

装置トラブルに対する工程の確保を意識している。また、生産現場の状況としてボトルネックが前方にある場合と後方にある場合、あるいは前後にある場合なども考慮に入れたものとなっている。

実験の結果、初期スケジュールの持つペナルティを、平均で約34%まで抑えることができた(表5.4)。また、計算停止までの反復回数は平均で1.6回(表5.5)であり、これは本アルゴリズムの効率のよさを示している。さらに、本アルゴリズムを他の探索法と組み合わせた場合の解の改善幅は高々3ポイント(表5.6)であり、この数字は、特にスケジュールの最適性を重視しない状況では、本アルゴリズムを山登り法として利用しても十分に実用性があることを示している。

対象スケジュールを、事象と制約によって表現する本研究のアプローチは、オペレーションの処理時間、オペレーションの先行関係、オペレーション間の間隔、オペレーション同時開始や同時終了、オペレーションの連続処理、そして最早開始時刻や納期など、生産現場における非常に多くの種類の制約を表現することを可能にした。また、実際にスケジューリングシステムを利用する場合、これらの制約をコンピュータに入力する手間が問題となるが、この問題も、グラフィカル・ユーザ・インタフェースにより、これらのさまざまな制約表現と本研究で用いる内部表現の間の自動変換を行うことで、十分に対応可能であることを確認した。

5.6 5章のまとめ

本章では、問題解決の枠組みの中の制約処理の部分に注目し、特に制約充足処理の研究対象として取り組んだ研究内容について紹介した。従来の制約充足問題に対する研究の多くは、組合せ探索問題として厳密計算を要求されるため現実的な問題に対する解を得ることが難しかった。本研究で開発したアルゴリズムは、制約の緩和を前提とした準最適解を求めるためのものであり、ペナルティ伝播グラフを用いて違反制約のペナルティを他の制約に伝播させることにより、徐々にペナルティの合計値を減少させていくというヒューリスティックな手法である。本章では、このアルゴリズムをスケジューリングにおける複雑な制約関係に適用した例を、実際の化学プラントの操業データをもとに行なった実験結果を交えて紹介した。

第6章

設計と計画の統合問題

6.1 はじめに

問題構造の空間的な側面を主に決定する問題は設計問題、時間的な側面を主に決定する問題は計画問題に相当する。従来は、空間的な構造の決定と時間的な構造の決定は、個別に行なわれるのが普通であったが、本研究ではそれらをできるかぎり同時並行的に行なう。本研究で提案するコンカレントな問題解決では、空間的な構造はリレーションに設定された制約関係をもとに計算し、時間的な構造は、オペレータに設定された因果関係をもとに計算する。これにより、これらの種類の異なる2つの問題を、同一の枠組の中で扱うことを可能にし、個別に行なうよりも効率的でかつ柔軟な推論を実現している。

本章では、このような空間的構造と時間的構造の同時並行的な決定のためのアルゴリズムを紹介し、簡単な例題によってその有効性を示す。まず次節では、本研究における時間の概念について説明する。続いて3節で、処理の概要として、空間的な構造と時間的な構造の扱いにおいていくつかの注意すべき点を説明する。続く4節では、コンカレントな問題解決の中心的アルゴリズムの1つであるAND-ORツリーの展開と、関連するロジックについて説明する。5節では、ここで説明した内容を例題を用いて解説し、最後に6節で、アルゴリズムの評価について述べる。

6.2 時間の概念について

時間を扱う研究は数多い。そもそも時間とは何かというテーマは哲学的な大問題であり、依然として明確な答えは得られていない。ここでは、本研究における時間の概念をあらかじめ説明する。

従来の研究の中で“時間”という概念は、非常に便宜的に用いられているが、それ

らを非常に大ざっぱに分類すると、“静的な時間”と“動的な時間”の2つに分けられる。前者は宣言的な時間概念、後者は手続き的な時間概念という言い方もできる。ここで静的と動的の違いは、対応とする現実世界の対象の不可逆的な変化を、その概念が表現しているかどうかにある。つまり、静的な時間を扱って問題解決を行なう場合には、問題解決の過程とそこであつかう時間概念とは独立しているが、動的な時間概念を扱う場合には、問題解決という行為を通して、時間概念そのものが生み出されていく。

例えば、時間をパラメータとした運動方程式などは、静的な時間概念の典型である。また、スケジューリング問題における時間概念も、時間をパラメータとして表現し、その関係を制約式によって定義しているような場合は、静的な時間概念に分類される。一方、プランニング問題 [Fikes71] における手順は動的な時間概念に相当する。なぜなら、手順という時間的な関係は、問題解決の過程によって生み出されるものであり、さらに手順には方向性がありその方向にそって状態遷移を起こすからである。このような分類からすると、人工知能の研究における、状況論理 [McCarthy80] は動的な時間概念を扱っており、非単調推論や時間推論 [Allen84], [McDermott82], [Shoham87] などはその代表的なものである。

現実世界における変化の大半は不可逆であり、これらの因果関係をより忠実に表現しようとする、動的な時間概念が必要となる。しかし、現在のところ動的な時間概念の表現形式は、離散化された時間しか表現できない点や、要求する結果を導くプランを得るのに莫大な計算コストを必要とするなど、欠点も多い。従って、本研究では、基本的に動的な時間概念を中心として問題解決の枠組みを定めるが、静的な時間も同時に扱う方針をとる。本研究では、時間概念はオペレータによって表現されるが、ここで静的な時間はオペレータの時刻パラメータが対応し、動的な時間は、オペレータの持つ因果関係によって起こる状態遷移、あるいは先行関係といったものが対応する。

オペレータの因果関係によって表現されていた動的な時間概念を、もし静的な時間概念に置きかえる必要があれば、オペレータの時刻を項とする制約式をもつリレーションを新たに定義すればよい。しかし、この結果、表現された対象世界の時間的な関係は、固定化されたものとなる。従って、問題明確化にあたっては、対象の持つ時間的な構造をどちらの形式で表現するかは、その都度人間が判断する必要がある。

6.3 処理の概要

問題構造は、エンティティ、プロセス、リレーション、そしてオペレータの4つのプリミティブによって表現されるが、計算機が行なう制約処理では、リレーションと

オペレータのみをノードとする R-O グラフが対象となる。従って、制約処理では、まず問題構造をこの R-O グラフの形式に変換し、その上で制約充足計算と状態遷移計算を行なう。

R-O グラフの生成

各リレーションはいくつかの制約式を持ち、その制約式の各項には他のリレーションのパラメータあるいはオペレータの時刻が設定されている。また、オペレータには、条件部と行動部にリレーションが設定されている。これらの関係をノードとリンクで表現したものが R-O グラフである。なお、これらの関係は、問題解決において、個々の問題構造として各プリミティブが設定され、それらのプリミティブ間のコネクションが与えられた時点で初めて確定する。

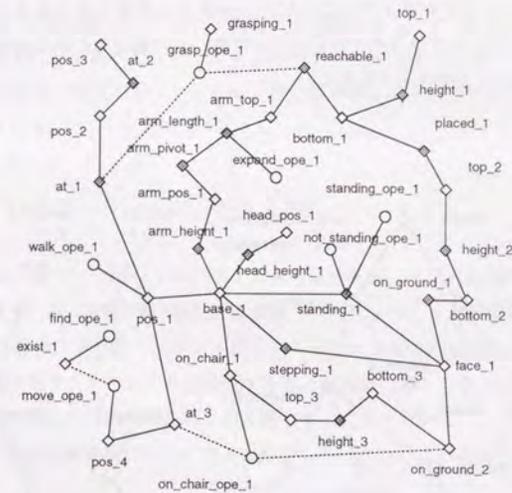


図 6.1: 花子さんの問題の R-O グラフ

図 6.1 は、図 3.4 に示した花子さんの問題の R-O グラフである。ここで、◇マークはリレーションを、○マークはオペレータを表している。制約式を持つリレーションについては、特に斜線をつけて表示している。例えば、花子さんが花瓶を掴むための前提条件となっている `reachable_1` のクラス `reachable` には、あらかじめ

$$entity@bottom \leq arm@arm_top$$

という制約式が設定されており、entity@bottom と arm@arm_top という2つのリレーションのクラスを制約式の項としてもつ。ただし、これらの項が実際に pot.l@bottom.l および arm.l@arm_top.l というインスタンスに対応するのは、計算機上でそれらのプリミティブとコネクションが設定された以降となる。

このためには、まず、reachable.l とコネクションを持つ pot.l, arm.l のなかから、entity と arm の下位クラスであるプリミティブが検索される。ここでは、entity に pot.l が、arm に arm.l が割り当てられる。続いて、pot.l, arm.l の持つプロパティの中から、それぞれ bottom および arm_top というクラスのインスタンスを探索する。この結果、bottom には bottom.l が、arm_top には arm_top.l が割り当てられる。

もし、プロパティの中に該当するインスタンスがない場合には、そのプリミティブをメンバーとして持つ上位のプリミティブについて探索する。例えば、at の制約には entity@pos という項が設定されているが、at.l とコネクションをもつ arm.l には pos というクラスのプロパティをもたないため、その親にあたる human.l のプロパティである pos.l が割り当てられる。

変更要求の展開

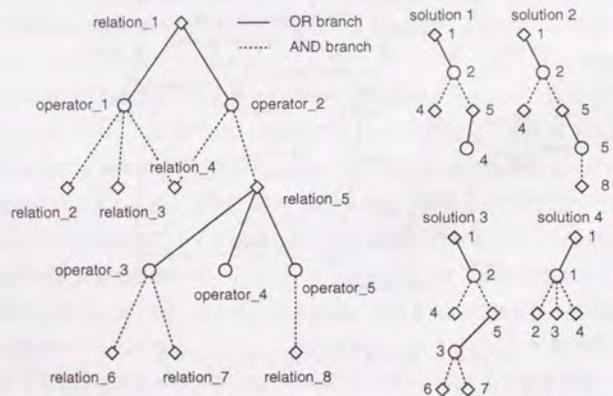


図 6.2: 変更要求の展開

制約充足計算において、パラメータの変更要求が発行されるか、あるいはユーザが直接パラメータを変更を要求した場合、状態遷移計算が起動される。変更要求の対象

となるパラメータを変更するためにはオペレータが適用されるが、オペレータが適用されるためには、そのオペレータの条件部にある複数のリレーションの持つ制約式が満たされていないなければならない。このため、このパラメータ変更処理は、AND-OR ツリーの探索問題となる。なぜなら、条件部にあるリレーションのいずれかが成立していない場合、そのリレーションを成立させるようなオペレータの適用を新たに検討する必要があるからである。図 6.2 に変更要求の展開の際の探索の様子を示す。このように、変更要求の展開では、リレーションが変更要求として最上位のノードに設定された後、リレーションを OR ノード、オペレータを AND ノードとして交互に展開される。図 6.2 の場合は、4 つの解候補が生成されることになる。なお、この AND-OR ツリーの生成アルゴリズムは、次節で詳しく説明する。

パラメータの変更方法

オペレータの行動部には、リレーションが1つ設定される。オペレータを適用させると、その行動部に設定されたリレーションのパラメータの値が変更される。変更後のパラメータの値は、関係するリレーションまたはオペレータの値をもとに以下の式に従って決定される。

$$x_k := \sum_{i \in S} c_i x_i \quad (6.1)$$

ここで、 x_i はリレーション i のパラメータの値であり、 k は行動部に設定されたリレーション番号、 S は条件部、行動部に設定されたリレーション番号の集合である。また、 c_i は、関係するリレーション i がオペレータによる値の変更への貢献度を係数で表したものであり、オペレータの変更する値に無関係のリレーションには 0 が設定される。なお、式 6.1 は他の制約式など異なり、代入式である。

パラメータの変更は状況に応じて以下に示すいくつかの方法が選択される。

1. 変更要求として x_k の新しい値がすでに指定してある場合、式 6.1 の関係を満たすように k 以外の S のいずれかのパラメータを変更した後に x_k を変更する。
2. リレーション k を有効化したい場合、式 6.1 に加えて、

$$v_k^{\min} \leq x_k \leq v_k^{\max} \quad (6.2)$$

を満たすように k 以外の S のいずれかのパラメータを変更した後に x_k を変更する。ここで、式 6.5 はリレーション k の有効範囲制約であり、 v_k^{\min} 、 v_k^{\max} はそれぞれパラメータの有効範囲の最小値、最大値である。

3. リレーション k を無効したい場合、式 6.1 に加えて、

$$r_k^{\min} \leq x_k < v_k^{\min} \quad \text{または} \quad v_k^{\max} < x_k \leq r_k^{\max} \quad (6.3)$$

を満たすように k 以外の S のいずれかのパラメータを変更した後に x_k を変更する。ここで、 r_k^{\min} 、 r_k^{\max} はそれぞれリレーション k の定義範囲の最小値、最大値である。この式は、リレーション k の定義範囲制約を満たしかつ有効範囲制約を満たさない状態を表す。

4. 上記以外の場合は、式 6.1 に加えて、定義範囲制約

$$r_k^{\min} \leq x_k \leq r_k^{\max} \quad (6.4)$$

を満たすように k 以外の S のいずれかのパラメータを変更した後に x_k を変更する。

先行関係制約の設定

AND-OR ツリーの展開において、ツリー上でリレーションを介して上下に接続する 2 つのオペレータには、先行関係が計算機によって自動的に設定される。図 6.3 のように、展開の方向を上から下方向とすれば、時間の方向はその逆方向である下から上となる。つまり、リレーションを介して下側にあるオペレータが先行オペレータ、上側にあるオペレータが後続オペレータとなる。

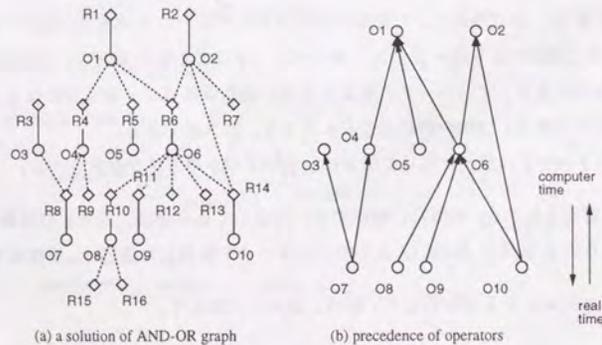


図 6.3: オペレータの半順序関係

先行関係の制約は、先行オペレータに以下の制約式が追加される。

$$t_{suc} - t_{pre} + \sum_{i \in S} d_i x_i = 0 \quad (6.5)$$

ここで、 pre は先行オペレータの番号、 suc は後続オペレータの番号で、AND-OR ツリー上でリレーションを介して上下に位置するすべての組が対応する。また、 t はオペレータのもつ時刻パラメータである。さらに、 d_i は、オペレータの条件部や行動部にあるリレーションが、2 つのオペレータの先行関係に関係する場合に、その割合を係数で表したものであり、無関係の場合には 0 が設定される。

オペレータの先行関係は最終的に、図 6.3(a) の AND-OR ツリーからリレーションを取り除いた図 6.3(b) のようなグラフとなる。このように、AND-OR グラフは、オペレータをノードとする半順序関係である束に変換でき、これが本研究が対象とする時間的な構造の中核となる。

副作用によるオペレータ適用

変更要求を処理するために AND-OR ツリー上のさまざまなリレーションを有効化することによって、意図しないオペレータが起動する場合がある。これは、問題解決過程における副作用といえる。PICSSS では、AND-OR ツリーの解候補が見つかった場合に、その解候補に従ってオペレータの適用を行ないパラメータの値を変更するが、その過程において、未適用のオペレータが適用可能となった場合にそのオペレータも同時に適用している。ただし、後で述べるように、この副作用によって矛盾が発生する場合には、そもその原因となる AND-OR ツリーの候補解を無効とし、それによって発生した副作用はすべて取り消す、といった矛盾の解消を行なっている。

6.4 計算アルゴリズム

以下に、AND-OR ツリーを展開するアルゴリズムを示す。以下のアルゴリズムは、AND ノードと OR ノードが交互に現われる性質を利用したものであり、AND ノードの展開と OR ノードの展開を順に行っている。ここでは、オペレータが AND ノードに、リレーションが OR ノードに対応する。なお、以下のアルゴリズムでは、全解探索は行わず、1 つの解候補を見つけた時点で停止する。

手順 1: 変更要求のリレーションをスタック R に積む

手順 2: スタック R からリレーションを取り出し i とする。もしスタック R とスタック O がともに空の場合は失敗終了。スタック R のみ空の場合は手順 4 へ。

手順 3: リレーション i を行動部にもつオペレータを探索木に追加し、それらのオペレータをスタック O に積む。リレーション i を行動部にもつオペレータが存在しない場合は、リレーション i を削除する。手順 2 へ。

手順 4: スタック O からオペレータを取り出し j とする。もしスタック O とスタック R がともに空の場合は失敗終了。スタック O のみ空の場合は手順 2 へ。

手順 5: オペレータ j の条件部にあるリレーションについて制約充足処理を行う。制約充足不可能な場合は、オペレータ j を削除する。制約充足にあたってパラメータの値を変更する必要がある場合は、そのリレーションを探索木に追加し、それらのリレーションをスタック R に積む。すべての条件リストが制約充足した場合は終端マークをつける。

手順 6: 終端マークのあるオペレータを下位にもつすべてのリレーションに終端マークをつける。また、下位のリレーションがすべて終端マークをもつオペレータにも、同様に終端マークをつける。もしツリーの最上位にあるすべてのリレーションに終端マークがつけられた場合は成功終了。それ以外は、手順 4 へ。

上記のアルゴリズムで、AND-OR ツリー上のリレーションを削除する場合は、その上位にあるオペレータも併せて削除する。また、オペレータを削除する場合は、その上位にあるリレーションがそのオペレータ以外に下位ノードを持たない場合に限りリレーションも併せて削除する。

制約充足計算の内部呼び出し

状態遷移計算において、以下の 2 つの場合にアルゴリズム内部で制約充足計算を呼び出す。なお、呼び出し先の制約充足計算の側で再び状態遷移計算を呼び出すことが必要となる場合があるが、ここではとりあえずそのような再帰的な呼び出しは行わないことにする。

