- (2) 部品間の固着のような故障を扱うため、実体だけでなく関連に対しても起こりうる故障とその影響を調べる方法を示した。また、挙動シミュレーションによって状態変化した設計対象に対しても、故障シミュレーションを実行することにより、設計対象のとりうる各状態に対応した故障の伝播を把握できることを示した。
- (3) CADシステムと連動するリアルタイム故障シミュレータを試作した。このシステムを使用して、モデリング作業中に故障の伝播の様子を把握、および設計変更による影響の把握など、が可能であることを示した。

## 第5章 組立順序生成の効率化

### 5.1 緒言

最近、製品設計の段階で、製品の組立性を評価し、組立性改善のための適切な設計変更を実施する必要性が認識され<sup>(39)</sup>、AEM<sup>(59)</sup>やDAC<sup>(54)</sup>などの組立性評価法<sup>(53)</sup>が開発されるようになった。また、このための種々の計算機ツールも開発された。これらのツールのいくつかは製品のモデルを用いて組立順序を求め、それを基に組立可能性の検証<sup>(49)-(51)</sup>や組立工程の評価<sup>(42),(52),(57)</sup>を行う。しかし、これらのツールでは、組立順序の生成に多くの計算時間を要する。

一方、製品設計では、次のような場合に設計対象の部分的変更がよく行われる。すなわち、

- (1) 組立順序の解析や他の統合化作業により設計対象の不具合が発見された場合の設計変更、
- (2) 過去の設計例を有効に利用する修正設計。

このような場合、設計変更前の設計対象の組立順序情報が存在すれば、それを有効に利用して、効率よく変更された設計対象の組立順序を生成できる可能性がある。本章では、第3章で提案したリアルタイム設計処理の概念を適用することにより、設計変更前の組立順序情報を利用する方法を提案する。なお、リアルタイム設計処理は、本来、設計対象のモデリング作業と統合化作業との並

行処理の実現を目的とするが、組立順序生成への適用に対しては部分問題の処理時間が長く、並行処理は不可能である。本章はこのような場合でも、リアルタイム設計処理の適用が有効であることを示すものである。

本章では、まず、5.2節で、組立順序を求める問題の部分問題への分割を示し、部分問題の処理方法を提案する。次に、5.3節ではリアルタイム設計処理を適用した組立順序生成アルゴリズムを提案する。最後に、5.4節では、CADシステムと統合された組立順序生成システムを示し、実行例によりその有効性を示す。

## 5.2 組立順序生成問題

### 5.2.1 限定条件

組立順序の生成に関する研究はこれまで盛んに行われている(49)-(52),(56),(57),(98)-(116)。ここで提案する方法も、既存の多くの研究と同様に、以下の限定条件を設ける。

- (1) 設計対象を構成する実体は全て剛体とし、組立と分解の可逆性を仮定する。つまり、塑性変形のような不可逆な過程を扱わない。ただし、接着や溶接は可逆であるものとして扱う。
- (2) 実際に組み立てる時の組立手段、例えば組立に使用する機械、を考慮に入れない。
- (3) 組立は、二つの実体（部品または部分組立品）の組み付けを繰り返すことにより行われるものとする。この時、一方の実体を静止させ、もう一方を移動させる。
- (4) 実体を組み付ける時は、実体を直進運動によってのみ移動させる。回転運動による実体の姿勢の変化を考慮に入れない。ただし、ボルトやナットは直進移動によって組み付けられるものとして扱う。

以上の限定条件の中で、(1),(4)は実際の製品の組立において満足されることは

少なくない。また、(3)は組立機械による組立では普通満足される。本手法では、(1)の仮定のもとで、設計対象の分解順序を求め、その逆順序として組立順序を得る。

### 5. 2. 2 組立順序生成問題の分割

組立順序を生成する問題にリアルタイム設計処理を適用するためには、この問題の分割が可能でなければならない。Fig.5.1に「ある設計対象Aの組立順序を求める」という問題S0の分割を示す。この図のように、この問題の分割は一通りではないため、AND/ORグラフ<sup>(117)</sup>で表される。この図でコネクタは問題の一つの分割方法を表す。コネクタでつながった親節点が表す問題は、そのコネクタの子節点が表す部分問題が解をもてば、解をもち、子節点の部分問題

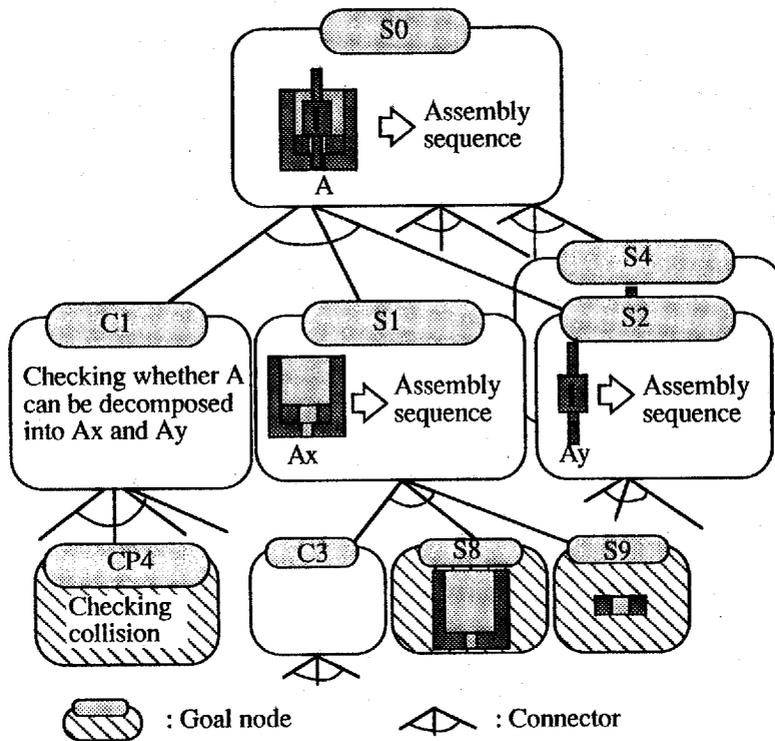


Fig. 5. 1 Decomposition of problem for generating assembly sequence

の解を統合することにより、親節点の問題の解が得られる。たとえば、問題S0は二つの実体Ax,Ayへの分解の可能性を調べる部分問題C1と、Axの組立順序を求める部分問題S1およびAyの同様の部分問題S2に分割され、部分問題C1,S1,S2の解を統合することにより、問題S0の一つの解が求まる。この問題S0は複数のコネクタと接続されているが、これはS0の分割方法が複数あることを表す。分解された実体の組立順序を求める問題（たとえば、S1,S2）は問題S0と同様により小さい部分問題に分割される。また、分解可能性を調べる問題（たとえば、C1）も以下の部分問題に分割される。

- (1) 問題が対象とする二つの実体（部分組立品または部品）のうち、一方を静止実体、他方を移動実体に決定する問題（CP1）
- (2) 移動実体の移動方向を決定する問題（CP2）
- (3) 移動実体の移動距離を算出する問題（CP3）
- (4) 移動実体の移動時における静止実体との衝突をチェックする問題（CP4）

これらの部分問題CP1~CP4の解を統合することにより、分解経路が分解可能性を調べる問題の解として得られる。組立と分解の可逆性の仮定より、その逆が組立経路となる。なお、以下では、AND/ORグラフを構成する問題のうち、実体の組立順序を求める問題をS問題、分解可能性を調べる問題をC問題と呼ぶことにする。

このようなAND/ORグラフの探索によって解グラフが得られる。この解グラフは問題S0から解をもつ終端節点へのパスを表す部分グラフであり、終端節点以外の各節点はその節点を親とするコネクタを一個だけもつ。なお、終端節点には、C問題の部分問題、および一個の部品を対象とするS問題があり、後者は絶えず可解であるとする。このような解グラフのトップダウンパスが設計対象の分解順序を表す。また、その逆が組立順序となる。もし、解グラフが得られなければ、その設計対象は組立不可能である。

このAND/ORグラフの探索は、各部分問題を順次分解し、処理することによって行われる。しかし、修正設計などにより設計変更された設計対象に対して

は、リアルタイム設計処理を適用することにより、全ての部分問題を処理する必要はなくなる。すなわち、設計変更によって新たに処理する必要のできた部分問題のみを処理し、他の部分問題は設計変更前のAND/ORグラフの情報を利用することにより新たに処理する必要はなくなる。

### 5. 2. 3 分解可能性の検証方法

分解可能性を調べるC問題は、分解すべき実体の移動経路を探索することにより処理される。もし、分解に至る移動経路が見つからなければ、分解不可能と判定する。ただし、分解すべき実体が十分遠方まで移動できれば分解できたと判断する。

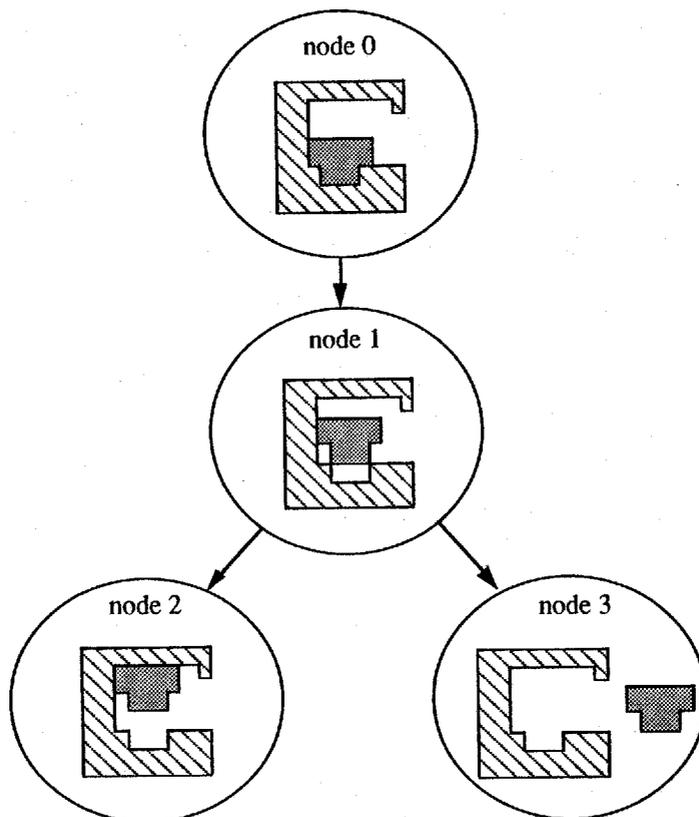


Fig. 5. 2 A graph for searching disassembly path

移動させる実体の移動経路は、Fig. 5.2 に示すように、節点を移動実体の位置とし、枝を実体の移動とするグラフ（以下、経路探索グラフ）を探索することにより求められる。ここではA\*アルゴリズム<sup>(117)</sup>を用いてこのグラフを探索し、分解の難易性を表すコストを最小化する経路を求める。ただし、コストを移動実体の移動方向の変更回数と移動距離を用いて算出する。Fig.5.2の例では、node0 → node1 → node3 へ至る経路が解として求まる。