1. オペレータの条件部に指定されたリレーションは、すべて有効でなければならない。通常、リレーションが有効であるということは、「リレーションのパラメータの値が有効範囲内にある」ことを指す場合と、「新たなパラメータ変更要求を発行せずにリレーションの制約充足が可能である」ことを指す場合がある。後者の場合、オペレータの適用可能性を調べる時点で制約充足計算が適宜呼び出される。

2. AND-OR ツリーの展開において、オペレータを新たなノードとして生成するとき、必要に応じて先行関係について制約充足計算を行なう。これによって関係する時刻がその都度計算し直され、常に時間的な関係構造が実現可能な状態に保たれる。

矛盾の正当化处理

本研究の制約処理において、矛盾とはリレーションのパラメータの値がその定義範囲にない状態をいう。定義範囲制約は、そのリレーションの論理的な正当性を保証する制約であり、この制約は緩和することができない（制約充足計算によって、この制約が違反することはない）。従って、オペレータによるパラメータの変更によって、この定義範囲制約が違反するような場合は、以下のような矛盾回避のための処理を行なう必要がある。

矛盾を回避する方法として、2 通りの方法が考えられる。第 1 の方法は、矛盾が発生した場合に、その原因となるリレーションの有効化を取り消し、オペレータが適用される前の状態に戻す方法である。第 2 の方法は、あらかじめ矛盾が発生しないようにその原因となるリレーションの有効化を禁止しておく方法である。

第 1 の方法では、定義範囲制約が違反し矛盾が発生した場合、以下の 2 つの中のいずれかの方法で矛盾を解消する。

- 矛盾するリレーションを r 、そのリレーションのパラメータを変更したオペレータを j 、オペレータ j の条件リストの中で直前までに変更され有効となったリレーション以外の集合を R_j とし、 R_j の中から任意のリレーションを 1 つ無効とする。
- 矛盾するリレーション r に対して、その矛盾を解消するオペレータ i が存在する場合は、そのオペレータ i の条件部にあるリレーションの集合 R_i の中で無効であるものについて有効化要求を発行し、状態遷移計算を行なう。

また、第 2 の方法では、先に説明した AND-OR ツリーの展開の手順に加えて、以下の手順を追加することで、矛盾を回避する。

手順 1: ある状態 t におけるパラメータの値を $\mathbf{p}^t = \{p_1^t, p_2^t, \dots, p_n^t\}$ とし、パラメータ $i \in \{1, \dots, n\}$ を行動部にもつオペレータの集合を O_i とすると、 $p_i^t + D_j$ が定義範囲の外となるようなオペレータ $j \in O_i$ を有効化禁止オペレータとする。

手順 2: 有効化禁止オペレータ j の条件部のリレーションの集合を R_j^{nd} 、その中で無効なリレーションの集合を \bar{R}_j^{nd} とすると、 $|\bar{R}_j^{nd}| = 1$ の場合、その無効なリレーション $i \in \bar{R}_j^{nd}$ を有効化禁止リレーションとする。

手順 3: 有効化禁止リレーションを AND-OR ツリーの展開において有効化する必要が生じた場合には、そのときの有効化禁止オペレータの行動部のリレーションのパラメータの値が変化しないよう、相殺するオペレータを適用させるようなパラメータ変更要求を発行する。

PICCSS では、基本的に第 2 の方法を用いているが、この第 2 の方法では、完全に定義範囲制約の違反による矛盾を回避することはできないため、もしそれでも矛盾が生じた場合は、人間の指示により第 1 の方法によって矛盾を解消する。

非単調性の管理

非単調推論とは、いったん真とされた事象が、推論の経過にともない後になって真偽が変更されるような推論である。プランニングのように、行為によって実世界のモデルを操作するような問題では、一般に非単調推論となる。本研究の制約処理も、リレーションの真偽値が、状態によって変化するため、非単調推論に分類される。

リレーションの真偽を決定するパラメータの値は、状態遷移計算の過程において通常オペレータによって変更されるが、矛盾の回避やユーザの指定によるバックトラックが行なわれた場合には、以前に設定した値が取り消され、過去の状態に戻る。このような場合に、問題構造の中で整合性を保つために以下のような処理が必要となる。

- パラメータの値の復元
リレーションおよびオペレータのパラメータの値を変更前の状態へ戻す。各パラメータは、変更された計算機時刻と値からなる変更履歴を保持しており、その内容に従って値を復元する。
- オペレータ適用の取消
オペレータは適用/未適用の区別を持っており、適用された状態の場合はその計算機時刻が割り当てられている。これをもとに、指定した計算機時刻以降に適用されたオペレータを適用前の状態とする。
- 先行関係の取り消し
オペレータの適用が取り消された場合には、そこに設定されてある先行関係の制約も削除する。

6.5 例題による説明

ここで、3 章で取り上げた花子さんの問題とスケジューリング問題の 2 つを例として、状態遷移計算およびその内部で呼び出される制約充足計算について説明する。

花子さんの問題

図 6.4 に花子さんの問題における制約処理の実行結果を示す。図 6.4 で、`grasp_ope_1`、`expand_ope_1`、`walk_ope_1`、`on_chair_ope_1`、`move_ope_1`、`find_ope_1` はそれぞれ、つかむ、伸ばす、歩く、椅子に乗る、動かす、そして見つけるにあたる行為を表わすオペレータである。これらのオペレータは、パラメータ間の制約充足を行う過程で適用される。

花子さんの問題における要求は、花瓶をとる、ということであるため、まず最初に、`grasping_1` というリレーションを有効化するような変更要求が生成される。`grasping_1` を有効化するためには、このリレーションを行動部にもつオペレータが必要であり、`grasping_ope_1` が選択され、その適用が検討される。`grasping_ope_1` は、`reachable_1` と `at_1` を条件部にもつため、これらのリレーションが有効でなければならない。初期状態では、これらのリレーションは無効であるので、`reachable_1`、`at_1` の有効化要求が生成される。ここで `at_1` は変更許可となっているためそのまま有効化し、それ以上展開されない。また、`reachable_1` は専用のオペレータである `reachable_1.self` によって有効化され、これも展開が終了する。

ここで、オペレータの適用条件が条件部のリレーションの有効化でなく、制約充足である場合には、`reachable_1` および `at_1` に含まれる制約式について制約充足計算によって制約が充足しているかをチェックする。もし制約違反がある場合には、オペレータの適用はできないため、違反の解消のためにさらに変更要求を生成する。`reachable_1` は垂直方向に対する位置関係を表す制約を持ち、`at_1` は水平方向に対する位置関係を表す制約を持つ。まず、現在のパラメータを変更しない場合には、`reachable_1`、`at_1` ともに制約が充足されていない。制約充足計算をおこなった後は、`at_1` に関する制約は充足するが、`reachable_1` に関する制約は充足されない。そこで、とりえず充足可能な `at_1` について `pos_1` の値を変更することによって、水平方向に関する制約充足を行なう。`pos_1` はオペレータ `walk_ope_1` によって変更され、この結果 `walk_1` というプロセスが具体化される。

制約違反が解消されなかった `reachable_1` に関する制約式は、4 章で説明した問題構造のレベルでの制約緩和および精緻化によって、新たな制約式に置き換えられる。その上で制約充足処理を行なうと、`arm_length_1`、`stepping_1`、`on_chair_1` が変更要

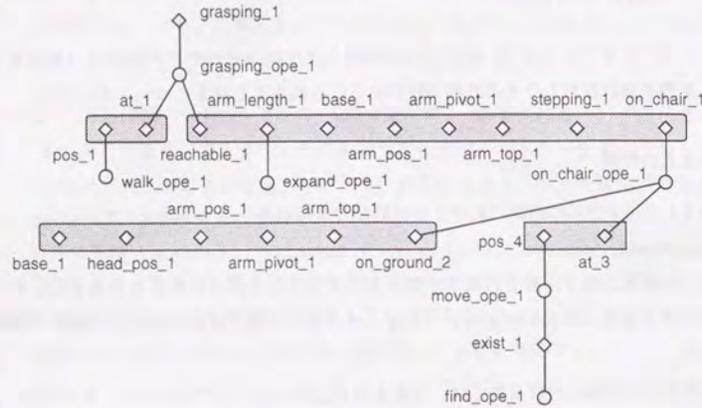


図 6.4: 花子さんの問題の AND-OR グラフ

求としてあらたに設定され、`arm_length_1` は `expand_ope_1` によって、`on_chair_1` は `on_chair_ope_1` によって、`stepping_1` は `stepping_1_self` によって有効化される。ここで展開された3つのオペレータの中で、`on_chair_ope_1` は条件部をもつため、さらにその条件にあたる `on_ground_2`, `at_3` について制約充足を調べる。

`on_ground_2` は、垂直方向の位置に関する制約関係を表しており、これを充足させるためには、`base_1`, `head_pos_1`, `arm_pos_1`, `arm_pivot_1`, `arm_top_1` の値を変更することによって可能である。これらのリレーションは、すべて変更許可となっているため、展開がここで終端する。また、`at_3` は、水平方向の位置に関する制約関係を表しており、これを充足させるために、新たに `pos_4` の値の変更要求が生成される。`pos_4` は、`move_ope_1` によって、変更され、`move_ope_1` の前提条件である `exist_1` の有効化を要求する。そしてこの `exist_1` を有効とするために、`find_ope_1` が適用され、これですべての展開が終了する。

以上の展開を PICCSS によって行なった際の実行ログを図 6.5 に示す。

スケジューリング問題

3章のスケジューリング問題の問題構造である図 3.6 に対応する R-O グラフは、図 6.6 のようになる。ここで、初期状態では、`raw_1`, `raw_3`, `duration_1`, `duration_2`, `duration_3` そして、`level_1`, `level_2`, `avail_1` が有効で、それ以外のリレーションは無

```

--- modify request : grasping_1(-1)
--- expand <R:grasping_1> - <O:grasp_ope_1>
--- expand <O:grasp_ope_1> - <R:reachable_1>
--- expand <O:grasp_ope_1> - <R:at_1>
--- expand <R:reachable_1> - <O:reachable_1_self>
--- self modify ok : at_1
--- modify request : standing_1(1)
--- modify request : stepping_1(-1)
--- expand <R:standing_1> - <O:not_standing_ope_1>
--- expand <R:stepping_1> - <O:stepping_1_self>
--- modify request : pos_1(20)
--- modify request : base_1(-10)
--- modify request : arm_pos_1(20)
--- modify request : head_pos_1(-10)
--- modify request : arm_pivot_1(-10)
--- modify request : arm_length_1(-50)
--- modify request : arm_top_1(-60)
--- modify request : stepping_1(-9)
--- expand <R:pos_1> - <O:walk_ope_1>
--- self modify ok : base_1
--- self modify ok : arm_pos_1
--- self modify ok : head_pos_1
--- self modify ok : arm_pivot_1
--- expand <R:arm_length_1> - <O:expand_ope_1>
--- self modify ok : arm_top_1
--- expand <R:stepping_1> - <O:stepping_1_self>
--- modify request : on_chair_1(-1)
--- expand <R:on_chair_1> - <O:on_chair_ope_1>
--- expand <O:on_chair_ope_1> - <R:at_2>
--- self modify ok : at_2
--- modify request : base_1(-20)
--- modify request : arm_pos_1(-20)
--- modify request : head_pos_1(-20)
--- modify request : pos_4(15)
--- modify request : arm_pivot_1(-20)
--- modify request : arm_top_1(-20)
--- modify request : stepping_1(-20)
--- modify request : at_2(-1)
--- self modify ok : base_1
--- self modify ok : arm_pos_1
--- self modify ok : head_pos_1
--- expand <R:pos_4> - <O:move_ope_1>
--- self modify ok : arm_pivot_1
--- self modify ok : arm_top_1
--- expand <R:stepping_1> - <O:stepping_1_self>
--- self modify ok : at_2
--- expand <O:move_ope_1> - <R:exist_1>
--- expand <R:exist_1> - <O:find_ope_1>

```

図 6.5: 制約処理の実行結果

効となっている。スケジューリングの目標は、製品 X (product_2) と製品 Y (product_1) を得ることであるので、要求として、exist_2 と exist_3 を有効化、つまり値を 0 から 1 へ変更要求を発行する。

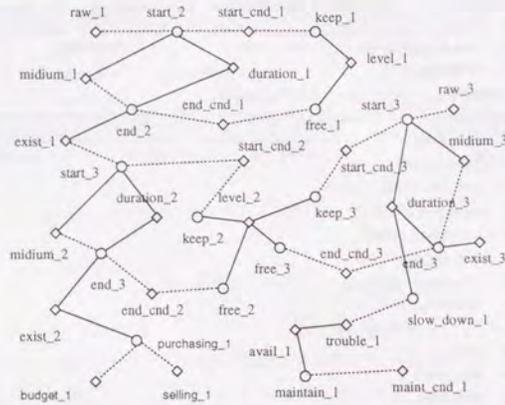


図 6.6: スケジューリング問題の R-0 グラフ

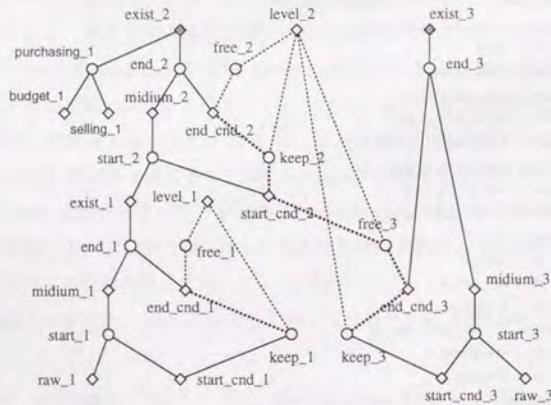


図 6.7: スケジューリング問題の AND-OR グラフ

図 6.7 は、この変更要求に対する AND-OR グラフの展開の様子を表している。こ

の AND-OR グラフの展開において、exist_2 は、end_2 によって有効化され、end_2 の前提となる midium_2, end_cnd_2 の有効化要求が発行される。もし、start_cnd_1, start_cnd_2, start_cnd_3, end_cnd_1, end_cnd_2, end_cnd_3 が有効であるとすると、AND-OR グラフは、引続き start_2, exist_1, end_1, midium_1, start_1 という順で展開され、最後の raw_1 があらかじめ有効であるため、展開が終端する。同様に、exist_3 については、end_3, midium_3, start_3 と展開され、raw_3 によって終端する。

さてここで、start_cnd, end_cnd というクラスに属するリレーションのインスタンスについて考える。start_cnd は作業を開始できる状態、end_cnd は作業を終了できる状態を表し、これらは機械の使用可能状態 level_1, level_2 に依存している。例えば、level_1 は、初期状態は有効 ($p = 1$) であるので、free_1 の適用はできない。なぜなら、free_1 というオペレータは、行動部のリレーションのパラメータの値を +1 するため、すでに有効である free_1 に適用すると、free_1 の値が定義範囲外になってしまうためである。free_1 を適用させるためには、それ以前に level_1 が無効 ($p = 0$) でなければならない。従って、end_cnd_1 を有効化するためには、その前提として level_1 が無効でなければならない。level_1 を無効化するオペレータである keep_1 が end_cnd_1 の仮想的な前提となる。図 6.7 では、このような仮想的な前提条件を太破線で表している。

6.6 アルゴリズムの評価

本章で提案したアルゴリズムの主な特徴としては、次の 2 つがあげられる。まず第 1 に、AND-OR ツリーの探索問題という離散的な問題と、線形等式または不等式の制約充足問題という連続的な問題を同時に扱っている点が挙げられる。人工知能の分野における一般的な制約充足問題は、解空間をあらかじめ離散化し組合せ探索問題として定式化するが、設計パラメータのような連続量を扱うことには向かなかった。離散値と連続量を合わせて扱う例としては、数理計画法の中の混合整数計画問題が挙げられるが、解の探索においてヒューリスティクスに頼っている点では基本的に本研究より勝るものではない。

第 2 の特徴としては、制約関係の動的な扱いが挙げられる。特にここで扱う制約式には、静的な制約と動的な制約の両方が含まれる点が大きな特徴となっている。ここで静的な制約とは、あらかじめリレーションに定義された制約を指し、動的な制約とは、問題解決の過程で明らかになったオペレータ間の先行関係を指す。通常、問題の定義と問題解決とは独立したものであり、対象とする問題は問題解決の過程で変化しない。本研究の制約処理が対象とする問題は、この動的な制約を含めて、問題の一部

である制約を追加したり、無効としたりすることも可能であり、これによって問題そのものが変化する。このような問題解決は、いわば自己言及的なプロセスにあたり、従来の問題解決のアプローチの中ではほとんど見られなかった。

このような特徴をもつ提案したアルゴリズムによって、設計問題という比較的連続量であるパラメータの値を決定しながら構造を定める問題と、計画問題という主に行うの順序や先行関係など離散的な構造を決めながらそれに対応するパラメータの値を決定する問題を、両者を特に区別することなく統一的な枠組みのなかで扱うことが可能となった。これにより、例えば製造を考慮した製品設計や、設計を考慮した製造計画などが柔軟に扱うことができる。つまり、空間的な構造と時間的な構造を同時に考慮しながら問題解決を進めることが、本アルゴリズムによって可能となる。

本アルゴリズムを用いて、空間的構造と時間的構造を同時に検討することのメリットをあらためてまとめると、以下ようになる。

- 不必要な意思決定を回避できる。

一般に、問題解決の最終的な解は、空間的な構造と時間的な構造が対応づけられた、あるいは対応づけられるものでなければならない。例えば、設計問題であってはいずれそこで設定された問題構造は、時間的な構造つまり製造方法と対応づけられる。問題解決をこのような全体のレベルでとらえると、いかに後戻りを少なくするかが問題解決の効率を左右する。コンカレントな問題解決において、空間的な構造と時間的な構造を問題解決の進行に沿って細かな粒度で対応づけることは、結果的に除外される探索枝をはやく発見し、その探索枝に陥ることによる計算時間の無駄を回避することにつながる。

- 新たな要求や事実を獲得しやすい。

空間的な構造を決定する視点と、時間的な構造を決定する視点は、いわば直交する視点である。例えば設計者と製造技術者は、製品開発という問題に対して“モノ”と“方法”というまったく異なる方向から捉える。このことは、場合によって、一方の問題解決が行き詰まったときに、他方が新たな視点から解決の方向を示すといった協調的な作業が可能であるというメリットとなる。これを制約処理の立場からいうと、膨大な探索空間の中から盲目的に探索するのではなく、ある程度の方向性を与えたり、探索枝が完全でないような粗な問題空間において、両者のインタラクションによって探索空間をみずから広げながら解に到達するといったことに対応する。