この経路探索時に、上述の4種類の部分問題CP1～CP4が処理される。移動実体を選定する問題CP1は、経路探索を始めるときに、実体の構成部品数などの情報を基に人間とシステムとの対話で処理される。CP2～CP4は経路探索グラフの節点を展開し、新しい子節点を作るときに処理される。ここでは、これらの問題を第3章で示したCADシステムによって作成されるモデルを用いて処理する。モデルを用いた部分問題CP2～CP4の処理方法を以下に示す。

#### (1) 移動方向の決定

経路探索時に経路探索グラフのある節点nを展開する場合、移動実体が節点nの位置から移動可能な方向をCP2の処理により求め、求めた方向に対応する子節点を生成する。ただし、子節点が表す移動実体の位置はCP3、CP4の処理により決定される。この移動可能方向は、移動実体と静止実体との幾何学的拘束関係から求めることができる。幾何学的拘束は構造モデルではフィーチャ間の関係として記述されている。ただし、フィーチャ<sup>(3)</sup>は穴、溝などの形状の特徴であり、部品の属性として記述されている。また、フィーチャ間の関係は穴とペグとの接続関係などであり、部品間の関連に属性間の制約関係として記述されている。ここでは実体間に存在する全ての幾何学的拘束はフィーチャ間の関係として記述されているものとして、フィーチャ間の関係を基に移動可能方向を算出する方法を提案する。

今、移動実体 $E_m$ を構成する部品を $P_{mi}$  ( $i=1\sim N$ )、静止実体 $E_s$ を構成する部品を $P_{sj}$  ( $j=1\sim O$ ) とする。 $P_{mi}$ と $P_{sj}$ との間に存在するフィーチャ間の関係の集合を $F_{ij}$ とすると、 $E_m$ と $E_s$ 間に存在するフィーチャ間の関係集合 $F_m^s$ は次式で表

せる。

$$F_m^s = \bigcup_{(p_{mi}, p_{sj}) \in F_m \times F_s} F_{ij} \dots \dots \dots (5.1)$$

これらのフィーチャ間の関係は、移動実体の移動により、順次解除されていく。経路探索グラフの節点nが表す状態において、 $F_m^s$ の中で解除されていないフィーチャ間の関係の集合を ${}_nF_m^s$ とする。フィーチャ間の関係 $f \in {}_nF_m^s$ によって拘束されない方向を表すベクトルの集合を $A_f$ とすると、節点nが表す状態において、移動実体が静止実体によって拘束されない方向（移動可能方向）を表すベクトルの集合 ${}_nA_m^s$ は次式で求まる。

$${}_nA_m^s = \bigcap_{f \in {}_nF_m^s} A_f \dots \dots \dots (5.2)$$

なお、 $A_f$ は部品間の関連に属性として記述されている。節点nがFig.5.2のnode2のように静止実体との衝突によって生じた状態を表す場合は、その衝突を引き起こした移動方向を ${}_nA_m^s$ から取り除く。もし、経路探索グラフの出発節点において ${}_nA_m^s = \phi$ であれば、その時点で分解不可能であることが分かる。

このようにフィーチャ間の関係を用いて移動可能方向を算出できるが、経路探索を続けると、移動実体と静止実体との間にフィーチャ間の関係がなくなったが、分解できていないという状態が起こる場合がある。この場合は、その状態の一つ前の状態において算出された移動方向を用いるが、人間がシステムとの対話により移動方向を与えることもできる。

## (2) 移動距離の算出

節点nの状態から方向 $a \in {}_nA_m^s$ に移動実体を移動させることにより、フィーチャ間の関係が新たに解除される可能性がある。ここでは最初にフィーチャ間の関係が解除されるまでの距離を移動距離 $d_a$ とする。部分問題CP3はこの距離を

以下の方法で求めるものである。移動距離 $d_a$ はフィーチャ間の関係 $f \in {}_n F_m$ を解除するまでの距離を $d_{af}$ とすると次式で表せる。

$$d_a = \min_{f \in {}_n F_m} d_{af} \dots \dots \dots (5.3)$$

ただし、 $\min_{a \in \Lambda} \alpha_a$  は $\Lambda$ の各元 $a$ に対応する $\alpha_a$ の中で最小のものを選択することを意味する。ここで、 $d_{af}$ は以下のように求まる。

今、Fig.5.3に示すように、フィーチャ間の関係 $f$ が対象とするフィーチャのうち、静止実体側のフィーチャを $f_s$ 、移動実体側のフィーチャを $f_m$ とし、 $f_s$ 、 $f_m$ を構成する頂点の位置ベクトルをそれぞれ $v_{si}$  ( $i=1 \sim N1$ )、 $v_{mj}$  ( $j=1 \sim N2$ )とする。これらの位置ベクトル $v_{si}$ 、 $v_{mj}$ の $a$ への正射影の長さ $v_{si}$ 、 $v_{mj}$ を次式で求める。

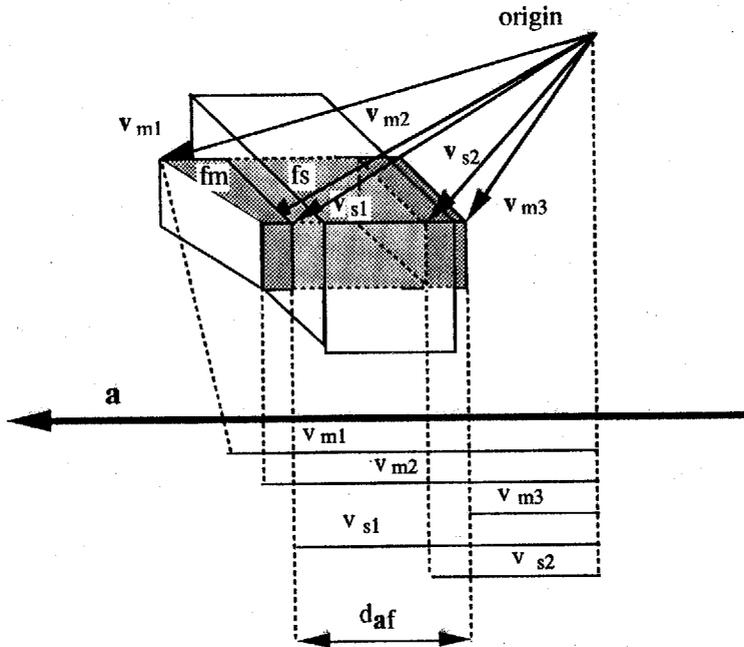


Fig. 5. 3 Calculation of a distance for removing a feature relation

$$v_{si} = \frac{(v_{si}, \mathbf{a})}{|\mathbf{a}|} \dots \dots \dots (5.4)$$

$$v_{mj} = \frac{(v_{mj}, \mathbf{a})}{|\mathbf{a}|} \dots \dots \dots (5.5)$$

これらの正射影の長さから、 $d_{af}$ は以下のように求まる。

$$d_{af} = \max\{v_{s1}, v_{s2}, \dots, v_{si}, \dots, v_{sN1}\} - \min\{v_{m1}, v_{m2}, \dots, v_{mj}, \dots, v_{mN2}\} \dots (5.6)$$

また、フィーチャ間の関係が全て解除されたが分解できていない場合は、十分大きな値を暫定的に移動距離として与える。もし、静止実体と衝突せずにこの距離だけ移動できた場合は、分解できたと判断する。この場合、節点 $n$ の位置から分解されるまでに最低限必要な移動距離を上述の距離の計算と同様な方法で計算する。すなわち、静止実体を構成する全ての頂点の位置ベクトルの $\mathbf{a}$ への正射影の長さを求め、その最大値を $d_{amax}$ とし、移動実体を構成する全ての頂点の位置ベクトルの $\mathbf{a}$ への正射影の長さを求め、その最小値を $d_{amin}$ とすると、移動距離は $d_{amax} - d_{amin}$ で求まる。

### (3) 衝突チェック

部分問題CP4は、移動実体が、節点 $n$ が表す位置から(2)で求めた移動距離 $d_a$ だけ静止実体と衝突せずに移動可能かどうか調べるものである。ここでは、ソリッドモデルを用いて交点計算法<sup>(118)</sup>に基づき衝突をチェックする。この方法では、移動実体の構成部品と静止実体の構成部品の全ての組み合わせに対して、部品間の衝突チェックを行うことになる。今、部品の組み合わせ $(p_{mi}, p_{sj}) \in E_m \times E_s$ に対して算出された衝突までの距離を $d_{ij}$ とする。ただし、衝突が起こらない場合は、 $d_{ij}$ に十分大きな値を与える。移動実体と静止実体との衝突までの距離 $d_c$ は次式で求まる。

$$d_c = \min\{d_{11}, d_{12}, \dots, d_{NO}\} \dots \dots \dots (5.7)$$

もし、

$$d_c < d_a \quad \dots \dots \dots (5.8)$$

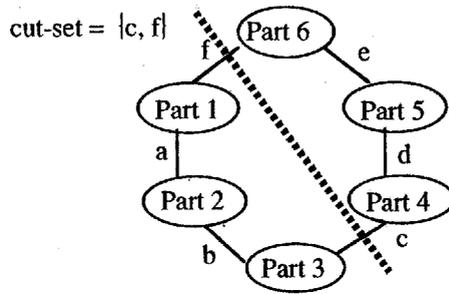
であれば、移動途中で衝突することになる。この場合、 $d_c$ を移動実体の移動距離とし、移動実体の衝突した位置を表す節点を節点 $n$ の子節点として経路探索グラフに生成することになる。移動途中で衝突がなければ、移動実体が $d_a$ だけ移動した位置を表す節点を節点 $n$ の子節点として生成する。なお、探索中のコスト値によっては、衝突位置から更に移動方向を変更し、探索を続ける場合もある。

### 5.3 AND/ORグラフ探索による組立順序の生成

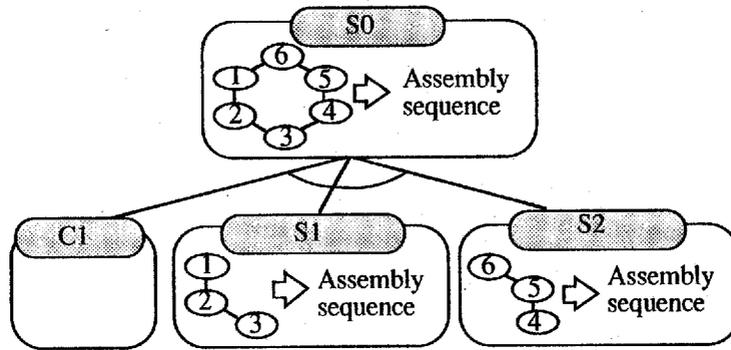
#### 5.3.1 探索空間の縮小化

AND/ORグラフの探索を、次々と問題を分割し、コネクタを生成することにより行うことができる。AND/ORグラフの各S問題を分割し、その問題のコネクタを生成するためには、どのような二つの実体への分解可能性を調べるかを決定する必要がある。この決定は、Fig.5.4のようにS問題が対象とする実体を部品接続グラフ<sup>(119)</sup>で表し、それに対して、初等的なカットセットを求めることによって行える。この図では、初等的なカットセット  $\{c, f\}$  により、問題S0がC1, S1, S2に分割されることが示されている。

既存の研究の中には、全ての初等的なカットセットを求め、それぞれに対して現実に分解可能かどうかを調べるものがある<sup>(57)</sup>。この場合、初等的なカットセットに対応するコネクタを全て生成することになる。しかし、構成部品数の多い設計対象に対してこのような処理を行うと、AND/ORグラフの規模が大きくなり、グラフの探索に多くの時間がかかる。例えば、Fig. 5.4の例でも、(a)の部品接続グラフに対する初等的なカットセットの数は15個あるため、S0のコネクタを15個生成することになり、AND/ORグラフの規模は大きくなる。



(a) Part connectivity graph



(b) Decomposition of problem

Fig. 5. 4 Decomposition of S problem by using cut-set

そこで、ここでは探索空間を縮小するため、部品間の関連の優先関係を用いて初等的なカットセットをいくつか生成し、それを基にコネクタを予め設定された数だけ生成する。グラフ探索中に、生成されたコネクタでは組立順序が得られないことが判明すれば、さらにコネクタを設定数だけ追加する。ただし、この数は人間が設定する。

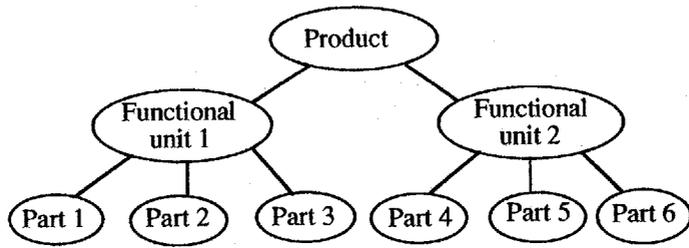
この優先関係は、二つの関連間の2項関係であり、製品の組立を行うときに、一方の部品間の関連が他方のものより早く現れる（優先される）可能性があることを表す。逆に、製品の分解時は、優先関係の低い部品間の関連が先に解除される可能性がある。S問題のコネクタは実体の分解を表すため、優先関係の低い関連から構成される初等的なカットセットが解を得るのに有効なコネクタ

を生成すると考えられる。

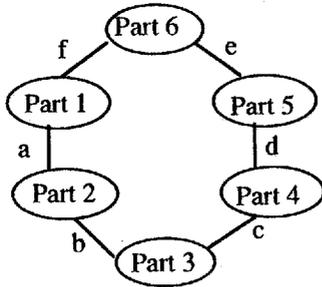
このような優先関係は以下の製品情報と関連する。

- (1) 部品の種類
- (2) 部品間の関連の種類
- (3) 製品の階層構造
- (4) 精度情報
- (5) 部品間の幾何学的な拘束関係

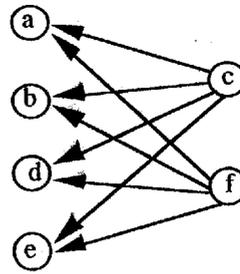
これらのうち、(5)はAND/ORグラフを構成するC問題の処理において考慮される。このため、ここではS問題の分割のための優先関係を(1)～(4)の情報を用いて生成する。優先関係と(1)～(4)の情報との関係は体系化されていないため、工程設計者のもつヒューリスティクスを利用する必要がある。本研究では、このようなヒューリスティクスをルール形式で記述し、(1)～(4)の情報に適用することにより優先関係を得る。ただし、現段階では製品の階層構造や組立精度などについて一般的によく知られたヒューリスティクスのみをルール化している。Fig. 5.5に階層関係を基に生成した優先関係の例を示す。これは、機能ユニットの組立を優先するというヒューリスティクスに基づいている。この図の(c)のように、ルールにより生成された優先関係は、部品間の関連を節点とし、優先関係を枝とする有向グラフ（枝の始点が終点より優先される）で表される。このような優先関係グラフは、優先関係の矛盾を表すループを含む可能性がある。この場合は、人間が適用ルールを評価して適切な枝を削除し、ループを分解する。



(a) Part-of hierarchy of a product



(b) Part connectivity graph



(c) Precedence graph among connections

Fig. 5. 5 An example of precedence graph based on Part-of hierarchy

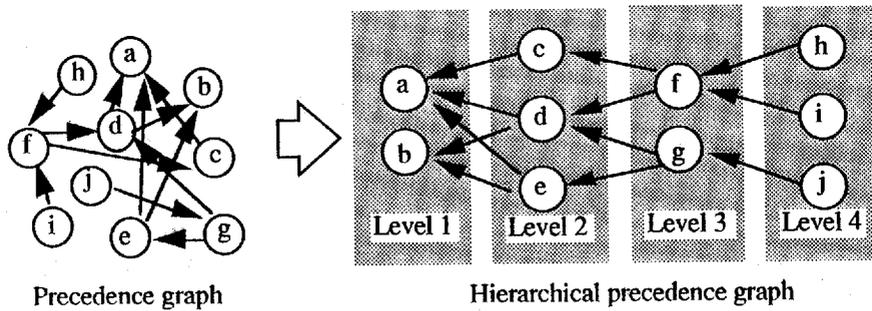


Fig. 5. 6 Generation of hierarchical precedence graph

このような優先関係グラフを用いたS問題のコネクタの生成は以下のプロセスで行われる。

Step1: 分割したいS問題と対応する優先関係グラフを求める。

Step2: 優先関係グラフを Fig. 5. 6のように階層化し、各節点に優先度を与える。

ただし、この図はレベル値が小さいほど優先度が高いことを表している。

Step3:最も優先度の低い節点の集合を求め、それをPとする。Fig.5.6に示す例では $P=\{h,i,j\}$ となる。

Step4:Pの部分集合が部品接続グラフにおいて構成可能な初等的なカットセットを全て求め、その初等的なカットセットの集合をCSとする。ただし、S問題を親節点とするコネクタが既に存在する場合は、そのコネクタと対応する初等的なカットセットをCSから取り除く。

Step5:CSに含まれる初等的なカットセットの数が予め設定された数より少ない場合は、Pに優先度の低い節点を追加し、Step4の処理を行う。CSに含まれる初等的なカットセットの数が設定された数より多い場合は、設定された数だけ選択する。

Step6:求まった初等的なカットセットを基に、S問題を分割し、コネクタを生成する。

なお、後述するシステムでは、このような処理をユーザとの対話で行っている。

### 5.3.2 組立順序の探索アルゴリズム

AND/ORグラフの探索のため、AO\*アルゴリズム<sup>(17)</sup>をベースとした探索アルゴリズムを開発した。これは、各節点にコストを与え、出発節点S0のコストを最小とする解グラフを縮小された探索空間内で求めるものである。このアルゴリズムを以下に示す。また、探索途中のAND/ORグラフの例をFig.5.7に示す。

Step1:出発節点S0だけからなる探索グラフGを作る。

Step2:S0にsolvedまたはunsolvableのラベルがつけられるまで以下の処理を行う。なお、solved、unsolvableはそれぞれ、そのラベルのつけられた節点が解けたこと、および、解くことが不可能なことを意味する。S0にsolvedのラベルがついている場合は、S0から発する印↓のついたコネクタをたどることにより、解グラフが得られる。なお、この印は以降のステップでつけられる。

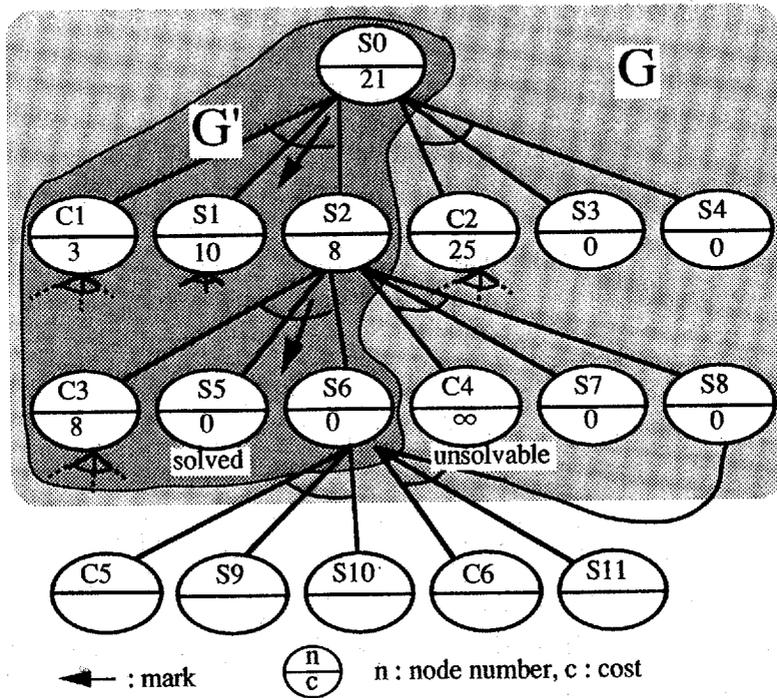


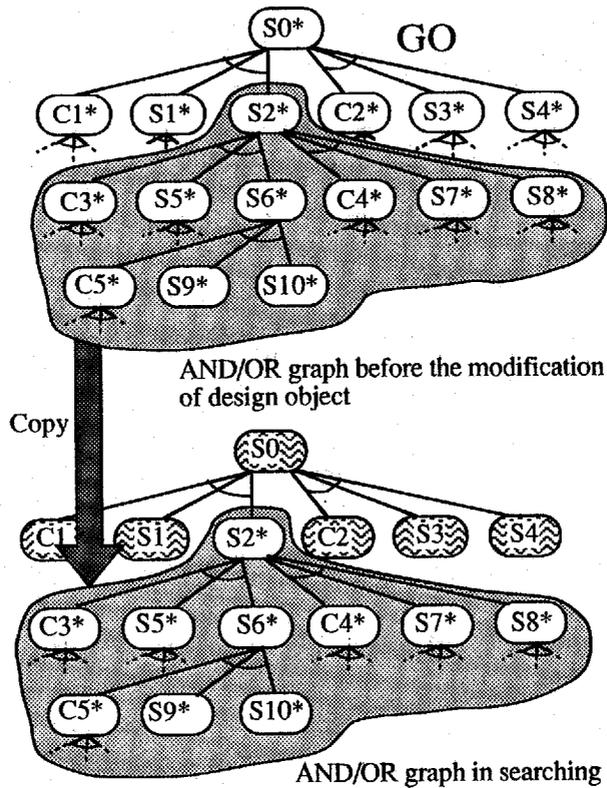
Fig. 5. 7 AND/OR graph search by using costs