また、従来のような問題解決を分離して行なう方法に対するデメリットとしては、以下のような点がある。

- 分業がしにくい。

通常、大規模な問題解決の場合には、問題を分割しそれぞれの問題の解を持ちよって当初の大規模問題の解を求める方法がとられる。設計問題と計画問題の場合は、問題の間の関連性が非常に強い一方の問題解決の結果をもとに他方の問題解決を行なうという流れ作業となる。この形態は、定型化した問題解決を行なう場合には、それぞれの作業が分担でき非常に効率的である。つまり、本研究で提案する枠組は、非定型的な問題解決には優れているものの、定型的な問題解決に対しては効率の点で従来の手法に劣る。

- 解が収束しにくい。

提案したアルゴリズムは、制約充足計算と状態遷移計算とが相互補間的に実行される。説明の中では再帰的なアルゴリズムの実行を制限したが、いずれ、バックトラックや仮説に対する多重世界の管理をきちんと行なうことでこの制限をはずし、本来の設計と計画の統合的なアルゴリズムとすることが望まれる。しかし、そのような場合には、おそらく動作が不安定となり、解が収束しないことが予想される。従って、計算量の問題はもちろんのこと、計算の停止の保証という点も検討が必要である。

6.7 6章のまとめ

本章では、本研究の問題解決の枠組みで説明した制約処理について、状態遷移計算を中心として、設計と計画を統合的に扱う制約処理について説明した。状態遷移計算とは、因果関係として条件部と実行部にリレーションが指定されているオペレータによって、リレーションのパラメータをその変更要求に従って変更する処理であり、AND-OR グラフの探索問題を拡張したものとなる。5章で説明した制約充足計算と適宜使い分けることによって、空間的な構造である設計問題の解と時間的な構造である計画問題の解を同時に得ることができる。本章では、このためのアルゴリズムを解説し、また例題をもちいてその動作を解説した。また、一般的な時間推論との比較を行ない、本研究の有効性を議論した。

第7章

製品開発過程への適用

7.1 はじめに

製品開発は、製造業における組織的な問題解決行為である。本章では、本研究で提案する問題解決の枠組みを、実際の製造業における製品開発に対して適用させるための準備として、製品開発の過程で必要となる知識を、本研究の知識表現の方法に対応づけて整理する。なお、ここで対象とする製造業は、加工組立型の機械産業に限定している。

製品開発の過程を計算機によって支援するためには、まずその大前提として、そこで用いられている知識を、単に形式だけでなく、ある程度の内容も含んだ形で表現できなければならない。従って、本章では、製品開発における特に典型的といえる知識について整理し、その内容を本研究の知識の表現形式に沿って具体的に記述する。なお、次章の実験では、本章で示した知識の内容を実際に利用する。

まず、次節では、製造業における製品開発の流れをまとめ、そこで重要となる製品のもつ機能の表現方法を続く3節で説明する。また、それらの機能を形状および工程そして展開するために、4節では形状の表現方法を、5節では工程の表現方法を、提案する知識表現のなかで説明する。また、6節では、これらの形状と工程の対応について述べる。

7.2 製品開発における情報の流れ

製造業は製品というモノを介して市場にサービス（価値）を提供する組織である。従って、製造業の行為は、価値のある情報を生み出すプロセスとして位置付けることも可能である。この場合、製品とは、製品開発によって付加されたさまざまな情報の総体となる。モノ作りを問題解決とみなす本研究の立場からすると、それらの情報は、

市場におけるさまざまな問題の解、あるいは解の一部として、それぞれの消費者の要求に対応することになる。

さて、一般の製造業では、市場からの要求やそこで働く人間および設備の現実に対し、市場調査、研究開発、製品企画、販売計画、生産計画、資材調達、製品設計、工程設計、生産準備、そして製造などのさまざまな個々の問題解決が存在する。モノ作りという製造業が行なう1つの問題解決は、それらの個々の問題解決からなるその企業全体の振舞いによって成立する。

製造業におけるこれらのさまざまな行為の中で、特に製品開発は多くの付加価値を与えるものであり、通常、製品企画、製品設計、工程設計、生産準備、そして製造などの中で、従来は存在しなかった新たな製品が生み出される。製品開発の過程で生成される情報のうち、典型的なものを挙げると以下のようなになる [水野 78]。

1. 製品企画

製品企画の段階では、市場情報に関する報告書や、開発企画書などが作成される。これらは製品の問題構造の再上位に位置付けられる。

2. 製品設計

製品設計では、まず製品に関する基本計画が立案され、そこで最も基本となる設計パラメータや構造が決定され、基本計画書や基本計画図が作成される。また、試作試験のための試作図、定常生産のための本作図を作成する。

3. 工程設計

工程設計では、試作図や本作図をもとに、製造可能な工程を設計し、機械の調整や手順に関するさまざまな技術資料を作成する。

4. 生産準備

生産準備では、工程設計で作成した情報をもとに実際に工程を準備し、不具合を調整する。必要に応じて工程設計や製品設計に情報をフィードバックする。また、品質に関する検討を行ない工程表にまとめる。

5. 製造

製造では、実際の製造を通して製品の品質を確保し、生産量や検査結果などの製造実績を作成する。特に初物生産の場合は、品質に関するさまざまな情報を製品設計や工程設計にフィードバックする。

本研究の問題解決の枠組みを、この製品開発に適用する。図 7.1は、本研究の枠組みに沿って製品開発の流れを示したものである。

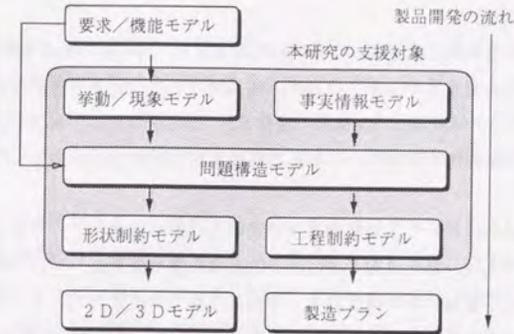


図 7.1: 製品開発の流れと情報モデル

ここで、工程設計、生産準備、そして製造の中での意思決定行為全般を製造計画と呼ぶことにすると、一般に製品開発の流れとしては、製品設計を行なった後に製造計画を行なう。製品設計は、そこで決定した製品の形状を、図面という形で製造計画へ送る。これは、本研究の枠組でいえば、問題解決の精緻化のサイクルが非常に大きく、しかも、第1の問題解決のサイクルでは行為つまり時間的な構造をほとんど考慮せずに形状つまり空間的な構造を決定し、第2のサイクルで時間的な構造を決定しているといえる。この方法では、第1のサイクルで形状を決定した時点で、多くの製造方法の可能性を切捨てていることになり、また、探索の効率としても後戻りが多くなる確率が高い。

製品設計と製造計画の統合化という場合、このような製品設計と製造計画を大きな2つの問題解決サイクルとしてとらえずに、非常にこまかな粒度で形状と工程のつき合わせを行なうことが要求される。つまり、形状がたとえば部品という形で確定する前に、部分的な形状である溝や丸みが設定される都度、対応する工程とつき合わせる。要するに、形状と工程の組み合わせを単位として問題の構造化を行なうことによって、部品あるいは製品を定義する。こうすることによって、製品設計が確定した時点で、製造計画がほぼ終了することになる。

7.3 機能の表現方法

機能とは、要求を表すための特徴的な構造である。通常、要求と機能は1対1ではなく、要求に対してその要求を満足させるような機能を対応づけることが容易でない

場合もある。しかし、ここではとりあえず要求と機能との対応づけは暗黙のうちに行なわれているものとする。

機械における基本的な機能は、つきつめて考えると、何らかの動きを実現すること、力を支えたり伝えたりすること、そして、部品間の空間的配置・接続などの相互関係を作ること、の3つであるともいえる [畑村 92]。製品開発では、状況や目的に応じてこれらの機能を生み出すために、これらの機能を具現化するためのモノや方法を割り当てていく。

問題解決の過程において、機能はより下位のレベルの機能によって置き換えることがよくある。例えば、物を運搬するという四輪車の持つ機能は、下位の機能である地面を転がるという車輪の持つ機能によって記述され、実現される。この場合の上位と下位の関係は、下位の部分問題にとって、与えられた機能は疑似的な要求となる。

このような、製品開発における機能の展開は、本研究の枠組の中のトップダウン精緻化によって進められる。一般的な問題分割によるトップダウンアプローチは、良構造問題の場合には有効であるが、悪構造問題（部分問題の間の独立性が保てない問題）では、有効な手法とはならない場合が多い。そのような分割問題の解が全体の問題の解の一部分とならないような場合でも、精緻化によって機能を展開する問題明確化のプロセスは、よりよい問題分割の方法を検討したり、問題分割を行なわないで部分的に粒度を調整することによる問題解決の方法などに有効となる。

機能センテンス

さてここで、機能を計算機上で記述する方法について考える。通常、人間は機能のことによって表現しており、機能の多くの部分は、このようなことばによる表現によって実際に情報伝達が可能である。従って、機能の計算機上の表現を考える場合に、人間が普段使っていることばをその出発点とすることにする。ここで、このような機能を表す文を“機能センテンス”と呼ぶ。

機能センテンスは、自然言語をベースとしているため、通常その構文はさまざまとなる。構文が厳しいと、人間の発想や機能の表現意欲を阻害することにもなりかねないが、ここでは、機能センテンスを、最終的に以下のような、挙動（行為）的な表現と、現象（属性）的な表現の2つの形式によって表現する。

- 挙動（行為）的な表現 — 「○○のとき、○○により○○を○○する」という形式で表現される機能センテンス。挙動的な表現では、「○○する」というような行為による手続的な表現であり、2つの状態の間での状態遷移を表す。ここで「○○のとき」は前提を、「○○により」は行為者を、「○○を」は行為

の対象をそれぞれ表す。

- 現象（属性）的な表現 — 「○○のとき、○○は○○である」という形式で表現される機能センテンス。現象的な表現は、挙動のもととなるような現象を表し、「○○である」という宣言的な表現となる。この表現は1つの状態を表す。ここで「○○のとき」は前提を、「○○は」は対象をそれぞれ表す。

さまざまな要求を機能として表現する場合、機能センテンスは必ずしもこの2つのような単純な形式にはならない場合が多い。このような場合は、まず、複文は先の2つの形式に合うように動詞に注目して分割し、よけいな形容詞、副詞を削除する。また、主語は3人称とし、問題解決者が主語とならないようにする。さらに、計算機自身が問題解決の過程に直接関与しないような表現（例えば計算機への入出力など）とする。このようにして、複雑な文章も、基本的にこの2つの表現形式の組合せによって、表現することが可能となる。

機能のプリミティブ

機能センテンスにおける各語彙は、プリミティブに対応する。機能センテンスの語彙としては、大きく名詞句（モノの表現）と動詞句（コトの表現）の2つがあり、機能はこれらの表現の中にうめこまれている。これらの語彙に対応するプリミティブは、おおよそ名詞句がエンティティに、動詞句がプロセスに対応し、それらのプロパティとしてリレーションやオペレータが設定される。

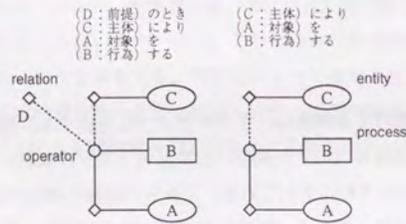
なお、機能そのものは一般的に、上記のモノ表現、コト表現の中間的な位置にあるため、場合によって複数の表現方法が考えられる。たとえば「液体の圧力を得る」という表現では、機能を「液体の圧力」というモノ表現とすることもできるし、「圧力を得る」というコト表現とすることもできる。このような場合は、あらかじめ登録されたプリミティブを参考に、問題解決者が適宜判断する。

機械設計における考慮すべき制約としては、空間、重量、加工法、組立・分解、操作性、安全性、耐久性、標準・法規などが挙げられるが [畑村 92]、これらの項目は、製品を全体としてとらえた場合の、非常に上位のレベルの要求にあたるプリミティブである。これに対し、製品の詳細な個々の部分に関する要求としては、位置（距離）、位相、運動、エネルギー、化学変化、物理属性、情報伝達、そして挙動属性、現象属性、一般属性などの項目が挙げられる。このように、機能センテンスで利用される語彙は、問題解決の精緻化のレベルによって、あらかじめある程度整理することが可能である。

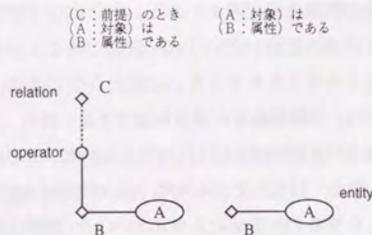
表 7.1 に、下位レベルの機能センテンスで比較的良好に用いられる語彙を、クラス階層の中で整理したものを示す。

機能センテンスの各語彙に対応するそれぞれのプリミティブは、それ単独では完全に機能を表すことができない。提案する知識表現における機能の表現は、これらのプリミティブの構造によって行なわれる。例えば、先の2つの機能センテンスの基本形は、図 7.2 のようなプリミティブの構造に置き換えられる。

汎用的な機能については、テンプレートによって、あらかじめ対応するプリミティブの構造を知識ベースに登録することにより、このような機能センテンスからプリミティブの構造への変換が容易になる。本研究の枠組みでは、これらのテンプレートは、知識ベース管理者が定義せずに、問題解決者が実際に行なった問題解決の事例の中から抽出する。



(a) 挙動 (行為) 的な表現



(b) 現象 (属性) 的な表現

図 7.2: 機能表現の基本形

機能の展開の方法

機能の展開の方法は、まず製品のもつべき基本的な機能を文章で記述した後、以下の2つの質問を問題解決者に対して行なうことによって進める。

- WHAT 型の質問: 機能センテンスに含まれる語彙の中で、実体に関する表現 (モノ表現) に注目し、それはどのようなものなのかを質問する。
- HOW 型の質問: 機能センテンスに含まれる語彙の中で、行為や方法に関する表現 (コト表現) に注目し、それはどのようなことなのかを質問する。

この機能の展開は、基本的に人間が計算機と対話的に行なう作業であり、展開に関する知識は人間が持つ。ただし、この機能の展開作業をできるだけ省力化するために、汎用的な機能は、先述のようにテンプレートとしてあらかじめ準備し、知識ベースとして登録しておく。これらのテンプレートは、提案する枠組において、問題構造のトップダウン精緻化を行なう中で適用し、その結果として、過去の有効な機能展開の知識による新たな展開の自動化が可能となる。

表 7.1: 機能のプリミティブ表現

内容	relation class	super class	connection
動力変換	trans.e	energy	pressure
圧力調整	regulate.e	energy	pressure
空間分割	divide	topology	space
道形成	pass	topology	hole
部品保護	guard.t	topology	part
流制約	restrict.m	motion	speed
部品移動	move	motion	part
動力伝達	trans.m	motion	part
復元力生成	reverse.m	motion	part
構造形成	structure.p	place	entity
状態回復	resume	place	entity
動作限定	restrict.p	place	entity
部品固定	hold	place	part
部品接合	fixed	place	part
情報表示	inform	information	media
漏れ防止	seal	phenomenon	liquid
摩擦軽減	friction	ph.feature	slide

7.4 形状の表現方法

製品という実体の表現は、従来よりプロダクトモデルの研究としてさまざまな方法が提案され、また実際の CAD システムにおいて利用されている [Kimura93]。製品の表現において、最も重要な情報は形状の幾何的な表現である。形状の幾何表現の方法として、代表的な手法が CSG (Constructive Solid Geometry) と B-Rep (Boundary Representation) である。また、これらの形状表現をベースとしていかに形状特徴 (Feature) を表現するかという研究がさかんである [Salomons93]。

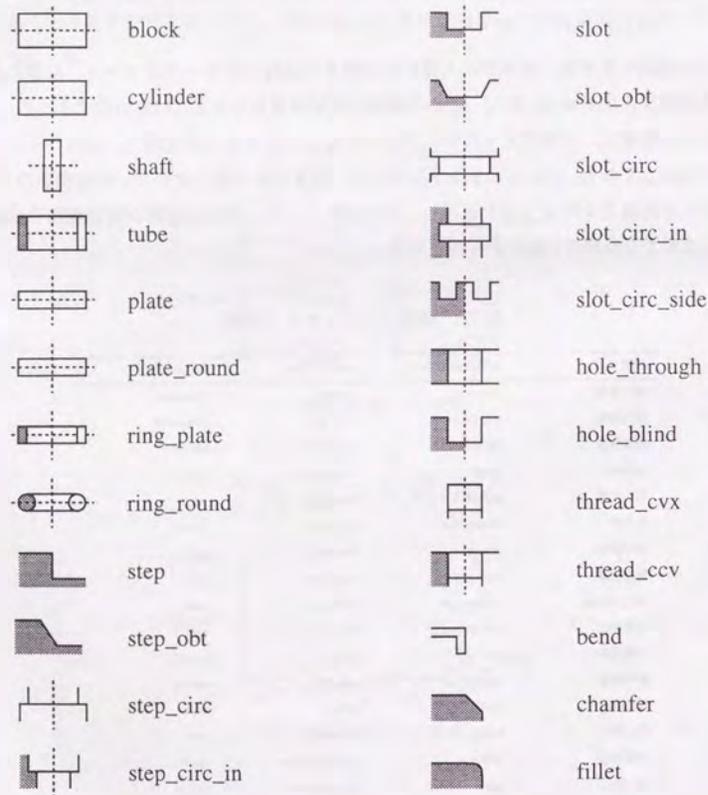


図 7.3: 形状表現のためのエンティティ

本研究では、製品の形状を表現するにあたり、これらの形状特徴に相当するものを、主にエンティティとリレーションからなるプリミティブの構造によって対応させる。形状特徴を表すために、あらかじめ知識ベースに登録したエンティティとしては、図 7.3 のような種類がある。これらのエンティティの多くは、従来の CAD 研究における形状特徴と、ほぼ 1 対 1 に対応している。

図 7.3 に示したエンティティは、実際には、それらのエンティティの置かれているグローバル空間上での基準方向によって、さらにサブクラスに分けられる。ここでは、製品形状は 2 次元としたため、例えば、block というクラスは、block.h (水平方向) と block.v (垂直方向) の 2 つのサブクラスのいずれかによって定義され、slot というクラスは、slot.r (右方向), slot.l (左方向), slot.u (上方向), slot.d (下方向) の 4 つのサブクラスのいずれかによって定義される。

形状特徴と表すそれぞれのエンティティは、プロパティとしていくつかのリレーションを持っている。これらのリレーションによって逆にエンティティの性質が表現されているといってもよい。エンティティのもつリレーションとしては、表 7.2 に示すようなものがある。ここで、cedge_ccv は、回転軸と垂直の断面の形状が凹であるものであり、いわゆる鞍状の形状 (見方によっては凹とも凸ともいえる形状) の稜線は、回転の軸によって cedge_cvx か cedge_ccv が決定される。

表 7.2: 基本形状の種類

内容	relation class	sub class
平面	pface	pface.r, pface.l, pface.u, pface.d
円筒面	cface	cface.r, cface.l, cface.u, cface.d
円筒面 (凹)	cface_ccv	cface_ccv.r, cface_ccv.l, cface_ccv.u, cface_ccv.d
六各柱の側面	hexa_face	hexa_face.r, hexa_face.l, hexa_face.u, hexa_face.d
R 円筒面	rface	rface.r, rface.l, rface.u, rface.d
球面	sface	sface.r, sface.l, sface.u, sface.d
球面 (凹)	sface_ccv	sface_ccv.r, sface_ccv.l, sface_ccv.u, sface_ccv.d
直線の稜線	edge	edge.r, edge.l, edge.u, edge.d
直線の稜線 (凹)	edge_ccv	edge_ccv.r, edge_ccv.l, edge_ccv.u, edge_ccv.d
円形の稜線	cedge	cedge.r, cedge.l, cedge.u, cedge.d
円形の稜線 (凹)	cedge_ccv	cedge_ccv.r, cedge_ccv.l, cedge_ccv.u, cedge_ccv.d
中心線	center	center.h, center.v
原点	org	org.x, org.y

表 7.2 に示す各基本形状は、位置を表すリレーションである pos.x または pos.y のいずれかのサブクラスとなっている。つまり、1 つのプリミティブが扱う位置情報は 1 次元であり、従って例えば pface という平面の位置を表すリレーションは、実際に

は pface_r(右), pface_l(左), pface_u(上), pface_d(下) といった複数のリレーションをサブクラスとしてもち、状況に応じて使い分けている。ここで pface_r は (1,0,0) を法線方向とする平面であり、x 軸上の位置をそのリレーションの値として持つ。同様に、pface_l は x 軸上の位置、pface_u, pface_d は y 軸上の位置の値を持つ。つまり、r, l, u, v を末尾にもつプロパティは pos_x のサブクラスであり、_u, _d, _h を末尾にもつプロパティは pos_y のサブクラスである。

上記以外の形状に関するリレーションとしては、図 7.3 に示すようなものがある。表 7.2 の基本形状はパラメータのみをもつリレーションであるのに対し、図 7.3 のリレーションは、制約式を持っている点の特徴である。図 7.3 には、制約式において参照している他のリレーションのクラスも併せて示す。

表 7.3: 位置関係を表すリレーション

内容	relation class	sub class	connections
直径	diam	diam_h, diam_v	cface, cface
	diam_in	diam_in_h, diam_in_v	cface_ccv
半径	radi	radi_r, radi_l, radi_u, radi_d	center, cface
	radi_in	radi_in_r, radi_in_l, radi_in_u, radi_in_d	center, cface_ccv
距離	length	length_h, length_v	pface, pface
		length_r, length_l, length_u, length_d	pface, edge
厚み	thick	thick_h, thick_v	pface, pface
深さ	depth	depth_r, depth_l, depth_u, depth_d	pface, edge
間隔	gap	gap_face_h, gap_face_v	pface, pface
ネジ部長さ	thread_length	thread_length_r, thread_length_l,	pface, cedge
		thread_length_u, thread_length_d	
位置調整	adjust	adjust_position_h, adjust_position_v	pos, pos
挿入	insert	insert_h, insert_v	cface, cface_ccv
接触	meet	meet_face_r, meet_face_l	pface, pface
		meet_face_u, meet_face_d	
位置合わせ	align	align_pface_r, align_pface_l	pface, pface
		align_pface_u, align_pface_d	

このようなリレーションを用いて、各エンティティは定義されている。表 7.4 に、いくつかのエンティティについてそのプロパティの内容を示す。表 7.4 において、例えば shaft_h は、軸方向が水平方向であるシャフトであり、左右の平面 pface_r, pface_l と上下の円筒面 cface_u, cface_d、そして水平方向、垂直方向の中心線 center_h, center_v を持つ。また、上下 2 つの円筒面の間の関係として直径 diam_h が定義されており、左右 2 つの平面の間で距離 length_h が定義されている。表 7.4 で取り上げた各エンティティの概略形状とリレーションの対応を、図 7.4 に示す。

表 7.4: 形状エンティティの内容

entity name	properties			
shaft_h	length_h	diam_h	center_h	center_v
	cface_u	cface_d	pface_r	pface_l
	org_x	org_y		
step_circ_u	diam_ppd_v	diam_v	depth_circ_r	depth_circ_l
	center_v	cedge_r	cedge_l	cface_r
	cface_l	pface_u	org_x	org_y
hole_through_v	diam_in_v	length_v	center_v	pface_u
	pface_d	cface_ccv_l	cface_ccv_r	org_x
	org_y			
thread_cvx_r	diam_h	thread_length_r	center_h	cface_u
	cface_d	pface_r	cedge_r	org_x
	org_y			

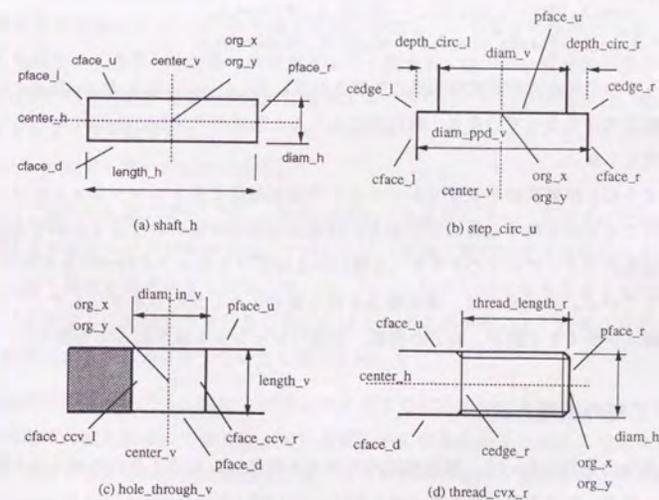


図 7.4: 形状エンティティの例

これ以外に、エンティティ間の従属関係を表すリレーションとして has_step, has_slot, has_hole, has_chamfer, has_fillet, has_thread, has_bend などがある。これらのリレーションは、コネクションをもつ2つのエンティティのうち、一方のエンティティのクラスがすでに決定しており、例えば has_step に属するリレーションは何らかのエンティティと step のクラスに属するエンティティとをつなぐものである。

この従属関係を表すリレーションは、そのリレーションがもつ制約式のパラメータの種類によって、コネクション相手のエンティティがある程度限定される。例えば、has_step_circ_r というリレーションは、所有エンティティの cface_u と従属エンティティである step_circ_r の cface_u を、また所有エンティティの cface_d と従属エンティティの cface_d とを対象に制約式を定義しているため、cface_u, cface_d というプロパティをもった cylinder, shaft などであれば所有エンティティとなれない。

形状を表す以上のようなエンティティをリレーションによって接続した場合、それによって計算機は、制約式をあらたに生成する。以下に、いくつかの例によってその過程を説明する。まず、tube_h_1 と shaft_h_1 を insert_h_1 によって接続した場合、insert_h_1 には、 $1@cface_ccv_u = 2@cface_u$ と $1@cface_ccv_d = 2@cface_d$ という2つの制約式が定義されているため、

```
tube_h_1@cface_ccv_u_1 = shaft_h_1@cface_u_1
tube_h_1@cface_ccv_d_1 = shaft_h_1@cface_d_1
```

という2つの制約式が計算機内部に生成される。もし、tube_h_1 と shaft_v_1 のように、軸方向が異なっていると、制約式はきちんと生成されずに、insert_h_1 という関係が成立しない。

このように、位置関係を表すリレーションや従属関係を表すリレーションには、そのリレーションが意味する概念世界あるいは現実世界の対象に沿うような形であらかじめ接続先のプリミティブのクラス（正確にいえばプリミティブの持つべきプロパティ）を決定している。本研究では、機械製品を扱う場合によく使われるプリミティブとして、登録エンティティ数が、約260種類、登録リレーション数が約520種類となった。

7.5 工程の表現方法

製品開発の過程において、製品形状のモデルと同様に、製造工程のモデル化が重要である。従来、製造工程に関するモデル化は、CAPP (Computer-Aided Process Planning) の分野、あるいはスケジューリングの分野で取り組まれてきた。また、古典的な IE (Industrial Engineering) の研究では、作業分析を通して人間の作業による生産性と品質の向上が進められてきた。

技術やノウハウといわれるものの多くは、行為や方法に関する知識として表現される。ここでこの種の知識のことを便宜上“操作的知識”と呼ぶ。この操作的知識は、その記述の難しさ故に保存や伝承が非常にむずかしい。やって見せることによってしか伝わらないような職人的な技術などは、通常我々が知識と呼んでいるものと区別し暗黙知などともいわれるが、これらの本来書き下すことのできないものを、何らかの形で表現したものが、この操作的知識である。本節で扱おうとしている製造工程の表現は、つきつめるとこの操作的知識に相当するため、ここで、あらかじめこの操作的知識の表現の難しさについて再確認しておくことにする。

さまざまな行為を表現する場合、そこには、その主体と対象との関係の中で何らかの目的概念がすでに含まれている。つまり、この種の知識は、現実の世界と人間の要求との接点として生み出され特徴づけられるものであるため、本来、状況や視点から独立して記述することができない。従って、この種の知識はもともと一般化することが非常に困難である。モノに関する知識は、一般化あるいは抽象化することで知識管理の効率化が図れるが、技術やノウハウといった行為をベースとする操作的知識は、この一般化が難しいのである。

次に、操作的知識は、要素の順序関係が意味をもつという性質がある。例えば A と B の2つの要素によって操作的知識を表現する場合、その要素の順序が A,B なのか B,A なのかで、通常その知識の内容がまったく変わる。従って、要素数が多くなればなるほど、扱うべき情報量が飛躍的に増大する。さらに、この順序関係は、番号で識別できるような全順序関係でなく、半順序関係つまりグラフ構造でなければ表現できないという点でも扱いが厄介である。

さらに操作的知識の場合、同一の対象を表現する場合にでも、表現された知識の粒度はさまざまなレベルが存在する。このために、知識表現の粒度が利用状況で必要とする粒度と異なる場合が多くなる。従って、同一の対象に対するさまざまな粒度をあらかじめ想定して知識表現を行わなければならないとすれば効率が悪く、適切な粒度を決定する基準はあらかじめ存在しない点も管理を難しくしている。

知識の分節についても、粒度と同様にさまざまな形態が考えられる。一般に行為は一連の動作の流れによって成り立っているが、その行為の始まりがどこで終りがどこかということをしきりと決めるためには、なんらかの恣意的な基準を導入する必要がある。通常、ある要求に見合うものが得られた時がプロセスの終了とされるが、本来行為というのはそれ以降もなんらかの行為が続いており、そこで設定した境界は、目的あるいは視点に依存したものである。このような操作的知識の分節の問題は、発想支援の研究 [堀 94] とも共通する。

工程の粒度について

標準化、ルーチン化された操作的知識は、扱うことが比較的容易であるが、これはあらかじめ想定された問題に対応する場合にのみ有効なものである。例えば、大量生産における流れ作業は、決められた動作を繰り返すことによって目的とする製品を生産することができるが、あたらな製品の開発時にはその工程に関する知識はまったく応用できない。しかし、通常、人間がもつ問題解決能力は、ある程度予期しない問題にも適用できるような操作的知識を持っており、このような状況で利用される知識を、できる限り記述する努力をする必要がある。そのためには、粒度の異なる知識の階層的な管理が必要となる。

工程に関する知識を記述する際に、その記述の粒度は、特徴の関連度、対象の類似性、同一主体、資源の共有、時間的な密度、グラフの強度、などを手がかりとして通常は決定される。工場における製造工程では、従来よりいくつかの階層が設けられ、その上で表現されている。通常、より下位の階層の知識によって上位の知識が定義されており、上位の知識が適用できない場合は、下位の知識を再構成することにより動的に対応する。

工程の階層化の例として、工程、単位作業、要素作業、そして単位動作という4つのレベルによって表現する方法がある [泉 94]。この方法は、IE の研究の一貫として作業の効率化をはかる目的で利用されている枠組であるが、本研究においても参考になる。この階層化の方法に従えば、例えば、工程レベルが、芯もみ、旋削、研磨に対応し、単位作業のレベルが、面1穴明け、面1芯もみ、面2穴明け、面2芯もみに、要素作業のレベルが、部品取付、面1穴明け、現品外しに対応し、そして、単位動作のレベルは、現品をトレイからとる、チャックを開く、現品を当てる、チャックを締める、などの作業者の個々の動作に対応する。

本研究の知識表現におけるプリミティブとしては、これらのすべてのレベルに対して用意する必要がある、問題の精緻化のレベルに応じて問題解決時に使い分けることが望まれる。適用する粒度は、その時点での問題解決の精緻化の対象に依存し、例えば、外注工程のように工程の内容をブラックボックス化できるような場合は、粒度は大きくとり、内製でしかも工場内の細かな資源との対応をとる必要がある場合は、それだけ粒度は細かくなる。

プリミティブの内容

一般的に、工程は、操作対象（製品、部品）と資源（機械、ツール）との対応関係を基準として表現される。これは、行為や方法の直接的な表現ではないが、操作対象

と資源を指定することで、ある程度そこで行なわれる行為や方法が想定可能であることによる。本研究でも、知識表現の最小単位であるプリミティブのレベルでは、対象と資源との関係において工程を特徴づけている。

工程に関するプリミティブを表現するために、あらかじめ資源に相当する工場の情報、機械の情報、そしてツールの情報をエンティティとして登録する。表 7.5、表 7.6、表 7.7 にそれぞれのエンティティの内容を示す。

工場、機械、そしてツールは、階層構造となっており、工場はいくつかの機械を、機械はいくつかのツールを所有している。各エンティティのメンバーとして設定されているプリミティブは、そのエンティティが一般に所有している下位レベルの資源である。なお、作業者は機械と同じクラスとして定義した。

表 7.5: 工場のプリミティブ表現

名称	entity class	members
組立ショップ	assy_shop	labor
切削加工ショップ	metal_cutting_shop	turning_mcn, milling_mcn, drilling_mcn, nc_mcn, machining_center
塗装ショップ	paint_shop	painting_mcn
メッキ工場	plating_shop	plating_mcn
研磨工場	grinding_shop	grinding_mcn
精密加工工場	lapping_shop	lapping_mcn
熱処理工場	heat_treating_shop	heat_treating_mcn
塑性加工ショップ	plastic_deforming_shop	press_mcn_small, press_mcn_large, flatter_mcn, cutter_mcn
射出成形工場	injection_mold_shop	injection_mcn
鋳造工場	casting_shop	casting_mcn
焼結工場	sintering_shop	sintering_mcn
ロストワックス工場	lost_wax_shop	lost_wax_mcn
溶接ショップ	welding_shop	welding_mcn

工程を表すプリミティブは、プロセスとオペレータの2つからなる。ここでオペレータはプロセスのプロパティであり、通常は、プロセスとオペレータは対になっており同様の名前を先頭を持つ。例えば、取るという作業は、put_proc というプロセスと put_op というオペレータから構成される。また、場合によっては、プロセスを put と動詞形とし、オペレータを putting と現在進行形とする。

工程を表すプリミティブは通常、表 7.8 ようなプロセスで表される。工程に対応するプロセスは、一般に、製品のそのプロセスにとって実行前の状態を in_product、実行後の状態を out_product とし、そのプロセスで利用する資源を tool, machine, factory などと表現すると、図 7.5(a) のような形になる。図で、predecessor, successor

表 7.6: 製造資源のプリミティブ表現

名称	entity class	members
作業者	labor	-
旋盤	turning.mcn	cutting.tool
フライス盤	milling.mcn	plain_milling, flat_end_mill, ball_end_mill
ボール盤	drilling.mcn	drilling.tool, center_reamer, reamer, tap.tool
NC 工作機	nc.mcn	cutting.tool
マシニングセンター	machining.center	cutting.tool
プレス機械 (小型)	press.mcn.small	press.die
プレス機械 (大型)	press.mcn.large	press.die
メッキ機械	plating.mcn	-
研磨機械	grinding.mcn	-
ラッピング機械	lapping.mcn	-
熱処理機械	heat.treating.mcn	-
かしめ機械	flatter.mcn	-
切断機	cutter.mcn	-
射出成形機	injection.mcn	-
鋳造機械	casting.mcn	-
焼結機械	sintering.mcn	-
ロストワックス機械	lost.wax.mcn	-
溶接機械	welding.mcn	-

表 7.7: ツールのプリミティブ表現

名称	entity class	members
切断工具	cutting.tool	-
平フライス	plain.milling	-
フラットエンドミル	flat_end_mill	-
ボールエンドミル	ball_end_mill	-
ドリル	drilling.tool	-
センターリーマ	center_reamer	-
リーマ	reamer	-
タップ工具	tap.tool	-
成形型	mold	-
プレス型	press.die	-