Step3: Fig. 5. 7のように、GにおいてS0から発する印↓のついたコネクタをたどって、部分グラフG'を求める。

Step4: G'においてall\_generatedとsolvedのどちらのラベルもつけられておらず、かつ終端節点でない節点から、節点のコストが最小のものを選択し、それをnとする。ただし、all\_generatedは、その節点を親節点とするコネクタが既に全て生成されていることを意味する。Fig. 5.7の例では、S6がnとして選択される。

Step5: 優先関係グラフを用いて、nを親節点とするコネクタをいくつか作る。新たに生成したコネクタと既に存在していたnのコネクタとの合計数がnの生成可能なコネクタ数の最大値と同じであれば、nにラベルall\_generatedをつける。

Step6: 新たに生成したコネクタの子節点の中で、Fig. 5. 7のS8のように既にGの中に存在する子節点については、それと統合する。Gに含まれていない各子節点ncについては以下の処理を行う。



⊙ : Problem whose input data contain modified design object data

Fig. 5. 8 Copy of partial graph in AND/OR graph search

- (1) もし、設計変更前の設計対象に対するAND/ORグラフGOが存在すれば、ncと同じ問題がGOの中に存在するかを調べる。もし存在すれば、Fig. 5. 8のようにGOにあるその問題（この図ではS2\*）より下位のグラフの情報を探索中のグラフにコピーする。もし、GOが存在しないか、または存在してもncと同じ問題がGOにない場合は、次の(2),(3)の処理を行う。
- (2) ncがS問題であれば、ncにコスト $q(nc)=0$ を与える。ncが終端節点であれば、それにsolvedのラベルをつける。
- (3) ncがC問題であれば、これを部分問題に分割し処理する。この時、分割された部分問題に対しても(1)と同様の処理を行う。ncが解をもてばncにsolvedの

ラベルをつけ、分解経路の探索によって求まるコストをncのコストとする。もしncが解をもたない場合（分解に至る経路が見つからない場合）はコスト $q(nc)=\infty$ を与え、ncにunsolvableのラベルをつける。

新たに生成した全ての子節点に対して、(1), (2), (3)の処理を実行したら、これらの節点をグラフGに加える。

Step7: 節点nだけからなる集合Nを作る。以下の処理により、Step6で求めた子節点のコストを用いて、グラフGの上部の節点のコストを変更する。

Step8: Nが空になるまで以下の処理を行う。空になればStep2に戻る。

Step9: Nから任意に一つの節点mを取り除く。

Step10: mを親節点とする各コネクタiの子節点集合を $\{ni1, ni2, ni3\}$ とし、次式によりコネクタiのコスト $qi(m)$ を求める。

$$qi(m)=q(ni1)+q(ni2)+q(ni3) \quad \dots \dots \dots (5.9)$$

ただし、 $q(ni1)$ ,  $q(ni2)$ ,  $q(ni3)$ の中で一つでも $\infty$ のものがあれば $qi(m)=\infty$ とする。これはコネクタiが表す分解が不可能であることを意味する。全てのコネクタの $qi(m)$ が $\infty$ であり、mにall\_generatedのラベルがついていれば、mはいかなる場合でも解をもたないので、節点mのコスト $q(m)$ を $\infty$ とし、mにunsolvableのラベルをつける。全てのコネクタの $qi(m)$ が $\infty$ であり、mにall\_generatedのラベルがついていない場合は、mは未生成のコネクタによって解を与えられる可能性があるので、とりあえず $q(m)=0$ とする。また、 $qi(m) \neq \infty$ のコネクタが存在すれば解が存在するので、次式により $q(m)$ を求める。

$$q(m)=\min\{q1(m), q2(m), \dots, qk(m)\} \quad \dots \dots \dots (5.10)$$

この時、その最小コストをmに与えたコネクタに印↓をつける。もし、この印をつけたコネクタの子節点が全てsolvedであれば、節点mにもsolvedのラベルをつける。

Step11: もし、mがsolvedというラベルをもつか、またはmのコストが変更された場合は、mの親節点で、mとの間が印↓のついたコネクタで結ばれているも

のをすべてNに加える。

Step12:Step8に戻る。

このアルゴリズムは、AO\*アルゴリズムと比較すると以下の特徴をもつ。

- (1) Step5でS問題を分割する時、全てのコネクタを生成するのではなく、優先関係グラフを用いて適当な数のコネクタだけを生成する。
- (2) C問題は生成されると即座に分割され処理される。
- (3) Step4で選択される節点はG'の葉節点だけでなく、G'の中間節点のうちまだ展開可能なもの (all\_generatedのラベルがついていないもの) も含まれる。このため、節点のコスト値によっては、中間節点にさらにコネクタが追加される場合がある。
- (4) 修正設計などで設計変更された設計対象に対しては、設計変更前に存在していたAND/ORグラフの情報を有効に利用する。

なお、この探索アルゴリズムは、全てのコネクタを生成するとは限らないため、最適な組立順序が求まらない場合がある。しかし、ヒューリスティックな優先関係を基にコネクタを生成しているため、満足解は求まる。また、組立不可能な設計対象に対して本アルゴリズムを適用すると、出発節点にunsolvableのラベルがつけられるまで、コネクタが追加され、出発節点に対しては全てのコネクタが生成される。これは、結局、探索空間は縮小されないが、組立可能性は必ず検証できることを意味する。

### 5. 3. 3 設計変更前のAND/ORグラフの利用

上述の特徴(4)は、設計変更を行った場合、リアルタイム設計処理の概念を適用することにより、新たに処理すべき部分問題のみを処理するというものである。この方法を以下に示す。

まず、設計対象を構成する部品の中で形状等が変更された部品のリスト（以下、変更リスト）を作成する。ただし、はめあいのように複数の部品と関連する情報が変更された場合は、その変更と関連する全ての部品を変更リストに入

れる。次に、AND/ORグラフの探索中に新たに節点が生成されると、その節点が表す部分問題の処理に必要な設計対象の情報に、変更リストの情報が含まれるかどうかを調べる。含まれていれば、それは新たに処理すべき部分問題である。含まれていなければ、それと同じ問題が設計変更前のAND/ORグラフGOにおいて存在するかどうか調べる。存在しなければ、それも新たに処理すべき部分問題である。同じ問題がGOに存在する場合、同一であると判断された二つの問題では優先関係の要因になる設計対象情報が同じであるため、その問題より下に生成されるコネクタも同じになる。このため、GOのその問題より下位のグラフを全て探索中のグラフにコピーできる。

同じ問題がGOに存在するかどうかの判断は、問題が対象とする実体が同じものをGOから探すことにより行われる。なお、C問題から分割される全ての部分問題に対してこのような処理を行うと、かえってAND/ORグラフの探索に時間がかかる可能性がある。このため、C問題の部分問題の中では、計算時間のかかる部品間の衝突チェックに対してのみ、このような処理を行う。なお、設計変更された部分によっては、設計変更前のAND/ORグラフをほとんど利用できない場合がある。この場合は、生成される組立順序は設計変更前のものと大幅に変わることになる。

#### 5. 4 計算例

本方法の有効性を示すため、第3章で示したCADシステムと統合された組立順序生成システムを構築した。本システムの構成をFig.5.9に示す。この図のように組立順序生成システムは、優先関係生成モジュール、探索モジュール、およびデータベースから構成される。データベースには、優先関係グラフ、設計変更に関する情報、設計変更前のAND/ORグラフ、が格納されている。

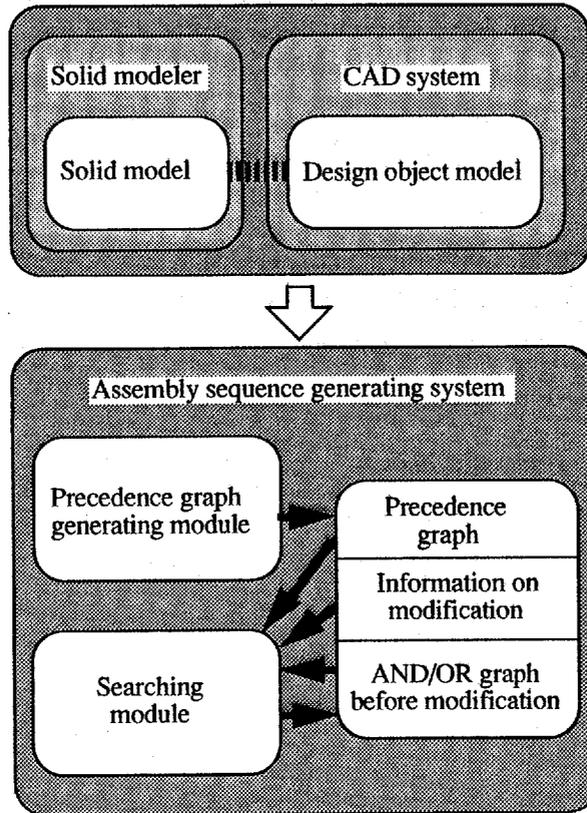


Fig. 5.9 System construction

システムの有効性を示すため、ここでは二つの実行例を示す。一つは Fig.5.10に示す4つの部品から構成される組立品を対象とした。この組立品の部品③の形状が変更された場合のAND/ORグラフの探索例をFig.5.11に示す。この図で、各節点に書かれた数値は探索によって算出されたコストを表し、太線で描かれた部分グラフはS0に最小コストを与える解グラフを表す。また、色をつけた節点 (S1,S3) は設計変更前のAND/ORグラフの問題の解を利用できたことを示す。なお、この図には示されていないが、C1の部分問題である部品間の衝突チェックも、設計変更前の解を利用することができた。計算時間は、変更前の情報を利用しない場合に比べ、約1/9に減少した。このシステムでは生成された組立順序や組立経路をCRT画面上で確認できる。画面に表示された