表 7.8: 工程のプリミティブ表現

工程名称	process class	target	precondition	resource
取出	put	entity		labor
装着	setup	entity	entity	labor
挿入	insert	entity	entity	labor
固定	fix	entity	entity	labor
ネジ固定	fix_by_thread	entity	entity	labor
塗装	painting	entity	entity	paint.mcn
突切り	cut_off	cface	cface	cutting.tool
正面旋削	cut_pface	pface	cface	cutting.tool
外周旋削	cut_cface_cvx	cface	cface	cutting.tool
内周旋削	cut_cface_ccv	cface_ccv	cface	cutting.tool
R旋削	cut_fillet_circ	fillet_circ	cedge	cutting.tool
面取旋削	cut_chamfer_circ	chamfer_circ	cedge	cutting.tool
段旋削	cut_step_circ	step_circ	cface	cutting.tool
ねじ加工	cut_thread_cvx	thread_cvx	cface	cutting.tool
センタモミ	cut_center_hole	center_hole	pface	cutting.tool
溝旋削	cut_slot_circ	slot_circ	cface	cutting.tool
溝加工	milling_slot	slot	pface	flat_end_mill
フライス加工	milling_pface	pface	pface	plain_milling
曲面加工	milling_sface	sface	pface	ball_end_mill
穴加工	drilling	hole_through	pface	drilling.tool
面とり	chamfer_by_drill	chamfer_circ_in	hole_through	center_reamer
リーマ加工	reaming	hole_through	hole_through	reamer
タップ加工	tapping	thread_ccv	hole_through	tap.tool
打ち抜き	press_cutting	plate	plate	press.die
曲げ	press_bending	bend	plate	press.die
プレス成形	press_forming	plate	plate	press.die
メッキ	electric_plating	entity	entity	plating.mcn
研磨	grinding	entity	entity	grinding.mcn
平面研磨	grinding_pface	pface	pface	grinding.mcn
ラッピング仕上	lapping	pface	pface	lapping.mcn
熱処理	heat_treatment	entity	entity	heat_treating.mcn
かしめ	flattening	entity	shaft	flatter.mcn
切断	cutting	entity	entity	cutter.mcn
鋳造	die_casting	entity	entity	casting.mcn
ロストワックス	lost_waxing	entity	entity	lost_wax.mcn
射出成形	injection_moldingP	entity	entity	injection.mcn
焼結	sintering	entity	entity	sintering.mcn
アーク溶接	arc_welding	entity	entity	welding.mcn

は、それぞれ先行プロセス、後続プロセスであり、製品を直接意識しないこのような関係も定義されることがある。

このようにして設定されたプリミティブの関係は、内部的には図 7.5 (b) のような R-O グラフとなる。この図 7.5 のように、通常、プロセスにはオペレータが定義されており、そのオペレータによって推論が行なわれる。また、この工程に関する推論に備えて、さらに、製品には exist というクラスのサブクラスが、資源には avail というクラスのサブクラスが設定され、それらが決められた形で接続される。

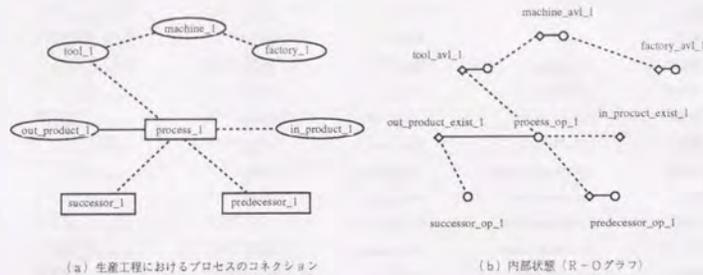


図 7.5: 工程の表現方法

本研究では、このような製造工程に関するプリミティブを知識ベースとして登録した。登録したプリミティブは、資源に関するエンティティが 42 種類、工程に関するプロセスとオペレータがそれぞれ 38 種類となった。

7.6 製品形状と製造工程の対応関係

実際の企業では、ユーザの要求は製品の注文などの形で与えられる。場合によっては、特別仕様、クレーム、修理の以来、部品交換といった要求が与えられることもある。製造業における問題解決としては、なにもない状態から時間をかけて新規に製品を作り上げていく開発設計 [長澤 89] も問題解決の 1 つであるが、このようなユーザ要求を、ある時間内に、製品 (部品) あるいはなんらかの行為によって解決することもまた、非常に一般的かつ頻繁に行われる問題解決である。本研究では、この両者を対象とする。

製品開発では、機能という形で表現された要求を、どのような粒度まで精緻化するか意思決定の大きなポイントとなるが、1 つの判断基準として製造可能性を考慮して決定する方法が挙げられる。つまり、そこで設定された問題構造が、直接的または

間接的に、空間的な構造にあたる形状のプリミティブが、時間的な構造にあたる工程のプリミティブと対応づけられているかどうかで判断する方法である。

本研究では、このために、プリミティブのレベルで形状と工法のつき合わせを、あらかじめ定義し知識ベースとして登録しておく。また、プリミティブとして定義できないような、マクロな対応関係は、テンプレートとして登録する。これらプリミティブやテンプレートを用いて、製品形状と製造工程とが対応づけられた構造となるまで問題構造を精緻化することが、その問題構造の製造可能性を保証することにつながる。

製品形状と製造工程の対応関係の一般的な形としては、図 7.6 のような、形状を表すエンティティ間の関係を表すリレーション (これは形状特徴に相当する) に対して、そのリレーションを有効化する独自のプロセスおよびオペレータを設定したものが挙げられる。図 7.6 の inserted という関係と inserting という行為との対応以外でも例えば、has_hole と cutting_hole など、さまざまなプリミティブのレベルでの対応関係が、あらかじめ登録可能である。

また、形状を表すプリミティブは通常 exist というクラスのリレーションを持ち、これが計算機内部に生成される時点で無効化された状態となっている。従って、問題解決では、その形状プリミティブを実際に存在させるための工程をオペレータとして定義し、そのオペレータによって、この exist のインスタンスを有効化する必要がある。このときの、形状と工程の対応関係もまた、非常に基本的な製品形状と製造工程の対応関係の一例である。

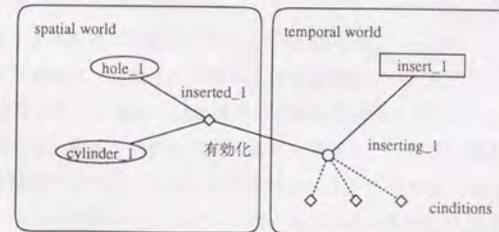


図 7.6: 形状と工法の対応関係

プリミティブのレベルでの形状と工法の対応関係は、非常に単純に表現できるのに対して、テンプレートレベルでの形状と工法との対応関係は、一般に複雑である。これらをあらかじめ十分に記述することは不可能といえる。また、この種の知識は、問題解決者それぞれ固有のものであったり、状況に非常に依存したものであったりする

ため、この知識をあまり固定的に定義して扱うことは、かえって知識の利用効率の低下や、あらたな知識獲得の阻害につながる可能性もある。従って、本研究では、きわめて一般的な対応関係といえる部分（プリミティブによる表現の補助的な使い方）以外は、このレベルの知識をあらかじめ定義することは避け、問題解決者が個々に行なった問題解決知識をできるだけ再利用する立場をとった。問題解決者が自分で行なった問題解決の断片や、状況と視点が類似している他の問題解決者の問題解決の断片であれば、利用による効果は大きくなる。

7.7 7章のまとめ

本章では、提案する問題解決の枠組みを実際の製品開発の過程に適用させるための準備として、製品開発過程で利用する知識の整理を行なった。製品開発の対象としては、ここでは機械製品とした。まず、製造業における製品開発の一般的な流れを、そこで生成される情報に着目して整理した。本研究の問題解決の枠組みを適用する範囲としては、その中の特に製品設計と製造計画とした。続いて本章では、製品の要求仕様である機能の表現方法と、それらの機能を展開した結果として得られる製品形状と製造工程のそれぞれの表現方法について、提案する知識表現の上で説明し、具体的にいくつかの知識の定義を行なった。ここで定義した形状の表現はCADのプロダクトモデルに、工程の表現はCAPP (Computer Aided Process Planning) の工程モデルにほぼ対応する。本章ではまた、これらの製品形状と製造工程の対応のとり方についても解説した。

第8章

実験と評価

8.1 はじめに

本論文ではこれまで、モノ作りを人間の行なう問題解決という視点でとらえ、そのための計算機による支援方法について論じてきた。本章では、本研究で提案する枠組みが、現実の製造業における製品開発に対して、どこまで適用可能であるかを、具体的な事例を用いて検証する。

本章で行う実験は、以下の3つから構成される。まず、実際の製品開発の過程を計算機上に表現可能であるかを検証する。続いて、設定された問題構造を用いて、各種のパラメータを計算する。特に、製品形状と工程プランというまったく異なる2種類の情報が製品開発の結果として計算可能であることを確認する。最後に、設計変更のシミュレーションを行なう。設計変更としては、パラメータの修正で対応できる場合と、問題構造を追加変更する場合に分けて行なう。

以下、まず2節では、対象として取り上げた事例について説明し、3節、4節で、新規に製品開発を行なう場合の問題構造の明確化の過程をPICCSSによって再現する。特に4節では、製造を考慮した設計過程を取り上げ、設計問題と計画問題とを統合的に扱う本研究のアプローチの有効性を確認する。5節では、得られた問題構造をもとにパラメータの計算をおこない、その結果を示す。続いて6節では、設計変更のシミュレーション結果について報告する。最後に7節で、実験結果の評価と考察を述べる。

8.2 対象事例の説明

事例としてとりあげたメーカーは、典型的な加工組み立て型の生産形態をとる中小規模の企業である。このメーカーでは、製品の企画から設計、製造、販売までを一貫

して行っており、モノ作りを市場の中における製造業の問題解決としてとらえる本研究の立場からすると、全体をとらえる上で非常に適当な事例といえる。

対象とする製品は、図 8.1 に示すようなエア駆動ポンプである。エア駆動ポンプは、圧縮エアを動力とし、その動力によって左右にある2つのダイヤフラムを往復運動させることにより、液体を搬送する。エア駆動ポンプは、以下のような特徴をもつ。

- グランド等の回転部がないため、ポンプからの液漏れは全くない。また、摩擦部がないため空転してもポンプの発熱、摩耗等がない。自吸タイプであるので、呼び水は不要。
- スラリーを搬送するときは、ポンプ内の流速が遅いので摩耗が少なくポンプの寿命が長い。また、乳化、攪拌が少なく、フロック等の柔らかいものでも破損しない。
- 吐出側で流量の調節ができる。吐出バルブを開閉することで吐出量の調整が容易に行なうことができ、ポンプの作動と停止もこの吐出バルブによって可能。
- 電気を使用しないエア駆動のため、可燃物の搬送に最適。防爆仕様が不要であり簡単で安全。また、供給空気圧以上には液圧は上昇しないため、安全弁も不要。

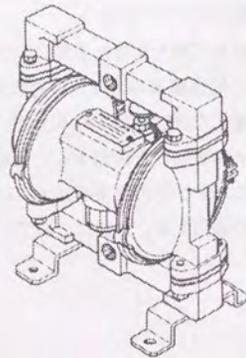


図 8.1: 対象とした製品

製品の構造と製造工程（設備）の構造を図 8.2 および図 8.3 に示す。図 8.2 は、このメーカーが扱う製品の中で比較的普及型の機種の内蔵構造である。図のように、この機種は、およそ 90 種類の部品から構成され、1 機種あたりの部品総数は約 220 である。

このポンプは、主に中央の機構部分と左右のダイヤフラム部分そして液の搬送経路となる上下のマニホールド部分から構成されている。機構部分には、エア供給のためのエアバルブ、往復運動を切替えるための切替バルブ、ポンプが異常停止した場合の復帰バルブ、そしてエアの排気口が取り付けられている。また、上下のマニホールド部分には、合計 4 つのボールバルブが取り付けられている。

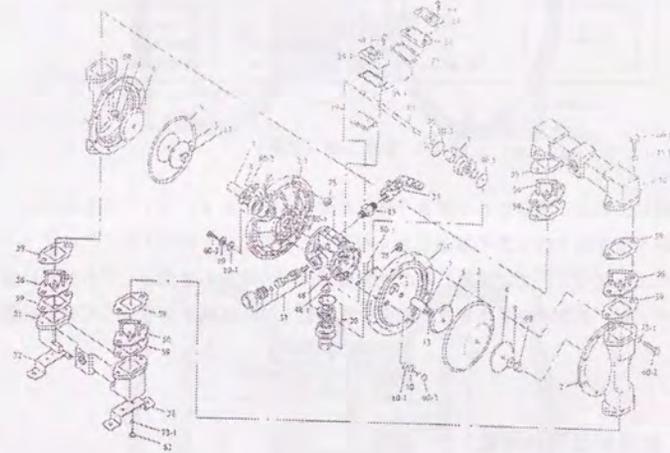


図 8.2: 製品の構造

このメーカーの製造設備の概略を図 8.3 に示す。第 1 工場では、旋盤、フライス盤、ボール盤などによる切削加工を行なう。また、第 2 工場では、より複雑な形状の部品や金型の製造を行なうために、マシニングセンターと NC を用いた切削加工を行なう。第 3 工場では、加工が終了した各部品の組み立て作業を行なう組立ショップ、一部の部品の塗装のための塗装ショップ、簡単な塑性加工ショップがあり、部品の間在庫と完成品在庫を持つ。さらに、以上の3つの工場で製造できない工程である、プレス加工、精密加工、成形加工、メッキ加工、研磨加工、熱処理加工などについては、外部の協力工場の利用が可能である。

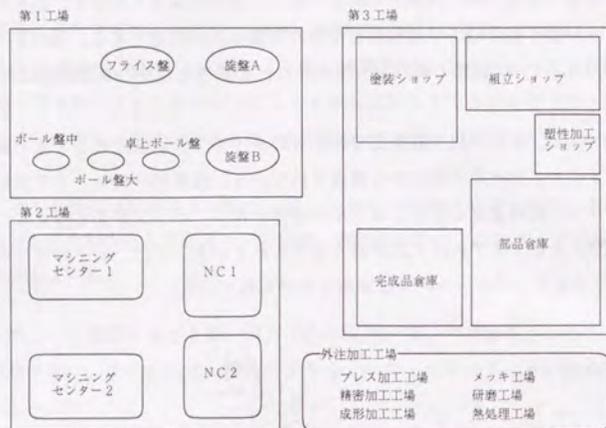


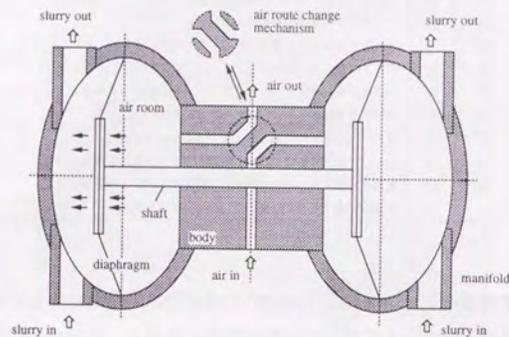
図 8.3: 製造工程 (設備)

最終的に、製品に期待される要求 (製品の機能) は、図 8.1 のような製品形状と、その形状を生み出す図 8.3 の製造設備を利用した製造工程を決定することによって生み出される。従って、先に説明したエア駆動ポンプの特徴を機能として具体的に実現するためには、具体的なさまざまな知識を総動員して問題解決を行なっていく必要がある。

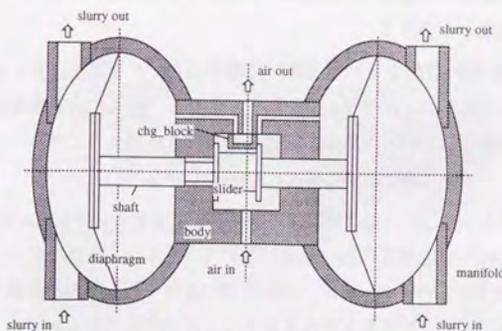
8.3 新規製品開発問題

製品開発を新規に行なうにあたって、実験では、すでに設計者の頭の中に、要求を満たすための形状と挙動が、ある程度まで漠然とできている状態から出発する。概念世界にある要求を具体化していく過程において、設計者は通常、図 8.4(a) のようなラフスケッチを書く。ただし、このラフスケッチに対応する問題構造とそれに至る過程は、通常、明示的に表現されない。実験ではまず、このような設計の初期段階での問題構造の明確化の過程を PICCSS によって表現してみることにする。

一般的に、製品開発の初期段階では、ことばをベースとして機能を展開していく。表 8.1 では、そもそもポンプという製品の持つべき機能からはじまり、徐々に機能が展開される様子を表している。設計の初期段階では、このような定性的な構造と、形状、重量、そして吐出量などのいくつかの定量的な仕様がおおまかに決定される。



(a) 製品のラフスケッチ



(b) 切替機構の開発

図 8.4: 製品開発の初期段階

表 8.1: 製品の機能センテンス

レベル	文番号	上位	機能センテンス
1	01	-	圧縮エアによってマニホールド入口の用液をマニホールド出口へ搬送する。
1	02	-	用液の流量を調節する。
2	03	01	ダイヤフラムによってエアの圧力を用液の流れに変換する。
3	04	03	ダイヤフラムの往運動で吐出圧を得る。
3	05	03	ダイヤフラムの復運動で吸入圧を得る。
2	06	01	ダイヤフラムを往復運動させる。
2	07	01	マニホールド内の用液の流れ方向を一方とする。
3	08	06	ダイヤフラムを左右におく。
3	09	06	圧縮エアを往運動をするダイヤフラムのエア室へ供給する。
3	10	06	復運動をするダイヤフラムのエア室のエアを排気する。
3	11	06	シャフトによって左右のダイヤフラムを接合する。
4	12	11	左のダイヤフラムの往運動を右のダイヤフラムの復運動とする。
4	13	11	右のダイヤフラムの往運動を左のダイヤフラムの復運動とする。
3	14	06	往運動が終了したら圧縮エアの供給先を切替える。

表 8.1 にある機能センテンスは、それぞれのセンテンスが上位のセンテンスに対しては手段、下位のセンテンスに対しては要求を表しており、これらの階層構造はそのまま問題構造に対応する。しかし、このままではことばのもつ曖昧性などが残っているため、使用する単語の整理や欠落情報の付加などをおこない、図 8.5、図 8.6 のような形で PICCSS に設定する。

図 8.5 は、表 8.1 の第 1 レベルである文番号 01, 02 をプリミティブの構造として表現したものであり、これが最上位の解候補となる。図 8.5 の問題構造を表 8.1 に従ってさらに精緻化していった結果、問題構造は図 8.6 のようになった。これは、図 8.4(a) のラフスケッチにおける問題構造に対応している。

実験ではこのようにして、製品の機能の展開を表す機能センテンスに従って、PICCSS に設定された当初の問題構造の該当部分に、プリミティブを追加していく形で徐々に問題構造を精緻化していった。特に、対象製品において最も重要な機能の 1 つであるエア切替機構は、ダイヤフラムを往復運動させるために必須のものであり、これについては、表 8.2 と図 8.7 に示す。

図 8.4(b) で示した切替機構に関するラフスケッチは、PICCSS によって図 8.7 のように表せられた問題構造の空間的な表現であるということが出来る。ラフスケッチにある形状情報からは、PICCSS の問題構造として表現された内容以上の情報（例えば液室が楕円形であるなど）が読みとれるが、それらは問題解決においてその時点では必然的なものではなく、便宜的なものである。従って、直観的な理解は困難であるものの、提案する知識の表現形式は、ラフスケッチによって表現される機能的な知識

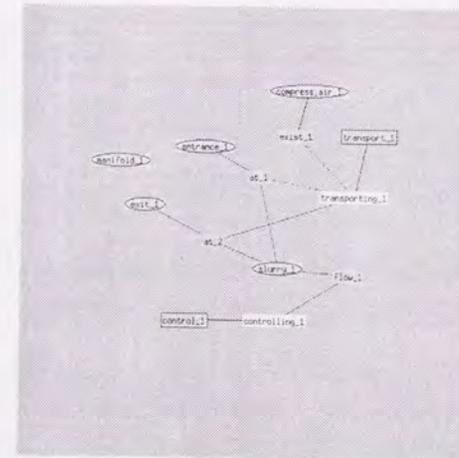
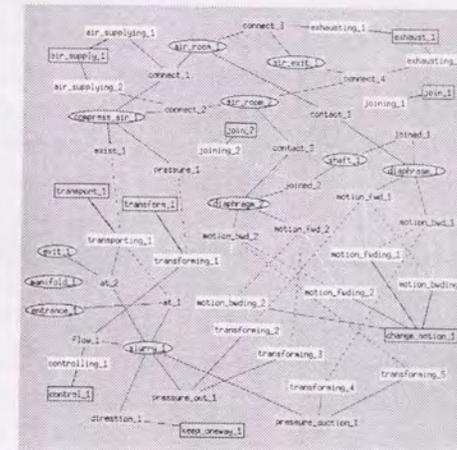


図 8.5: 初期段階の問題構造 1 (レベル 1 まで)



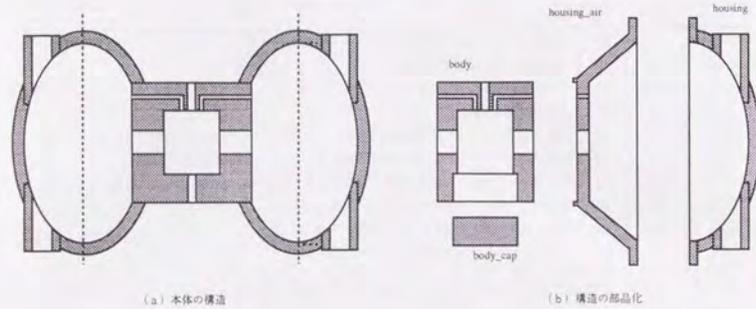
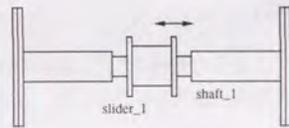


図 8.8: ポンプ本体の分割



(a) スライダーとシャフト部形状

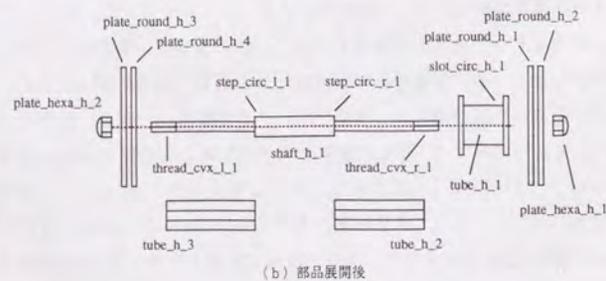


図 8.9: スライダーとシャフト部の分割

とは、製造可能なパターンをテンプレートとしていくつか用意し、それらのパターンに近い問題構造について、過去の知識を有効利用する方法である。また、ひとたび問題構造として設定された製造方法が、現実に適用可能であるかについて、制約処理によって厳密な検証を行なうことも、計算機支援の大きなメリットである。

本節では、製造を考慮した問題構造の精緻化の例として、図 8.4(b)におけるスライダーとシャフトの部分を取り上げる。この部分は、シャフトの中心部分が外側より径が小さくなっており、それを利用してスライダーがその範囲で左右にスライドする機構となっているが、このままではスライダーを組み立てることが不可能である。このスライダーとシャフト部分を、製造可能性を考慮しながら問題構造を精緻化すると、図 8.9のような部品に分割されることになるが、この過程における問題構造は PICCSS 上で、図 8.10のように表現することができる。

図 8.10 におけるスライダーとシャフトの問題構造は、個々の部品の形状と同時に、その形状を製造するための製造行為が割り当てられている。例えば、シャフト (shaft_h_1) の形状を設定する時点で、棒材 (mtr.iron_bar_1) を切断 (cut_off_bar_1) し、段の加工 (cut_step_circ_1, cus_step_circ_2) を行ない、さらにネジ加工 (cut_thread_1) という一連の工程が、リレーションとオペレータのコネクションを手がかりに設定されている。なお、図 8.10 では、図が繁雑になるのを防ぐために、プリミティブの中でコネクションを1つしか持たないものは表示されていない。また、図 8.11 に、この部分を含めた最終的な問題構造を示す。

この段階で、問題構造に含まれるプリミティブの数は、エンティティが 55、プロセスが 39、リレーションが 423、オペレータが 103 であり、合計 620 のプリミティブによって問題構造が表現できた。設定されたプリミティブをクラス数として見ると、エンティティが 35、プロセスが 20、リレーションが 86 そしてオペレータが 23 で、合計 164 であった。設定されたこれらのプリミティブのクラスの中で、前節においてあらかじめ登録されたものは 123 であり、実験で利用したプリミティブのクラスのうち 75.0% が、汎用的な知識によって記述可能であることが確認できた。

8.5 パラメータ計算

前節で得られた問題構造において、各パラメータの値を計算で求める。実験で用いたプリミティブは、製品の形状は 2次元空間上で表現しているため、ここで計算する位置パラメータは、x 方向と y 方向のいずれかとなる。また、時刻パラメータとしては、ポンプの挙動に関する時間の経過と、ポンプ製造に関する時間に関する 2つの時刻パラメータがあるが、ここでは、後者の製造に関する時刻について計算する。これ

以外のパラメータとしては、重量や用液の流量、圧力などいくつか存在するが、上記のパラメータの計算に関係しないものについては、今回の対象からは除外した。

問題構造をパラメータと制約からなるグラフ構造で表すと、x方向の位置パラメータとy方向の位置パラメータ、そして時刻パラメータは完全に分割される。従って、以下に示すように、制約充足計算では、位置パラメータについて、x方向、y方向についてそれぞれ独立して計算している。

製品形状の計算

製品を構成するすべての部品は、水平方向の位置としてx座標値をもっている。これらのパラメータの値の決定に関係する制約式の一部を列挙すると、図8.12のようになる。なお、これらの制約式は、計算機内部に自動的に設定され、この情報をもとに制約充足計算が実行される。

設計では、より上位の機能に関係するリレーションのパラメータに対して、要求として値が設定される。実験では、何度かの試行錯誤の結果、表8.3に示すような値を要求としてパラメータに設定した。また、表8.3以外のパラメータとして、表8.4に示すデフォルト値を設定した。表8.4にあるパラメータは、設計パラメータというよりは、常識的にある程度決められるものであり、関連する知識ベースを充実させることにより、いずれ入力省略できる。

以上の設定値に基づき、形状パラメータを計算した。パラメータの計算結果は、可視化によって、水平方向と垂直方向とに分け、それぞれを関連するエンティティごとにまとめて表示される。図8.13と図8.14は、スライダとシャフト部のそれぞれx方向とy方向の位置関係を表している。また、エア切替機構の部分については、図8.15と図8.16に示す。

このように、PICCSSによる形状の計算結果は、1次元表示であるが、この情報を用いて2次元あるいは3次元表示することは、現在のCADシステムの技術を利用すれば比較的容易であろう。図8.17に、本実験で計算した結果をもとに人間が作成したいくつかの構成部品に関する図面を示す。

工程プランの計算

以上の形状パラメータの計算においては、制約充足計算によって各パラメータの値を計算した。この結果が問題解決の解となるためには、それらのパラメータの値があらかじめ有効化されており、さらに計算されたパラメータの値となるように関連するオペレータの適用を行なう必要がある。ここでは、状態遷移計算によってこれらの処

```

-1*adjust_from_pface_l_1@adjust_from_pface_l_1 -1*step_circ_l_1@org_x_10 1*tube_h_1@pface_l_2
-1*adjust_from_pface_r_1@adjust_from_pface_r_1 -1*step_circ_obt_r_1@org_x_3
  1*thread_cvx_r_1@pface_r_3
-1*adjust_from_pface_r_2@adjust_from_pface_r_2 -1*step_circ_in_l_1@org_x_6 1*tube_h_1@pface_r_4
-1*adjust_from_pface_r_3@adjust_from_pface_r_3 -1*step_circ_r_2@org_x_9 1*tube_h_1@pface_r_4
-1*adjust_origin_h_1@adjust_origin_h_1 -1*step_circ_r_1@org_x_2 1*step_circ_obt_r_1@org_x_3
-1*align_pface_l_1@align_pface_l_1 1*tube_h_3@pface_l_8 -1*tube_h_1@pface_l_2
-1*align_pface_l_2@align_pface_l_2 1*tube_h_1@pface_l_2 -1*tube_h_2@pface_l_7
-1*align_pface_r_1@align_pface_r_1 1*tube_h_3@pface_r_9 -1*step_circ_in_h_1@pface_r_5
-1*depth_cedge_l_1@depth_cedge_l_1 -1*slot_circ_side_l_1@cedge_l_1 1*slot_circ_side_l_1@pface_l_5
-1*depth_cedge_r_1@depth_cedge_r_1 1*slot_circ_side_r_1@cedge_r_2 -1*slot_circ_side_r_1@pface_r_7
1*step_circ_in_l_1@pface_l_3 -1*slot_circ_in_h_1@pface_l_4
1*thread_cvx_r_1@pface_r_3 -1*cylinder_h_1@pface_r_1
-1*gap_face_h_1@gap_face_h_1 -1*slot_circ_in_h_1@pface_r_5 1*slot_circ_in_h_1@pface_l_4
1*tube_h_1@pface_l_2 -1*slot_circ_side_l_1@cedge_l_1
1*tube_h_1@center_h_5 -1*slot_circ_side_l_1@center_h_8
1*step_circ_r_2@pface_r_6 -1*slot_circ_side_r_1@cedge_r_2
1*step_circ_r_2@center_h_9 -1*slot_circ_side_r_1@center_h_11
1*tube_h_1@center_h_5 -1*step_circ_in_l_1@center_h_6
1*tube_h_1@center_h_5 -1*step_circ_l_1@center_h_10
1*step_circ_r_1@center_h_2 -1*step_circ_obt_r_1@center_h_3
1*cylinder_h_1@center_h_1 -1*step_circ_r_1@center_h_2
1*tube_h_1@center_h_5 -1*step_circ_r_2@center_h_9
1*step_circ_obt_r_1@center_h_3 -1*thread_cvx_r_1@center_h_4
-1*length_h_1@length_h_1 1*cylinder_h_1@pface_r_1 -1*cylinder_h_1@pface_l_1
-1*length_h_2@length_h_2 1*tube_h_1@pface_r_4 -1*tube_h_1@pface_l_2
-1*length_h_3@length_h_3 1*tube_h_2@pface_r_8 -1*tube_h_2@pface_l_7
-1*length_h_4@length_h_4 1*tube_h_3@pface_r_9 -1*tube_h_3@pface_l_8
-1*meet_face_r_1@meet_face_r_1 1*step_circ_r_1@pface_r_2 -1*tube_h_2@pface_l_7
-2*cylinder_h_1@center_v_1 1*cylinder_h_1@pface_r_1 1*cylinder_h_1@pface_l_1
1*step_circ_obt_r_1@org_x_3 -1*step_circ_obt_r_1@cedge_obt_r_1
1*cylinder_h_1@org_x_1 -1*cylinder_h_1@center_v_1
1*step_circ_in_l_1@org_x_6 -1*step_circ_in_l_1@pface_l_3
1*step_circ_l_1@org_x_10 -1*step_circ_l_1@pface_l_6
1*step_circ_r_1@org_x_2 -1*step_circ_r_1@pface_r_2
1*step_circ_r_2@org_x_9 -1*step_circ_r_2@pface_r_6
-1*slant_step_circ_r_1@slant_step_circ_r_1 1*step_circ_obt_r_1@cedge_ccv_obt_r_1
  -1*step_circ_obt_r_1@cedge_obt_r_1
-1*thread_length_r_1@thread_length_r_1 1*thread_cvx_r_1@pface_r_3 -1*thread_cvx_r_1@cedge_r_1

```

図 8.12: 水平位置に関する制約式の一部

表 8.3: 形状パラメータの要求

記号	内容	値
A	左右ダイヤフラム間隔	140
B	プレート直径	56
C	プレート厚	5
D	ストローク /2	12.5
E	本体部分幅	76
F	本体部分高さ	52
G	本体内部幅	46
H	エア経路径	3
I	吸気口間隔	12
J	シャフト径	10
K	ネジ径	6
L	切替ブロック高さ	5

表 8.4: 形状パラメータ省略値

記号	内容	値
M	シャフト溝深さ	3
N	スライダ厚	2
O	スライダ爪幅	3
P	スライダ溝深さ	4
Q	切替ブロック深さ	2
R	切替ブロック間隔	2
S	ネジ部長さ	12
T	ネジかみ合い	6
U	ナット径	12
V	ナット幅	10
W	ナット深さ	8

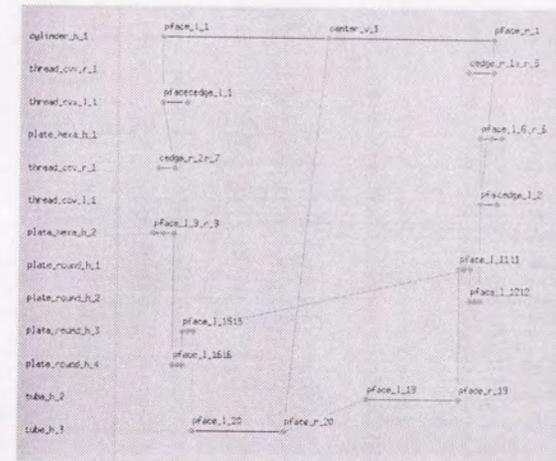


図 8.13: スライダーとシャフト部の計算結果 (水平方向)

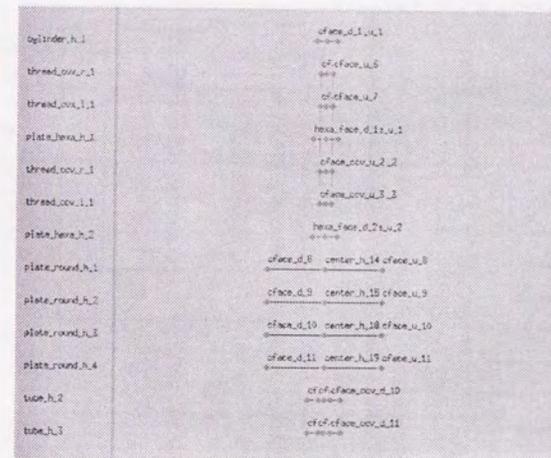


図 8.14: スライダーとシャフト部の計算結果 (垂直方向)

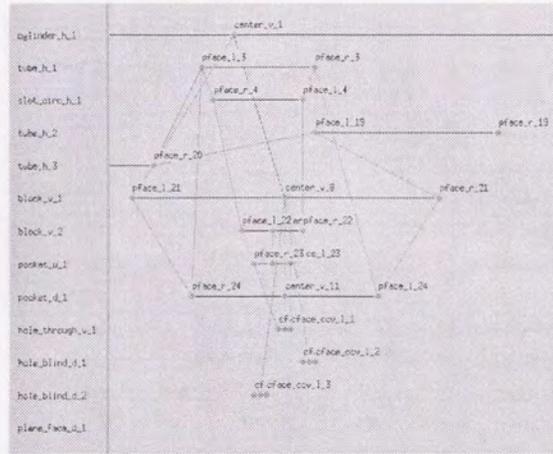


図 8.15: 切替機構の計算結果 (水平方向)

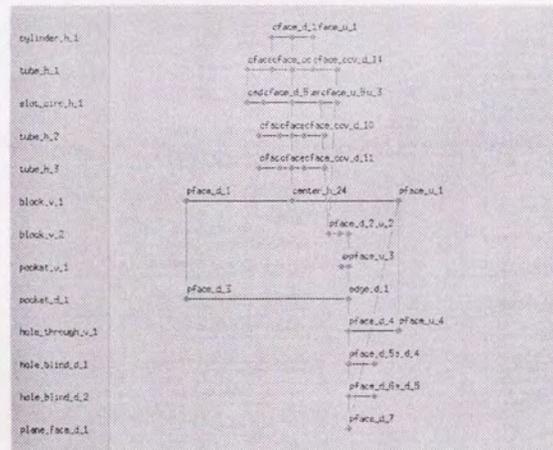


図 8.16: 切替機構の計算結果 (垂直方向)