組立順序をFig.5.12に示す。また、Fig.5.11の節点C1の解として求まる組立経路の表示例をFig.5.13に示す。

もう一つの実行例は、Fig.5.14に示す歯車ポンプを対象としたものである。ただし、使用したコンピュータの機能上の制約から、この対象に対しては幾何モデルを扱うことが不可能であった。このため、衝突チェックなどの幾何モデルを必要とする部分問題の処理に対しては、人間が解を与えた。Fig.5.15はこの組立品の部品⑩、⑪の形状が変更された場合の探索例である。この例では、設計変更前のAND/ORグラフを利用しない場合に比べて、処理した部分問題の数が1/7に減少した。なお、本システムでは、設計者がシステムの生成した組立順序以外のものを知りたい場合は、設計者が適当な節点にunsolvableのラベルをつけることにより、探索を続行させることになる。

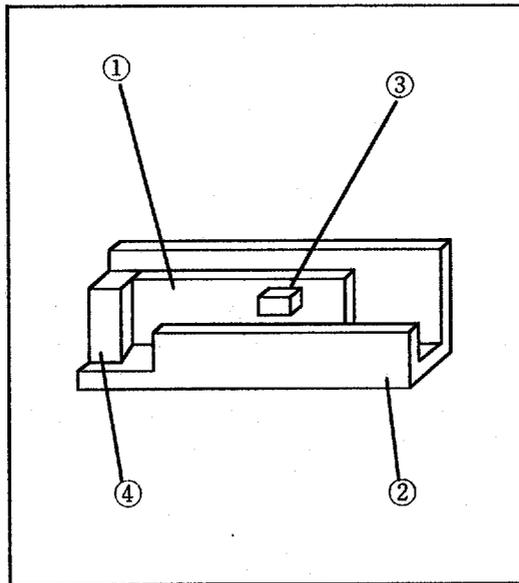


Fig. 5. 10 A design object whose part ③ is modified

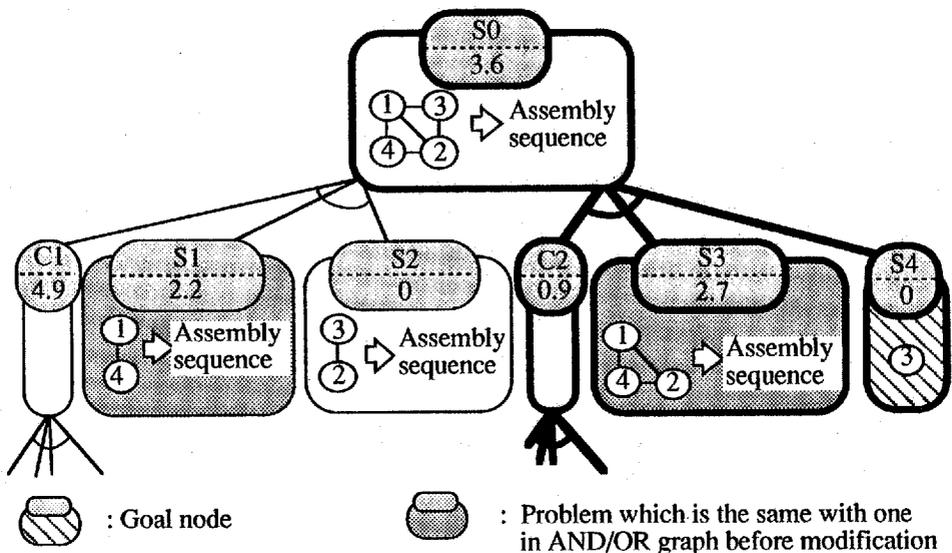


Fig. 5. 11 An example of AND/OR graph search for a modified design object

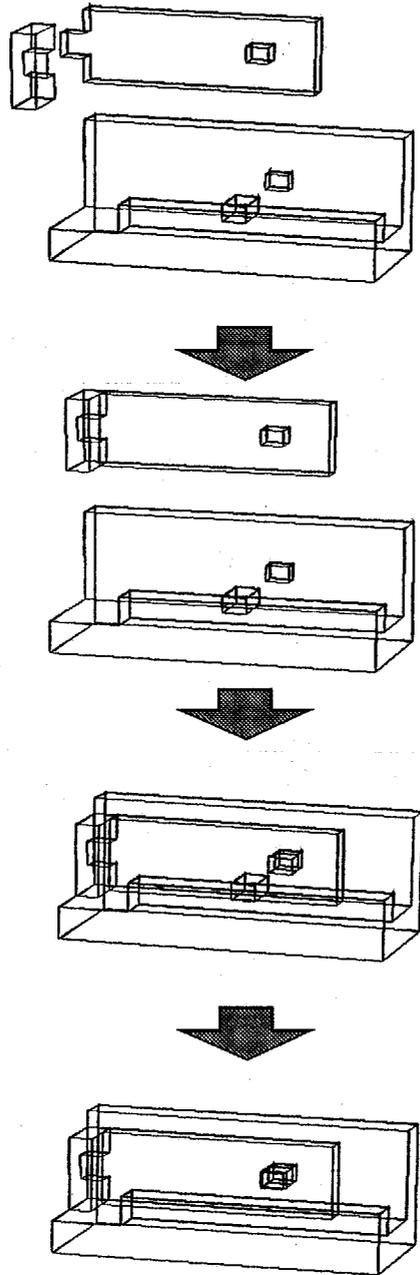


Fig. 5. 12 An assembly sequence generated by the system

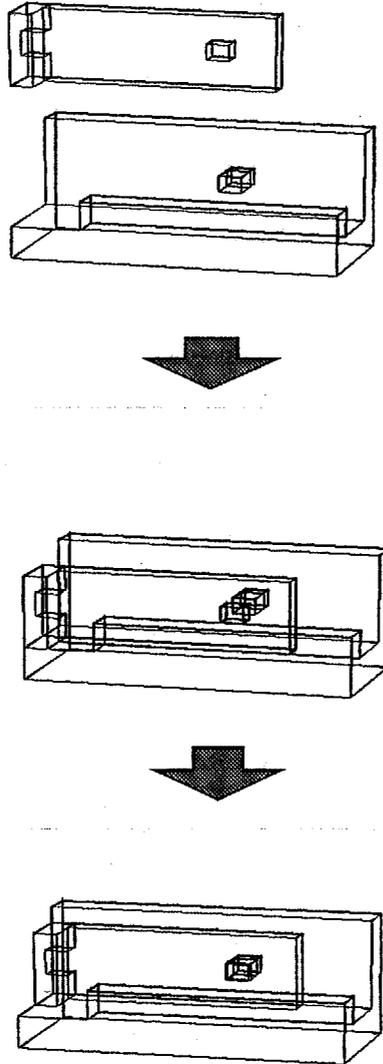


Fig. 5. 13 An assembly path generated by the system

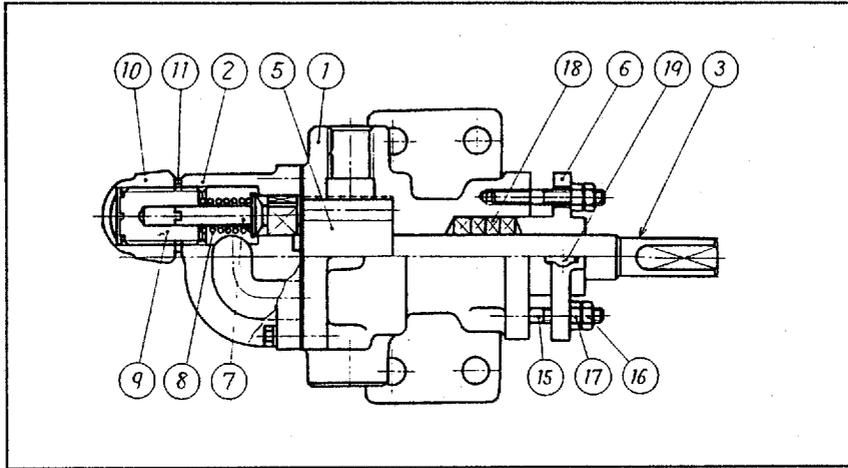


Fig. 5. 14 A design object whose part ⑩, ⑪ are modified

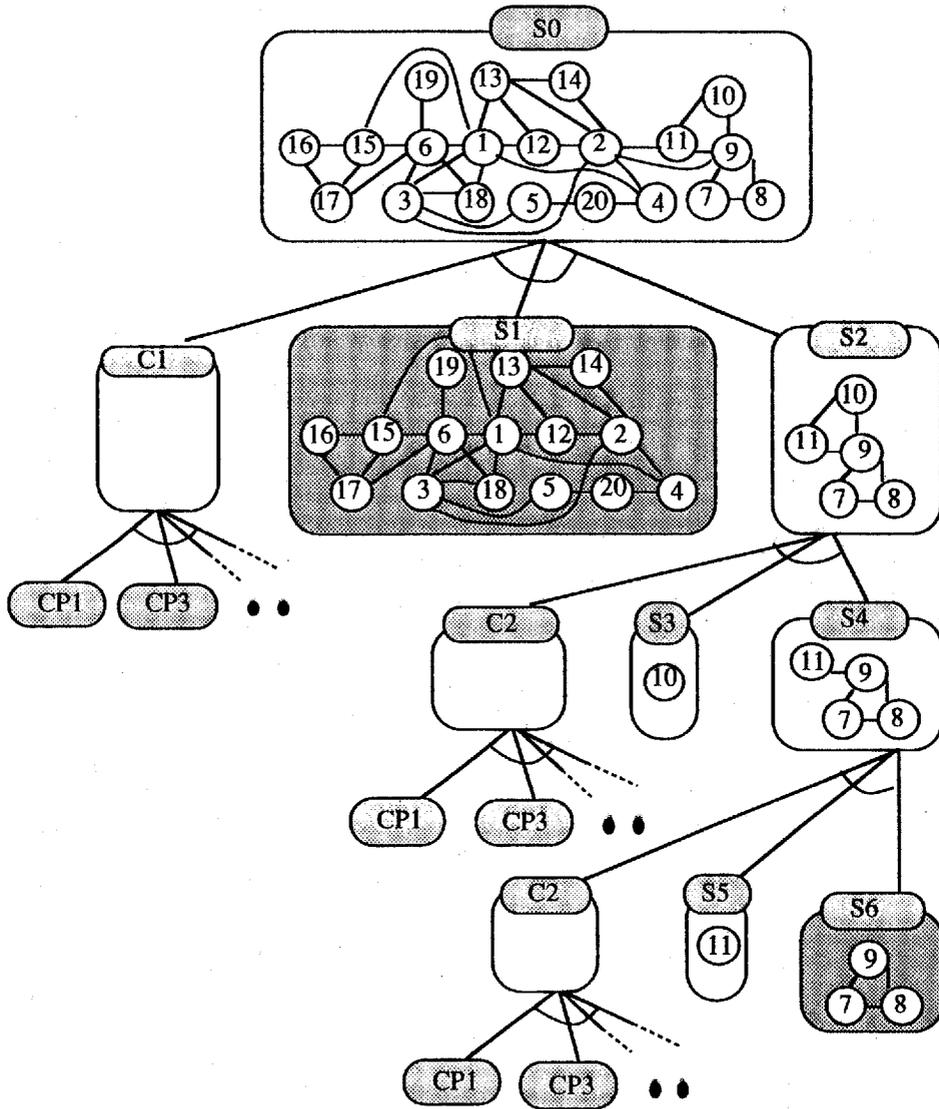


Fig. 5. 15 An example of AND/OR graph search for a modified design object

## 5. 5 結 言

本章で得られた結論を以下に示す。

- (1) 組立順序を生成する問題の部分問題への分割がAND/ORグラフで表せ、これを探索することにより組立順序が得られることを示した。また、この探索にリアルタイム設計処理の概念を導入することにより、設計変更によって生じた部分問題のみを処理するだけで効率よく組立順序が得られることを示した。
- (2) フィーチャに関する情報とソリッドモデルを用いて部分問題を処理する方法を提案した。
- (3) 構成部品数の多い設計対象に対しては、AND/ORグラフの探索に多くの時間を要するため、部品間の接続関係間の優先関係を用いることにより探索空間を縮小する方法を示した。
- (4) 縮小された探索空間内で、コストを最小化する組立順序を求める探索アルゴリズムを提案した。
- (5) CADシステムと統合された組立順序生成システムを開発し、その実行例により本方法の有効性を示した。

## 第6章 結 論

コンカレント・エンジニアリングは製品設計と同時に製造性、安全性などの特性を考慮に入れるものであり、これを実現するため、多くの場合、設計者と専門技術者との協調作業が行われる。これに対し、本論文は、設計者自身に製品のモデリング作業時に様々な特性を考慮に入れさせる設計支援システムの実現を目的とした。本論文は、これを実現するための方法とその適用についてまとめたものである。

各章で、得られた結論を以下に要約する。

第2章では、製品のモデリング作業と統合化作業とを並行処理することにより、設計者を知的に支援する方法を提案した。まず、設計者のモデリング作業を設計対象の特性の観点から知的に支援するためにシステムが満足すべき要件を明かにし、これを満足させるためにはモデリング作業と統合化作業との並行処理が必要であることを示した。次に、この並行処理を実現するため、統合化作業を部分問題に分解し、モデリング作業と並行して処理可能となった部分問題を順次処理する方法を提案した。さらに、この並行処理に基づく設計支援方法において、モデリング作業中の設計者にフィードバックすべき情報として、設計対象の不具合に関する情報と設計変更による影響に関する情報があることを示した。この方法の利点をまとめると以下のようなになる。

- (1) 設計者は、モデリング作業の初期段階から設計対象の不具合を発見できるため、不具合に対する対策を早期に施すことができる。