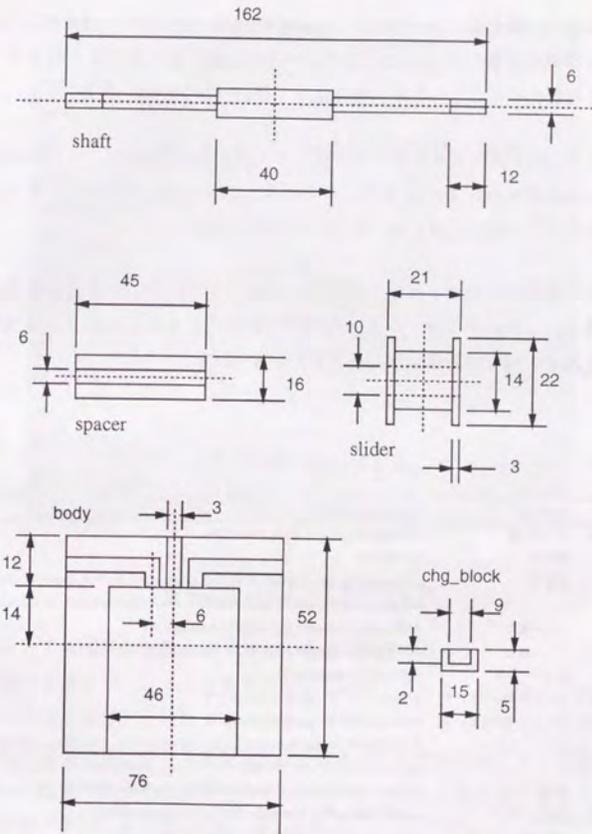


図 8.17: 実験結果 (部品図)

理を行ない、その結果として工程プランを得る。

本実験の場合、状態遷移計算において考慮すべき中心的なものは、以下の3つ種類のパラメータであり、これらのパラメータはあらかじめ有効でないため、それらを有効化することが必要となる。

- 組み立てに関するリレーション：insert や meet_face など、個々のエンティティ間の関係を定義しているようなリレーションは、あらかじめ有効でなく、それらを有効化することにより、組み立て工程のプランが得られる。
- エンティティ間の従属関係：エンティティ間の従属関係として、has_step や has_hole_through などが無効である場合、これらのリレーションを有効化することにより、加工に関する工程のプランが得られる。
- 上記2種類のリレーションに加えて、素材やツールなどがあらかじめ有効でない場合、これらのリレーションも併せて有効化することにより、製造に関するほとんどすべての工程を含んだプランが得られる。

表 8.5: 生産工数テーブル

factory	machine	operator/process
第1工場	スライス盤	milling_slot(12), milling_pface(10)
	旋盤 A	cut_off(10)
	旋盤 B	cut_pface(8), cut_cface_cvx(8), cut_cface_ccv(12), cut_chamfer_circ(6), cut_step_circ(7), cut_thread_cvx(15), cut_center_hole(4), cut_slot_circ(10)
	ボール盤大	drilling(4), chamfer_by_drill(2), tapping(15)
	ボール盤中	drilling(4), chamfer_by_drill(2), reaming(8), tapping(15)
第2工場	卓上ボール盤	drilling(4), reaming(8)
	マシニングセンター1	milling_slot(12), milling_pface(10)
	マシニングセンター2	milling_slot(12), milling_pface(10)
第3工場	NC1	cut_pface(5), cut_cface_cvx(5), cut_cface_ccv(8), cut_fillet_circ(10), cut_step_circ(4), cut_thread_cvx(12), cut_center_hole(3), cut_slot_circ(8)
	NC2	cut_step_circ(4), cut_thread_cvx(12), cut_center_hole(3), cut_slot_circ(8)
	組立ショップ	put(2), setup(2), insert(2), fix(5), fix_by_thread(5)
	塑性加工ショップ	press_cutting(2), flattening(2), cutting(5)
	塗装ショップ	painting(30)
外注工場	プレス加工工場	press_cutting(10), press_bending(20), press_forming(20)
	メッキ工場	electric_plating(50)
	精密加工工場	lapping(60)
	研削工場	grinding(10), grinding_pface(10)
	成形工場	die_casting(150), lost_waxing(90), injection_molding(90), sintering(120)
	熱処理工場	heat_treatment(90)

一方、知識ベースには、製造工程として、7章で説明したようなプリミティブがあらかじめ定義されている。実験では、それらのプリミティブに対して、事実情報として、処理時間を表 8.5 のように設定した。表 8.5 の括弧内の数字が、処理時間 (分) を表す。

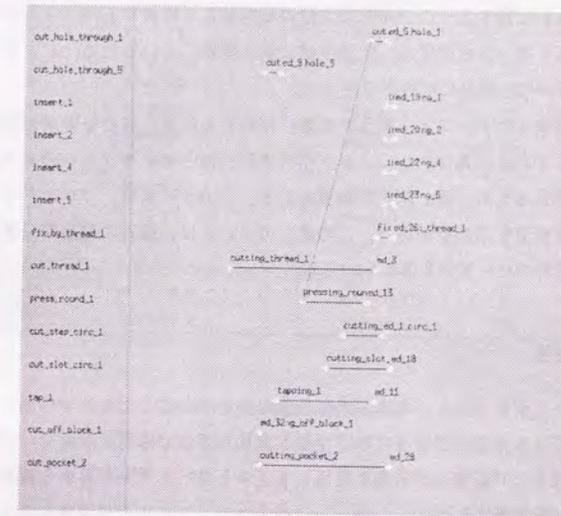


図 8.18: 工程プランの計算結果

計算された工程プランを図 8.18 に示す。ここで計算されたスケジュールは、定常的な生産におけるスケジュールとは多少異なっているが、これは、製品開発における試作品の生産や初物生産などに対応するスケジュールということができる。ただし、製品開発と製造の関係が現在のような期間的に分かれている形態から、さらに多品種少量生産が進み、設計の大半を注文を受けてから行なうような生産形態になると、ここで計算した工程プランにより近い形態となるだろう。

8.6 設計変更問題への適用

設計変更とは、すでに作成した問題構造を対象として行なう修正ベースの問題解決であるということができる。ただし、通常このような設計変更では、一部分の変更が関連するさまざまな部分に影響を与えるため、それらの影響の結果が整合性を保った

状態となっているかを、きちんと管理する必要がある。いなければならない。特に、精緻化のサイクルにおけるより上位の問題構造の変更やパラメータの修正は、非常に多くの影響を他に与えることが予想される。

ここでは、設計変更の2つのケースについて実験を行なった。最初のケースは、ポンプの基本的なスペックであるダイヤフラムの往復運動のストロークの変更である。これは、図 8.4 に示すように、非常に上位の問題構造を構成するパラメータの一部である。このストロークの変更は、切替機構に深く関係しているため、どの寸法をどれだけ変更するかは、簡単には決定できない。

設計変更の第2のケースは、製品を実際に使用した結果、新たな要求が追加される場合を想定している。具体的には、ポンプの本体内部とダイヤフラムのエア室との間でエア漏れがあるため、漏れ防止の処置をする。このケースは、プリミティブからなる問題構造を変更する必要がある。この第2のケースは、既存の製品開発事例をもとにした改良設計のケースであるともいえる。

ストローク変更

ストロークを変更すると、さまざまな形状の変更が必要となる。どの形状パラメータが関係しているかは、部品図の単位で図面を見ただけでは判断が難しい。先の実験ですでに生成した問題構造の結果を利用して、ストローク変更の要求を新たに加えることにより、問題解決を行なう。

変更要求は、パラメータ `stroke.length.1` に新たな値として 20 を設定し、制約処理を再実行する。この際、制約伝播に関係したパラメータを列挙すると、(0)adjust_center.h.5, (1)mirror_face.3v.1, (1)meet_face.r.5, (2)ge.6, (2)adust_center.h.1, (3)adjust_center.h.4, (3)length.2h.6, (4)le.1, (4)length.h.11, (5)length.2h.1, (6)meet_face.l.8, (7)ge.1, (8)diam.in.v.1, (9)mirror_face.v.10, (10)mirror_face.v.11, (11)gap_face.h.2, (12)diam.in.v.1, (13)mirror_face.v.9, (13)gap_face.2h.1, (14)length.2h.8, (14)length.2h.9, (14)length.h.12, (15)gap_face.2h.3, (16)ge.9, (17)thick.h.3, (18)mirror_face.v.1, (19)has_thread.ccv.r.1, (20)has_thread.ccv.l.1, (21)has_hole.through.h.1, (22)has_hole.through.h.2, (23)thick.h.4, (24)thick.h.6, (25)align.pface.l.1, (26)align.pface.r.1, (27)meet_face.l.3, (28)meet_face.l.4, (29)meet_face.r.3, (30)meet_face.r.4, (31)meet_face.r.6, (32)meet_face.l.6, (33)length.2h.7 の合計 40 個のプリミティブとなった。ここで、括弧内の数字は、伝播の深さを表す。

この制約伝播の結果、各パラメータの値が決定された。変更前と変更後のパラメー

タの値を表 8.6 に示す。

表 8.6: ストローク変更の結果

記号	内容	プリミティブ名	変更前	変更後
A	左右ダイヤフラム間隔	length.2h.7	140	145
B	プレート直径	diam.h.8	56	56
C	プレート厚	thick.h.3	5	5
D	ストローク /2	adjust_center.h.5	12.5	20
E	本体部分幅	length.h.11	76	95
F	本体部分高さ	length.v.1	52	52
G	本体内部幅	gap_face.h.3	46	46
H	エア経路径	diam.in.v.1	3	3
I	吸気口間隔	adjust_center.h.2	12	12
J	シャフト径	diam.h.1	10	10
K	ネジ径	diam.h.6	6	6
L	切替ブロック高さ	length.v.2	5	5
-	シャフト溝幅	gap_face.2h.1	40	55
-	シャフト長さ	length.h.1	162	167

ストロークの変更という要求に対する設計変更の結果は、このように、スパーサー、スライダ、シャフト、そして本体内部の寸法を変更する必要があるという情報を提示した。なお、ストロークをどこまで大きくすることができるか、といった問題については、その値を直接求めることはできなかったが、ストロークの値を何度か変更して実験を繰り返すことにより、おおよその見当はつけることができる。本ケースの場合は、ストロークを大きくする場合は、ポンプ本体の幅に直接伝播するため、外形寸法の制約がこのストロークの最大を決定していることが分かった。

エア漏れ防止

次に、新たな要求を追加するケースについて、提案する問題解決の枠組の有効性を検証する。製品を使用した後に、圧縮エアの圧力に対して用液の吐出圧が上がらないという症状がみられ、原因解析の結果、ポンプの本体からのエア漏れが原因であることが分かったような状況を想定する。

エア漏れ防止を施す場合は、形状のパラメータを変更するだけでは対応しきれず、問題構造に新たな知識を追加し構造を変更することが必要となる。一般に考えられるエア漏れ防止のための方法として、知識ベースに登録されたいくつかの知識を用いて、問題構造をさらに精緻化していくことが、この場合の設計変更に対応する。つまりこのような設計変更は、当初の問題構造に対して、漏れ防止という新たな機能を追加す

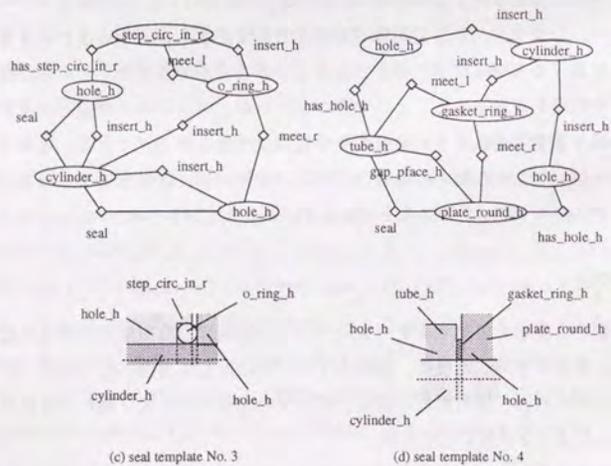
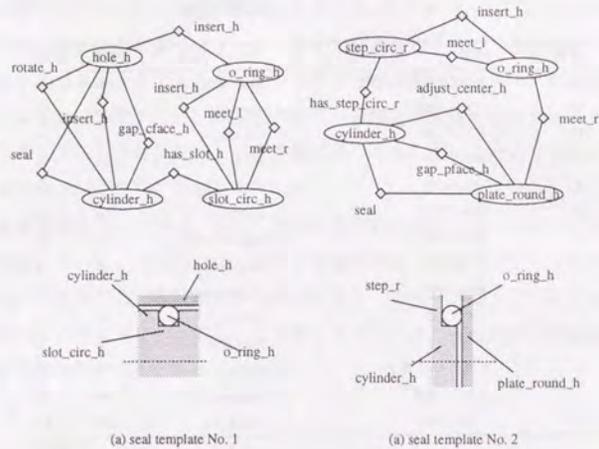


図 8.19: 漏れ防止に対応するテンプレート

ることによる問題の精緻化を行なうことに相当する。

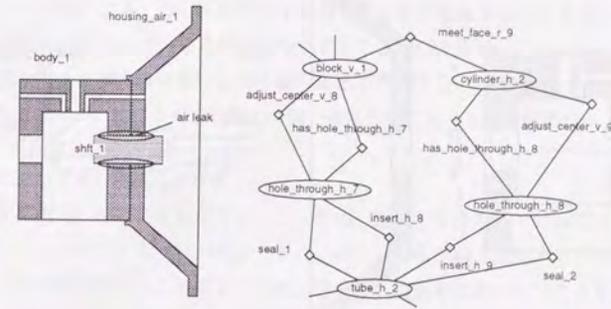


図 8.20: 漏れ防止の追加要求

一般に、シールのための機能部品としてOリング、ガスケット、Uパッキン、Vパッキン、ダストシール、オイルシール、そしてメカニカルシールなどがある。2つの部品が固定されている場合には、Oリングやガスケットが用いられ、回転や滑りなど可動部分については、比較的圧力が低い場合には、Oリング、Uパッキン、Vパッキンが使用され、高圧の場合はオイルシールやメカニカルシールが利用される。

本研究の知識表現の上では、例えばOリングやガスケットはエンティティとなり、漏れ防止という機能を表現するためには、図8.20のようなテンプレートを用いて、いくつかのプリミティブの関係によって表現される。図8.20のように、漏れ防止という機能は、適用される周辺の形状に応じて適用可能な複数のテンプレートがあり、これらの候補の中から問題解決の過程で選択される。図8.20は、円筒面である cface をもっている場合のいくつかの例であり、Oリングが用いられている。接触面が円筒でなく平面であるような場合には、漏れ防止に対応する別のテンプレートとして例えばガスケットを用いた構造など候補となる。

実験で取り上げたケースでは、図?? のように、エア漏れはシャフトと本体との接触部分でおこるので、シャフトの一部である tube.h.2 とボディの穴にあたる hole_through.h.7 と hole_through.h.8 それぞれのプリミティブの間にもまず seal.1, seal.2 というリレーションを挿入する。このリレーションは、漏れ防止という機能を表すリレーションであり、このプリミティブの存在を手がかりとしてテンプレートが検索される。結果として、図8.20(c)のような構造がシャフトと本体との間に挿入され、製品は図8.21のような形状となる。

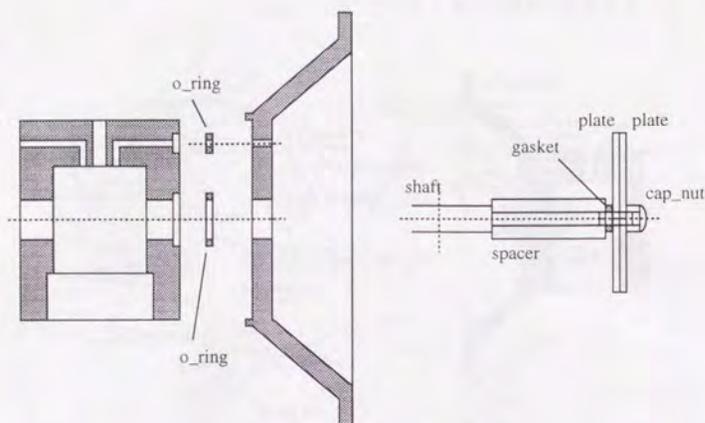


図 8.21: 漏れ防止の結果の製品構造

8.7 評価と考察

以上の実験を通して、提案する問題解決の枠組みおよび支援ツールに関して、以下のような知見が得られた。

- トップダウンな問題解決支援に有効

本研究で提案する問題解決の枠組みによって、トップダウンな製品開発の過程が表現でき、特に開発した支援ツール PICCSS は、製品の機能を中心に問題を精緻化していく上で十分に有効な支援ツールであることがいえた。通常の機械設計では、部品図の形状を決定してから全体を再構成するというボトムアップアプローチがとられるが、提案する方法では、完全にトップダウンに行なうことが可能であった。

- 知識の効率的な再利用が可能

知識の再利用という点で、提案する知識の表現形式は非常に有効であった。これは、プリミティブによって製品開発における基本的な語彙を表現することによって、実際の製品開発の段階では非常に少ない労力で問題構造の設定が可能であると同時に、そこで利用されたプリミティブの大半は汎用的な知識であることから確かめられた。

- 設計パラメータの試行錯誤のツールとして有効

問題構造を設定し、いくつかの設計パラメータを設定することにより、詳細なレベルのパラメータを制約関係を利用して自動計算ができた。決定パラメータは従来の方法に比べて格段に少なくすることができたばかりか、これによって設計パラメータの決定における試行錯誤が容易となり、より質の高い設計が可能となった。特にこの点に関しては、設計変更における効果が著しかった。

- 製造を考慮した設計が可能

製品開発の中で製品形状を決定する際に、提案する枠組みでは製造可能性をある程度考慮できることが確認できた。また、製品形状を設定する時点で、製造工程を割り当てることにより、製品開発の上流過程においてラフな工程プランが自動生成できることが確認できた。

また、実験を通して次のような今後の検討課題が得られた。まず第1に、本研究の制約処理では、まだ詳細なレベルの製造可能性の検証と工程プランの自動生成はできない。例えば、金型などは、非常に複雑な形状をもっており、幾何的なさまざまな制約を表現できる必要がある。また、工程プランを生成するためには、探索問題に対する性能が直接影響するが、現在のところ開発した支援ツールでは計算効率その他の面に対応しきれていない。

第2に、問題解決の過程そのものを陽に表現することができないという点が挙げられる。本研究の枠組みでは、知識はすべて問題構造の中に結果として埋め込まれ、そのような問題構造に至った過程は直接表現されない。知識の再利用を考える場合には、過去に行なった問題明確化の過程を参照することがなんらかの形で有効なことがあるが、現在のところその点は未検討である。

第3に、基本的なアイデアの生成や発想の支援のための考慮はないという点が挙げられる。特に製品開発の上流においてはアイデアをいかに生成するかや発想をいかに製品に結び付けるかといったことが重要であるが、この点に関する研究は現在過渡期であり、提案する支援システムではこれらに対する明確な方法を提供していない。今後の課題とした。