- (2) 設計者が設計変更を行った場合、それが様々な特性に及ぼす影響を把握できる。
- (3) 設計時間が短縮される。

第3章では、オブジェクト指向に基づく設計対象の新しい表現方法を提案し、その方法に基づいて開発したCADシステムについて述べた。まず、機械の構造を実体と関連からなるグラフとして捉え、実体、関連をそれぞれオブジェクトとして表すことにより、摩擦係数のような複数の実体と関わる属性などを記述できることを示した。また、この実体や関連に記述される情報は、機械の動的な挙動により変化する可能性があることを示し、この挙動をベトリネットで表し、これも実体や関連と分離したオブジェクトとして扱う方法を示した。これにより、挙動を先にモデル化し、そのモデルを基に、その挙動を発現する機械構造を決定することができ、また、実体や関連の種類の変化を引き起こす挙動も表現することが可能となる。開発したCADシステムを用いることにより、構造が変化する設計対象のモデリングが容易にでき、また、モデルを用いた挙動シミュレーションにより、設計対象の挙動を確認できることを示した。

第4章では、第2章で提案したリアルタイム設計処理を故障シミュレーションに適用した。まず、第3章で述べたCADシステムにより構築される設計対象モデルを用いた故障シミュレーションの方法を提案した。この方法は以下の特徴をもつ。

- (1) 実体だけでなく関連に対しても起こりうる故障とその影響を調べるため、部品間の固着のような故障も扱うことができる。
- (2) 挙動シミュレーションによって状態変化した設計対象に対しても、故障シミュレーションを実行するため、設計対象のとりうる各状態に対応した故障の

伝播を把握できる。

次に、この故障シミュレーションとモデリング作業との並行処理を実現するため、故障シミュレーションを部分問題に分解し、モデリング作業中のモデルに対する操作と連動して部分問題を処理する方法を示した。そして、CADシステムと連動するリアルタイム故障シミュレータを試作し、これを使用して、モデリング作業中に故障の定性的な伝播の様子を把握、および設計変更による影響の把握など、が可能であることを示した。

第5章では、組立順序の生成作業を対象に、リアルタイム設計処理による並行処理が不可能でもその適用が有効であることを示した。まず、組立順序を生成する問題の部分問題への分割がAND/ORグラフで表せ、これを探索することにより組立順序が得られることを示した。また、この探索にリアルタイム設計処理の概念を導入することにより、設計変更によって生じた部分問題のみを処理するだけで効率よく組立順序が得られることを示した。また、CADシステムによって作成されるフィーチャに関する情報とソリッドモデルを用いて部分問題を処理する方法を提案した。さらに、構成部品数の多い組立品に対しては、組立順序の探索に多くの時間を要するため、ヒューリスティクスを用いることにより探索空間を縮小し、縮小された探索空間内で、コストを最小化する組立順序を求める探索アルゴリズムを提案した。CADシステムと統合された組立順序生成システムを開発し、その実行例により本方法の有効性を示した。

以上のように、本論文は設計者を知的にサポートする設計支援の新しい方法を提案し、その有効性を示した。なお、今後の課題として以下のものが挙げられる。

- (1) 本方法論の適用可能な範囲を理論的に明かにする必要がある。
- (2) 統合化作業を部分問題に分解する場合、適切な分解を決定する方法を検討する必要がある。

- (3) 部分問題間で矛盾やデッドロック等が起こる場合、その解消方法を検討する必要がある。
- (4) モデリング作業中に不具合が発見された場合、適切なアドバイスを導出する方法を検討する必要がある。
- (5) 複数の特性を同時に調べる場合、特性間のトレードオフを処理する方法を検討する必要がある。

## 謝 辞

本研究を行うに当たり、適切な御指導と懇切なる御鞭撻を頂いた広島大学工学部 大場史憲教授に深く感謝致します。また、本研究を遂行するに当たり有益な御示唆と御助言を賜り、本論文の執筆に際しても綿密な校閲を頂きました広島大学工学部 橋本雅文助教授に心から感謝致します。

本論文をまとめるに当たり、査読をしていただき、適切な御助言を頂きました広島大学工学部 米澤洋教授、中村信人教授に心からお礼を申し上げます。

平素より温かい励ましの御言葉と有益な御助言を頂いた大阪府立大学工学部 小野敏郎教授、大阪工業大学工学部 加藤清助教授、宮崎大学工学部 高庄幸孝助手に感謝致します。また、本研究を遂行するに当たり貴重な資料を提供して頂いた神戸大学工学部 杉村延広助教授に感謝致します。

さらに、本研究に協力してくれた広島大学工学部機械設計工学講座生産システム研究室の学生、研究生の皆様に感謝します。

最後に、本論文を執筆するに当たり、協力し、また、平素より温かい励ましの言葉をかけてくれた我が妻、千恵に感謝します。

## 参 考 文 献

- (1) 瀬口, 尾田, 室津 (編) : 機械設計工学2 [システムと設計], 培風館, (1988).
- (2) 日本機械学会 (編) : CAEとCAM, 技報堂出版, (1988).
- (3) 吉川, 富山 (編) : インテリジェントCAD (上) - 理念とパラダイム -, 朝倉出版, (1989).
- (4) 吉川, 伊藤 (編) : コンピュートロール25 特集/インテリジェントCAD, コロナ社, (1989).
- (5) 赤木 : AI技術によるシステム設計論, コロナ社, (1988).
- (6) 赤木, 藤田 : 設計エキスパートシステムの基礎と応用, コロナ社, (1990).
- (7) 吉川, 富山 (編) : インテリジェントCAD (下) - テクノロジと展望 -, 朝倉出版, (1991).
- (8) H. Yoshikawa and D. Gossard (Eds.) : Intelligent CAD, I, North-Holland, (1989).
- (9) H. Yoshikawa and T. Holden (Eds.) : Intelligent CAD, II, North-Holland, (1990).
- (10) 木村 : CAD/CAM統合化のためのプロダクトモデリングと情報処理, 日本機械学会誌, Vol. 89, No. 10, pp. 1157-1163, (1986).
- (11) 岸 : 知識依存型概念設計システムの研究 - 機械構造の知識表現 -, 精密工学会誌, Vol. 51, No. 11, pp. 83-89, (1985).
- (12) 藤田, 赤木, 辻本 : オブジェクト指向モデリングを用いたリンク機構の最適形状設計支援システムの構築, 日本機械学会論文集 (C編), Vol. 58, No. 545, pp. 310-315, (1992).
- (13) 藤田, 赤木 : 機能設計における設計対象のモデリングと形状モデルとの融合, 日本機械学会論文集 (C編), Vol. 57, No. 535, pp. 412-419, (1991).
- (14) 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一, 日本機械学会論文集 (C編), Vol. 54, No. 505, pp. 2300-2305, (1988).
- (15) 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, pp. 228-233, (1988).
- (16) 赤木, 鷹津: オブジェクト指向言語による配管設計支援システムの研究, 日本機械学会論文集 (C編), Vol. 54, No. 499, pp. 787-791, (1988).
- (17) 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, pp. 1017-1024, (1988).
- (18) 赤木, 鷹津: オブジェクト指向言語による配管設計支援システムの研究 (第2報), 日本機械学会論文集 (C編), Vol. 55, No. 520, pp. 3087-3092, (1989).
- (19) S. Mittal and M. Morjaria : PRIDE : An Expert System for the Design of Paper Handling System, IEEE Computer, pp.102-114, (1986).
- (20) D. C. Brown and B. C. Karen : Expert Systems for a Class of Mechanical Design Activity, Knowledge Engineering in Computer Aided Design, pp.259-264, (1985).
- (21) H. Schmekel : Functional Models and Design Solutions, Annals of the CIRP, Vol. 38, No. 1, pp. 129-132, (1989).
- (22) P. Veerkamp, V. Akman, P. Bernus, and P. Hagen : IDDL:A Language for Intelligent Interactive Integrated CAD Systems, Intelligent CAD Systems 2, Springer-Verlag, pp. 58-74, (1988).
- (23) 吉川: 一般設計学, 人工知能学会誌, Vol.2, No.3, pp.17-23, (1987).
- (24) P. N. Suh : An Axiomatic Approach to Manufacturing System, Trans. of ASME, Vol. 100, No. 5, pp.127-131, (1978).
- (25) 岩田 (監修) : コンピュータ設計・製図 I - CAD/CAMの基礎 -, 共立出版, (1987).
- (26) E. G. Schlechtendahl (ed.) : Specification of a CAD\*I Neutral File for CAD Geometry, Wireframes, Surfaces, Solids, Version 3.2, Revised and Englanged Edition,

Springer-Verlag, (1987).

- (27) 日本コンピュータ・グラフィックス協会：「実現へ向けて第一歩を踏み出したCADデータ交換国際規格STEP」セミナーテキスト，(1989).
- (28) I. L. M. Ricarte : MOODS : A Modular Object Oriented Design Database System, Proc. of CAPE' 89, pp.225-229, (1989).
- (29) T. Kiriyama, T. Tomiyama and H. Yoshikawa : A Model Integration Framework for Cooperative Design, Proc. of MIT-JSME Workshop on Cooperative Product Development, pp.411-441, (1989).
- (30) 末次，大久保（監訳）：コンカレント・エンジニアリング，日本能率協会マネジメントセンター，(1992).
- (31) D.Sriram, R.Logcher, and S.Fukuda (Eds.) : Computer-Aided Cooperative Product Development, Springer-Verlag, (1991).
- (32) N.H.Chao and S.C.-Y.Lu (Eds.) : Concurrent Product and Process Design, ASME DE-Vol.21, (1989).
- (33) E.J.Haug (Ed.) : Concurrent Engineering of Mechanical Systems, ASME DE-Vol.22, (1990).
- (34) 福田：CONCURRENT ENGINEERING の動向，第4回D&Cセミナー「設計とCADの未来像」資料集，pp.89-101, (1990).
- (35) W. Kuo and J. P. Hsu : Update : Simultaneous Engineering Design in Japan, IE, No.10, pp.23-26, (1990).
- (36) M. Sapossnek, S. Talukdar, A. Elfes, S. Sedas, M. Eisenberger, and L. Hou : Design Critics in the Computer-Aided Simultaneous Engineering (CASE) Project, Concurrent Product and Process Design, DE-Vol.21, ASME, pp.137-141, (1989).
- (37) S. Subramanyam and S. C. -Y. Lu : Computer-Aided Simultaneous Engineering for Components Manufactured in Small and Medium Lot Sizes, Concurrent Product and Process Design, DE-Vol.21, ASME, pp.175-184, (1989).
- (38) F. Kimura and H. Suzuki : A CAD System for Efficient Product Design Based on

- Design Intent, Annals of the CIRP, Vol. 38, No. 1, pp. 149-152, (1989).
- (39) G. Boothroyd : Design for Assembly and Design for Manufacture, Annals of the CIRP, Vol. 36, No. 2, pp. 435-437, (1987).
- (40) J. Farris and W. A. Knight : Design for Manufacture : Expert Processing Sequence Selection for Early Product Design, Annals of the CIRP, Vol. 41, No. 1, pp. 481-484, (1992).
- (41) W. A. Knight : Design for Manufacture Analysis : Early Estimates of Tool Costs for Sintered Parts, Annals of the CIRP, Vol. 40, No. 1, pp. 131-134, (1991).
- (42) E. Molloy, H. Yang, and J. Browne : Design for Assembly within Concurrent Engineering, Annals of the CIRP, Vol. 40, No. 1, pp. 107-110, (1991).
- (43) H. J. Warnecke and R. Babler : Design for Assembly - Part of the Design Process, Annals of the CIRP, Vol. 37, No. 1, pp. 1-4, (1988).
- (44) J. Peters and D. Van Campenhout : Manufacturing Oriented and Functional Design, Annals of the CIRP, Vol. 37, No. 1, pp. 153-156, (1988).
- (45) N. P. Suh : Basic Concepts in Design for Producibility, Annals of the CIRP, Vol. 37, No. 2, pp. 559-566, (1988).
- (46) J. Corbett : Design for Economic Manufacture, Annals of the CIRP, Vol. 35, No. 1, pp. 93-97, (1986).
- (47) C. C. Hayes, S. Desa and P. K. Wright : Using Process Planning Knowledge to Make Design Suggestions Concurrently, Concurrent Product and Process Design, DE-Vol.21, ASME, pp.213-218, (1989).
- (48) A. Makino, P. Barkan, L. Reynolds and E. Pfaff : A Design-for-Serviceability Expert System, Concurrent Product and Process Design, DE-Vol.21, ASME, pp.213-218, (1989).
- (49) 伊藤, 曾布川 : 機械のアセンブリ計画の自動生成 (第2報) - システムの実現 -, 精密工学会誌, Vol. 57, No. 1, pp. 104-109, (1991).
- (50) 伊藤, 曾布川 : 機械のアセンブリ計画の自動生成 (第1報) - 問題の定

- 式化一，精密工学会誌，Vol. 56，No. 7，pp. 96-101，(1990).
- (51) 北嶋，西山，吉川：機械の有向接続グラフに基づく対話型組立可能性判定システムの開発，精密機械，Vol. 49，No. 2，pp.66-73，(1983).
- (52) J.Browne, K. Tierney and M. Walsh : A Two-Stage Assembly Process Planning Tool for Robot-Based Flexible Assembly Systems, Int. J. Production Research, Vol. 29, No. 2, pp. 247-266, (1991).
- (53) 宮川：生産性と製品の機能，精密工学会誌，Vol. 56，No. 6，pp. 26-30，(1990).
- (54) 山際：組立性評価法DAC－最適生産システムの実現は商品設計から－，オートメーション，Vol.35，No. 1，pp. 27-32，(1991).
- (55) A. Gairola : Design Analysis for Automatic Assembly, Int. J. Production Research, Vol. 24, No. 4, pp. 839-849, (1986).
- (56) L. S. H. de Mello : A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences, IEEE Transactions on Robotics and Automation, Vol. 7, No. 2, pp. 228-240, (1991).
- (57) D. F. Baldwin, T. E. Abell, M. M. Lui, T. L. De Fazio, and D. E. Whitney : An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products, IEEE Transactions on Robotics and Automation, Vol. 7, No. 1, pp. 78-94, (1991).
- (58) A. J. Scarr : Product Design for Automated Manufacture and Assembly, Annals of the CIRP, Vol. 35, No. 1, pp. 1-5, (1986).
- (59) S. Miyakawa, T. Ohashi, S. Inoshita, and T. Shigemura : The Hitachi Assemblability Evaluation Method (AEM) and Its Applications, Proc. of MSET21, pp. 277-282, (1990).
- (60) D. S. Domazet and S. C. Y. Lu : Concurrent Design and Process Planning of Rotational Parts, Annals of the CIRP, Vol. 41, No. 1, pp. 181-184, (1992).
- (61) S. E. Lander, V. R. Lesser, and M. E. Connell : Knowledge-Based Conflict

Resolution for Cooperation Among Expert Agents, Proc. of MIT-JSME Workshop on Cooperative Product Development, pp.253-268, (1989).

(62) K. P. Sycara : Cooperative Negotiation in Concurrent Engineering Design, Proc. of MIT-JSME Workshop on Cooperative Product Development, pp.269-297, (1989).

(63) 吉村, 井谷, 人見 : 機械製品に対する製品設計と工程設計の統合的最適化に関する研究, 日本機械学会論文集 (C編), Vol.54, No. 505, pp. 2313-2320, (1988).

(64) M. J. Jakiela : Intelligent Suggestive CAD Systems : Research Overview, Proc. of MIT-JSME Workshop on Cooperative Product Development, pp.411-441, (1989).

(65) 畑村 : 実際の設計, 第9回設計シンポジウム講演資料, (1991).

(66) 小野里, 吉川 : 設計における機械構造表現法の研究, 昭和61年度精密工学会秋季大会論文集, pp.439-440, (1986).

(67) 酒井, 堀内 : オブジェクト指向入門, オーム社, (1989).

(68) M. J. Combs (ed.) : Development in Expert Systems, Academic Press, p. 23, (1984).

(69) 鈴木, 木村 : プロダクトモデルに基づく幾何学的拘束関係の記述と寸法処理への応用, 精密工学会誌, Vol.52, No.6, (1986).

(70) T. Tomiyama : Object Oriented Programming Paradigm for Intelligent CAD Systems, Intelligent CAD Systems 2, Springer-Verlag, pp. 3-16, (1988).

(71) Z. Ruttkay : Multi-Media Presentation in CAD Systems, Intelligent CAD Systems 2, Springer-Verlag, pp. 77-92, (1988).

(72) R. E. Phillips and L. W. Rosenfeld : A Knowledge-Based System for Design Automation, Proceedings of 6th International Symposium on Offshore Mechanics and Artic Engineering, (1987).

(73) L. Peters (白井, 奥島 訳) , 実践的構造化分析と設計, 近代科学社, p. 54, (1990).

(74) 長澤, 手越 : 建築物実設計のためのスケッチパッド, 昭和63年度精密工

学会春季大会論文集, pp.435-436, (1988).

(75) J. L. Peterson : Petri Net Theory and the Modeling of the Systems, Prentice Hall, (1981).

(76) Jr. A. M. Johson, and M. Malek : Surveys of Software Tools for Evaluating Reliability, Availability, and Serviceability, ACM Computing Surveys, Vol. 20, No. 4, pp. 227-240, (1988).

(77) 佐山：プロセスシステムのフォールト・ツリー作成法に関する研究（第1報）－基本アルゴリズムとその応用－, 安全工学, Vol. 20, No. 5, pp. 286-293, (1981).

(78) 佐山：プロセスシステムのフォールト・ツリー作成法に関する研究（第2報）－2個の離散値を有する変数－, 安全工学, Vol.21, No.1, pp.17-22, (1982).

(79) 佐山, 鈴木：プロセスシステムのフォールト・ツリー作成法に関する研究（第3報）－フィードバックとフィードフォワード制御回路結合系－, 安全工学, Vol. 21, No. 3, pp. 148-154, (1982).

(80) 佐山, 鈴木：プロセスシステムのフォールト・ツリー作成法に関する研究(IV)－制御回路と保護計装回路（しゃ断回路, 安全弁）の結合系－, 安全工学, Vol. 22, No. 2, pp. 96-102, (1983).

(81) 鈴木, 亀山, 佐山：プロセスシステムのフォールト・ツリー作成法に関する研究(V)－複雑な構造を有するシステムのフォールト・ツリー作成－, 安全工学, Vol. 22, No. 4, pp. 204-212, (1983).

(82) 鈴木, 亀山, 佐山：オペラビリティ・スタディと知識工学的手法によるフォールト・ツリー自動生成法, 安全工学, Vol. 28, No. 4, pp. 202-210, (1989).

(83) 鈴木, 佐山, 亀山：デシジョンテーブルと知識工学によるフォールト・ツリー自動生成法, 安全工学, Vol. 28, No. 5, pp. 291-303, (1989).

(84) S. H. Rich, and V. Venkatasubramanian : Model-based Reasoning in Diagnostic Expert Systems for Chemical Process Plants, Comput. Chem. Engng., Vol. 11, No. 2, pp. 111-115, (1987).

- (85) 熊本, 山品, 坂本 : AND/ORグラフの再帰的探索に基づくフォールトツリーの自動生成, 計測自動制御学会「あいまい情報処理と知的システム制御」講演論文集, pp. 91-94, (1988).
- (86) G. J. Powers, and F. C. Thompkins : Fault Tree Synthesis for Chemical Processes, *AIChE J.*, Vol. 20, No. 2, pp. 376-382, (1974).
- (87) S. A. Lapp, and G. J. Powers, : Computer-Aided Synthesis of Fault Trees, *IEEE Trans. Reliability*, R-26, Dec., pp. 316-321, (1977).
- (88) S. L. Salem, G. E. Apostolakis, and D. Okrent, : A New Methodology for Computer-Aided Construction of Fault Trees, *Annals of Nuclear Energy*, No. 4, pp. 417-422, (1977).
- (89) 荻野, 宇田川, 近藤, 芝 : 設計と試験を統合的に支援する知的CAD/CATシステム, 人工知能学会, Vol.5, No.4, pp.492-501, (1992).
- (90) 熊本, 山品, 坂本 : 再帰的ルールベース・システムによるフォールトツリーの自動生成, 計測自動制御学会論文集, Vol.26, No.3, pp.334-351, (1990).
- (91) 大森, 上野 : 対象モデルによるハイブリッド型故障診断システムーモデル表現と推論ー, 人工知能学会誌, Vol. 5, No. 5, pp. 604-615, (1990).
- (92) 山口, 溝口, 中村, 小澤, 鳥越, 野村, 角所 : 対象モデルと故障モデルに基づく知識コンパイラIIの構築と評価, 人工知能学会誌, Vol. 7, No. 4, pp. 663-674, (1992).
- (93) 山口, 溝口, 田岡, 小高, 野村, 角所 : 深い知識に基づく知識コンパイラの基本設計, 人工知能学会誌, Vol. 2, No. 3, pp. 333-340, (1987).
- (94) 梅田, 小池, 佐藤, 富山, 吉川 : 対象モデルに基づく定性物理を用いた故障診断, 精密工学会誌, Vol. 58, No. 4, pp. 154-160, (1992).
- (95) 村上, 中村, 中島 : フィーチャー・ディスクリプションに基づく設計診断に関する研究 (方程式生成による平面リンク機構の自動機能推定), 日本機械学会論文集 (C編), Vol. 53, No. 490, pp.1278-1282, (1987).
- (96) 三根 (訳) : 電子機器・部品の信頼度予測 (MIL-HDBK-217B), 総合電子

- 出版社, (1973).
- (97) E. Green : Reliability Technology, John Willy.
- (98) 山田, 阿部, 辻 : 電気ドリル分解・組立コンサルタント・システム, 人工知能学会誌, Vol. 1, No. 1, pp. 116-123, (1986).
- (99) 関口, 今村, 小島, 井上 : 回転機能部品の部品展開手法に関する研究(第2報) - 組立・分解手順の生成とその規則化 -, 精密工学会誌, Vol. 53, No. 8, pp. 43-48, (1987).
- (100) 関口, 小島, 井上, 本多 : 回転機能部品の部品展開手法に関する研究 - 組立構造の表現法と組立・分解手順の生成への応用 -, 精密機械, Vol. 51, No. 2, pp. 101-107, (1985).
- (101) G. Dini and M. Santochi : Automated Sequencing and Subassembly Detection in Assembly Planning, Annals of the CIRP, Vol. 41, No. 1, pp. 1-4, (1992).
- (102) L. Laperriere and H. A. ElMaraghy : Planning of Products Assembly and Disassembly, Annals of the CIRP, Vol. 41, No. 1, pp. 5-9, (1992).
- (103) W. H. ElMaraghy : Integrating Assembly Planning and Scheduling - CAPP-Related Issues, Annals of the CIRP, Vol. 41, No. 1 pp. 11-14, (1992).
- (104) H. J. Warnecke : Integration of CAPP and PPC - The PPC Part, Annals of the CIRP, Vol. 41, No. 1, pp. 15-18, (1992).
- (105) H. K. Tonshoff, E. Menzel, and H. S. Park : A Knowledge-Based System for Automated Assembly Planning, Annals of the CIRP, Vol. 41, No. 1, pp. 19-24, (1992).
- (106) C. L. P. Chen : Automatic Assembly Sequences Generation by Pattern Matching, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 2, pp. 376-389, (1991).
- (107) L. S. H. de Mello : Two Criteria for the Selection of Assembly Plans : Maximizing the Flexibility of Sequencing the Assembly Tasks and Minimizing the Assembly Time Through Parallel Execution of Assembly Tasks, IEEE Transactions on

- Robotics and Automation, Vol. 7, No. 5, pp. 626-633, (1991).
- (108) T.C. Woo and D. Dutta : Automatic Disassembly and Total Ordering in Three Dimensions, Trans. ASME, Journal of Engineering for Industry, Vol. 113, pp. 207-213, (1991).
- (109) Y. F. Huang and C. S. G. Lee : A Framework of Knowledge-Based Assembly Planning, Proc. IEEE Robotics and Automation, pp. 599-604, (1991).
- (110) L. S. H. de Mello and A. C. Sanderson : Representations of Mechanical Assembly Sequences, IEEE Transactions on Robotics and Automation, Vol. 7, No. 2, pp. 211-227, (1991).
- (111) C. J. M. Heemskerk and L. N. Reijers : A Concept for Computer-Aided Process Planning of Flexible Assembly, Annals of the CIRP, Vol. 39, No. 1, pp. 25-28, (1990).
- (112) E. Zussman, E. Lenz , and M. Shpitalni : An Approach to the Automatic Assembly Planning Problem, Annals of the CIRP, Vol. 39, No. 1, pp. 33-36, (1990).
- (113) M. Shpitalni, G. Elber, and E. Lenz : Automatic Assembly of Three-Dimensional Structures via Connectivity Graphs, Annals of the CIRP, Vol. 38, No. 1, pp. 25-28, (1989).
- (114) C. J. M. Heemskerk : The Use of Heuristics in Assembly Sequence Planning, Annals of the CIRP, Vol.38, No. 1, pp.37-40, (1989).
- (115) E. Arai, S. Morishita, and K. Iwata : CAD System to Support Product Assembly / Disassembly Planning, Proc. of MSET21, pp.545-550, (1990).
- (116) A. K. Subramani and P. Dewhurst : Automatic Generation of Product Disassembly Sequences, Annals of the CIRP, Vol. 40, No. 1, pp. 115-118, (1991).
- (117) N. J. Nilsson (白井, 辻井, 佐藤訳) : 人工知能の原理, 日本コンピュータ協会, (1983).
- (118) 川辺 (編) : 3次元シミュレーション技術, 工業調査会, pp.179-183, (1988).

(119) 吉川：機械のトポロジ，精密機械，Vol.38, No.12, pp.1018-1023, (1972).

## 学術研究論文一覧

1. T. Murayama, T. Ono, and F. Oba : Integration of CAD and Trouble Shooting System, Proceedings of International Conference on Manufacturing Systems and Environment - Looking Toward the 21st Century, pp.551-556, (1990).
2. T. Murayama and F. Oba : Real Time Design Operation for Integrated CAD System, Proceedings of Japan-U.S.A. Symposium on Flexible Automation, pp.1201-1204, (1990).
3. 村山, 大場 : 製品設計と信頼性解析の統合 (第1報) -リアルタイム故障シミュレーション-, 精密工学会誌, 第57巻, 第4号, pp.725-731, (1991).
4. T. Murayama and F. Oba : Design Aid Based on Concurrent Processing of Design Tasks, IFIP Transactions B-3, pp.503-513, (1992).
5. T. Murayama and F. Oba : Transformable Object Model for a CAD System, Proceedings of Japan-U.S.A. Symposium on Flexible Automation, pp.995-998, (1992).
6. T. Murayama and F. Oba : Concurrent Processing of Product Modeling and Fault Simulation for Intelligent Design Aid, Proceedings of IMACS / SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems '92 Kobe, pp.1525-1530, (1992).
7. T. Murayama and F. Oba : Concurrent Engineering Approach for Assembly Planning in Computer Aided Design : A Method for Generating Assembly / Disassembly Sequence, Proceedings of Pacific Conference on Manufacturing, pp.107-113, (1992).