以下に、本実験を通して得られた提案する問題解決の枠組みの有効性について述べる。なお、本研究の枠組みの有効性を客観的な指標によって実証することは困難であるため、ここでは、実験で対象とした問題の一般性と、その問題における提案する手法の有効性によって、評価することとする。

評価：事例の一般性

まず、実験で取り上げた事例がどの程度一般的であるかについて考える。対象とした事例における製品は、典型的な加工組立による製品であり、多くの機械製品はこのような生産形態によって生産される。実際に、機械設計の教科書 [畑村 88]、[畑村 92] や、加工技術に関する教科書 [萱場 95] およびハンドブック [JIS93a]、[JIS93b] から得られた情報をプリミティブとして登録し、実験ではその内容を広く問題明確化の中で利用した。このことから、この事例の持つ一般性がいえる。

ただし、問題構造の複雑さの点からいえば、ここで取り上げた事例は比較的簡単な製品に分類できる。従って、空間的な制約が厳しい、あるいは金型の加工など製造方法がより複雑、あるいは最適設計をより追求したような状況に関する本枠組の適用の有効性については、この事例だけでは説得力がない。今後、より複雑な製品開発の事例によって、有効性の検証を行なう必要がある。

また、製品開発の中でも、機械設計以外の分野、例えば航空機の設計や食品や化学製品などに関する設計については、実験で取り上げた事例とは製品開発過程が大きく異なる。そのような分野についての適用可能性も直接実証することはできないため、今後の課題とした。

ただし、ここで取り上げた事例によって、提案する問題解決の枠組とそこでの具体的な方法を明らかにすることができた。これによって、それぞれ個々に傾向の異なる製品開発についても、どの程度の知識をプリミティブとして用意すれば有効な支援が可能となるかを、ある程度まで予想することができる。

評価：事例における有効性

本章で取り上げた事例について、手法の有効性を検討する。本事例は機械製品における加工組立型の製品としては非常に一般的であるため、この事例における有効性が確認できれば、一般的な問題においてもほぼ同程度の効果が期待できであろう。

提案する手法の評価項目を、表 8.7 のように、問題記述の容易性、表示情報の可読性、定義知識の再利用性、計算結果の妥当性、そして支援範囲の有効性という 5 つの項目に分けた。まず、問題記述の容易性については、本研究の知識表現の枠組は、現状ではプリミティブを直接扱って問題構造を設定していくため、プリミティブそのものの内容を知らない問題の表現が困難である。プリミティブの名称だけで、その内容まで知らなくともある程度の問題の記述が可能となるようにさらに柔軟なユーザーインターフェースが望まれる。

次に、表示情報の可読性であるが、これはあまりよくないといえる。問題構造は、

表 8.7: システムの評価

評価項目	評価
問題記述の容易性	△
表示情報の可読性	×
定義知識の再利用性	○
計算結果の妥当性	◎
支援範囲の有効性	○

プリミティブからなるネットワークによって表示されるが、この構造を設計者がそのまま見ても、それが意味する内容を理解できない場合がある。特に問題が複雑になると、コネクションが錯綜し、可読性はさらに悪くなる。従って、現在のチャート表示に加えて、さらにこの問題構造をさまざまな形で加工して表示するような仕組みが必要である。

続いて、定義知識の再利用性については、これはよいといえる。製品開発の初期段階では、機能センテンスにさまざまなことばが含まれ、それらに対応するプリミティブが存在しないことが多いが、問題構造を精緻化するにつれて利用するプリミティブの知識ベースに登録されている割合は増え、最終的な形状や工程の段階に至っては、ほとんどのプリミティブが知識ベースに登録されてあるもので対応できた。なお、知識ベースに登録したプリミティブは、教科書的な内容に限定してある。

計算結果の妥当性については、非常に優れている。計算機を利用したため計算結果が正確であることはいうまでもないが、そのためにはきちんと必要な情報が計算機が理解できる形式で設定されていなければならない。特に、制約関係を中心とした問題構造の内部モデルを用いて計算した製品形状と工程プランは、現実の製品開発の出力としてもある程度利用可能なレベルであった。

最後に支援範囲の有効性であるが、従来の知的 CAD への期待の中には設計の自動化が大きなウエイトを占めていたが、ここでの計算機の利用ではそのような自動化は目指さず、あくまで支援の立場をとっている。従って、結局のところ人間の設定した知識に従って製品開発が進められるという、いたって当たり前の範囲を脱することはできない。ただし、計算機が、従来は設計結果のドラフティング、あるいはパラメータ情報の整理、あるいはシミュレーションといった範囲から、さらに一歩踏み込んだ領域といえる、機能の展開から具体的な形状や方法に至るトップダウンな製品開発の過程をその内部で扱い、設計者が計算機の上で思考を展開できるような環境を作って

いるという点は重要である。このように製品開発における問題解決を計算機の内部で行なうことにより、過去の設計知識の有効利用やさまざまな制約処理を直接計算機に行なわせ、人間はより高次元の処理を行なうという役割分担が可能となる。

8.8 8章のまとめ

本章では、実際の製品開発の事例に対して、本研究の問題解決の枠組みを適用することで、提案する方法の有効性を実証した。ここで取り上げた事例は、エア駆動ポンプであり、典型的な加工組立型の製品である。実験ではまず、新規設計における機能のトップダウンな精緻化を行なう過程を計算機上で表現し、それを支援することが可能であることを確認した。特に、機能を精緻化する場合に製造可能性が問題となったが、これを計算機上で形状と工程の対応をとることによって十分に考慮しながら問題解決を進めることができ、提案する方法の有効性を示した。また、形状パラメータや工程プランを自動計算した結果を示し、計算機支援の具体的な範囲を明確に示した。また、新規設計以外にも、改良設計や設計変更における支援についても実験をおこなった。最後に、以上の実験の結果をふまえ、支援システムの評価と、提案する問題解決の方法の有効性について考察した。

第9章

関連研究

9.1 はじめに

本研究の対象とする範囲は、従来の研究領域でいうと非常に多くの領域をまたがっている。応用の視点から見た場合、知的 CAD の研究、工程設計支援の研究、スケジューリングの研究、そしてコンカレント・エンジニアリングなどの研究分野が関連する。また、工学的な手法の視点からすると、問題解決やプランニングの研究、知識獲得や要求定義に関する研究、制約充足問題や組合せ最適化問題の研究、そして発想支援や知識体系化に関する研究も関連する。

本章では、応用の視点から見た場合の関連研究の現状をサーベイする。特に問題解決の手段として本研究と似たアプローチをとっている研究を紹介し、同時に本研究との違いを明らかにする。まず次節では、設計支援に関するものとして知的 CAD に関する研究を、3節では、工程設計に関する研究を、そして4節では、スケジューリングに関する研究を取り上げる。また、5節は、それらの各問題を統合的に行なうアプローチである CIM (Computer Integrated Manufacturing) やコンカレント・エンジニアリングに関する研究を取り上げる。

9.2 知的 CAD に関する研究

設計の自動化あるいは設計支援の研究は、大きく設計対象に関する研究と設計過程に関する研究に分けられる。設計対象に関する研究では、CSG (Constructive Solid Geometry) や B-Reps (Boundary Representation) に代表される幾何形状の表現 [Mantyla88] に加え、形状特徴の表現 [Salomons93], [Shah91] によるプロダクトモデルの研究 [Laakko93], [Martino94] によって、幾何制約の取り扱いなど、より高度な支援が可能となった [鈴木92], [Ando89], [清水94]。

また、設計対象を単に幾何形状としてだけでなく機能としてとらえ、それらを計算機上で表現しようとする試みが最近さかんである [Tomiyama93], [桐山 95], [Kimura91], [Gui94], [安藤 90]。これらの研究では、形状特徴や部品などの要素の組合せをパターンや制約などによって記述し、要素だけでは表現できない機能というものの表現を行っている。

設計を計算機によって支援する際に、特に設計者の持つ意図や設計ノウハウなどに着目し、それらを表現する言語あるいはツールを用意することで設計行為を支援しようという試みは、設計工学や知識工学の分野で古くから行なわれてきた [市田 81], [溝口 92]。特に知識工学の研究分野では、得てしてアドホックになりがちな設計知識を、いかに共通の枠組の中で扱うかに焦点をあてた研究が行なわれている [Sriram89], [服部 94], [岸 95]。特に、設計過程にはさまざまな側面があり [Finger89a], それらを形式的に扱おうとするいくつかの試みがとられている [吉川 87], [Suh90]。

知的 CAD に関する研究で特に近年の流れとして、設計対象である製品をそのライフサイクル全体を通してとらえた設計を支援することが望まれている点が挙げられる [Finger89b]。例えば故障に対する修復、部品の交換、さらに廃棄における分解可能性などを考慮した設計のための研究が増えつつある。例えば、自己修復型の機械に関する研究 [梅田 92] や、リサイクルを前提とした設計 [松下 93] が行なわれている。また、設計のより上流にあたるコンセプト生成や発想支援 [Hori94] なども徐々に知的 CAD 研究の対象となりつつある。

本研究では、モノ作りを問題解決ととらえているが、その問題構造の中で特に空間的な構造に着目すると、知的 CAD の研究と共通する部分が多い。特に、知識の表現形式や制約処理の部分は、従来の知的 CAD 研究の中に似たアプローチが多くある。しかし、本研究の従来の知的 CAD との根本的な違いは、従来の知的 CAD のほとんどが、形状を非常に厳密に扱い、形状をベースとしてモデルを組み立てているのに対し、本研究では、形状はあくまで最終結果あるいはユーザとの問題構造の確認に用いるインターフェースとして位置付けている点である。つまり、本研究における問題解決では、形状は必須ではなく、従来の知的 CAD の研究の対象からすると、より上位の問題を扱っているといえる。

9.3 工程設計に関する研究

工程設計あるいは CAPP (Computer Aided Process Planning) に関する研究は、さまざまな工程パターンをあらかじめ登録し、製品の種類に応じてそれらを検索して利用する variant 方式と、製品の形状その他のプロダクトモデルを利用して、その都

度に対応する工程を生成する generative 方式あるいは semi-generative 方式の2つに大別できる [Alting89], [Elmagraghy93]。前者の variant 方式は、すでに多くの企業で何らかの形で実用化されており [Eversheim93]、現在の研究動向としては、後者の generative または semi-generative 方式が中心である。

まず generative 方式のものとしては、特に部品の部分的な形状特徴に対して加工作業の手順の自動生成 [乾 88], [Marefat92] や、NC の加工経路の決定 [松田 89], [倉賀野 89] を行なう研究がある。またアセンブリにおける組立計画を設計者のメンタルモデルの情報を利用して自動生成するシステム [伊藤 90], [伊藤 91] などがある。

また、semi-generative 方式といえるものには、FMS における工程計画とその運用を考慮した研究 [岩田 88] や、過去の事例を積極的に用いる方法 [Tsatsoulis88], [Marefat93] などがあり、これらを含めて、実用となりつつあるさまざまなシステム [Houten91], [Tonshoff89] が発表されている。特に、注目すべき傾向としては、装置のコントロールやスケジューリングと統合しようという試み [Iwata90], [Tonshoff90] や、生産現場の不確実性に対応しようという試み [Detand90] などが挙げられる。また、知識工学の分野で古くから行なわれているプランニングの研究 [Fikes71], [Sacerdoti74], [Carbonell86] と結び付けようという斬新な研究 [Nau95] なども注目に値する。

工程設計を自動化あるいは省力化しようというアプローチ以上に、近年は製造そのものを容易にし、工程設計の時点で製造不可能であるといったことがないように、製造を考慮した設計 (Design for Manufacture) に関する研究がさかんである [Boothroyd94], [Sodhi94]。これらの研究は、製品を形状特徴や製造特徴などからなるさまざまな表現方法によって記述し [Kimura90], [松田 88]、成形性を考慮した金型設計 [Dighe93] や、組立て順序の自動生成 [Ko87], [Chen94] を実現している。特に、組み立てにおける部品構成の検討 [Mantyla94] や、設計過程をトップダウンに行なうアプローチの研究 [Hsu93] などがある。

まず、工程設計の研究に関しては、従来の工程設計の研究が、非常に部分的な形状に対する部分的な工程の自動計算を目指しているのに対して、本研究では、自動ではないが、加工や組立てさらには搬送などの間接的な作業も含む、全体的な工程の流れを対象としている点で異なる。つまり、本研究では、製品の機能に関係する行為がすべて対象となり、従来の形状からのボトムアップ的、あるいは演繹的な工程設計とは大きく異なる。これは従来の variant 方式とも異なり、工程設計に対する新しいアプローチといえる。

本研究のアプローチは、最終的な目的は多少異なるが、どちらかというところ、製造を考慮した設計の研究アプローチに近い。ただし、現在のところ、この研究領域では、個々の問題に対してはいくつかの成果が報告されているが、より一般性をもった方法

論あるいは手法は存在しない。従って、本研究は、この分野における先進的な研究であるといえる。

9.4 スケジューリングに関する研究

スケジューリングに関する研究では、従来のジョブショップ問題の最適解を求めるようなアプローチが現実的に困難であるため [茨木 94]、さまざまな準最適解をもとめる手法が提案されている [小林 92]。これらの研究は、大きく OR 的アプローチ、AI 的アプローチ、そしてマンマシンシステム・アプローチに分類できる [黒田 95]。OR 的アプローチでは、対象を基本的に組合せ最適化問題として定式化するが、その解法においてさまざまなヒューリスティクスを用いる。それに対して、AI 的アプローチでは、特定の問題領域の知識をできるだけ計算機上に表現し、それらを用いて解を求める。さらにマンマシンシステム・アプローチでは、計算機にすべての情報を表現せずに、ユーザや他のシステムなどとのインタラクションを通じて解が求められる。

近年特に、さまざまな探索手法の普及と、計算機能力の向上により、スケジューリングの研究は、従来のような理論研究から現実問題を多く扱う研究に変わりつつあり [Zweben94], [Sche95]、それにともなって実用のサイドからさまざまな期待が寄せられている [松本 94], [福田 93]。その中でも特に、設計変更や資材の遅れや装置トラブルなど、さまざまな製造現場の変化に対応して、ダイナミックに計画を修正する状況対応型のスケジューリングが望まれている [西岡 94]。

スケジュール修正問題に関する従来の研究として、Ow, Smith and Thiriez [Ow88] や Collinot, Pape and Pinoteau [Collinot88], そして Miyashita and Sycara [Miyashita94] などが、制約中心の AI 的アプローチによってこの問題を扱っている。また、OR 的な手法でスケジュール修正問題を扱ったものとして、ジョブショップ問題に対する Watatani and Fujii [Watatani92] や Li, Shyu and Adiga [Li93] などがある。前者のアプローチでは、ルールや事例によって修正戦略（ヒューリスティクス）を選択しており、解の最適性を直接考慮したものではない。一方、後者のアプローチでは目的関数が明確ではあるが、逆に生産現場の実際的な制約を表現することが困難となっている。

本論文の中では、スケジューリング問題をあまり強調していないが、スケジューリング問題は悪構造の制約充足問題として、本研究の枠組みの中でそのまま扱うことができる。例えば、上記のような変化する環境に対応して計画修正を行なうようなより難しいクラスの問題に対しても、5章で示したように本研究の制約処理の中で対応できる。さらに、本研究のように、製造業が全体の問題解決の枠組みの中で個々のスケ

ジューリング問題を位置付けることにより、スケジューリングと工程計画、あるいはスケジューリングと製品設計の統合が容易に可能である。現在のところ、このような統合的な問題解決の研究としては、実際に計算機上にインプリメントし実問題の扱いが可能であるレベルのものは、本研究以外には存在しない。

9.5 コンカレント・エンジニアリングに関する研究

製造業におけるさまざまな情報を統合化し、製品設計、工程計画、スケジューリングなど個々の問題解決を、孤立することなく全体として最適なものとしようという試みが、CIM という形で以前より行なわれてきた。特に近年は、製造という行為が1つの企業内にとどまらない状況となっており、よりオープンな環境における統合化に関する研究 [ESPRIT93] が進められている。また、この流れに沿って、設計や製造に関するさまざまな情報を標準化する動きが活発であり [鈴木 93]、特に自動車業界では具体的な標準化が行なわれつつある [ISO95], [長坂 94]。

これらの情勢は、設計や製造の方法そのものを変える可能性を秘めており、特に仮想生産 [Kimura94] は、次世代の生産システムへ向けての国際プロジェクトである IMS (Intelligent Manufacturing System) の主要な研究課題として取り上げられている。この仮想生産を実現するためには、生産対象である製品はもちろんのこと、工場モデルなどさまざまな生産環境とそこでの製造プロセス、そして、それらの情報を対象として行なう設計過程など問題行為そのものをモデル化する必要がある [Kimura93], [Onosato93]。

一方、このような仮想生産のための研究は、具体的にはたとえばコンカレント・エンジニアリングのような技術として製造業の中で利用されていくことが期待できる [木村 93]。コンカレント・エンジニアリングとは、従来のような厳密な分業による逐次実行方式から、より柔軟な仕事の区分に基づく並列実行方式へ変換しようというものである [Parsaei93], [Reddy93]。特に、従来は人と人との柔軟なコミュニケーションにたよっていたものを、計算機支援方式により画期的に合理化しようというねらいがある。

コンカレントエンジニアリングに関する研究としては、特に製造側の情報を設計に伝えるためのインタフェースに関する研究 [Houten92], [Gaddam93] や、製品開発に関する実際の異なるシステム間を、仲介的なプロトコルを介してコミュニケーションしようという実験的な試み [Cutkosky93]、そして、異なる機能や視点をもつ複数の構成員がコラボレーションを行なうための計算機環境に関する研究 [Wong93] などがある。

以上の研究は、生産に関するさまざまな情報の表現形式に関する研究、そしてそれ

らをもちいたコミュニケーションに関する研究、あるいはより情報工学的な研究であるといえる。本研究では、生産という行為に関する知識をいかに効率的に獲得し再利用するかを強く意識し、知識表現のための基本的な形式を提案している点で、これらの研究の中でも位置付けられる。また、設計と計画という本質的に異なる知識を統一的に扱っている点でも、例えばコンカレント・エンジニアリングの研究としても十分に新規性がある。ただ、現在のところ、本研究では、問題解決者として複数の主体を想定しておらず、従って、コミュニケーションに関する研究にはあたらない。

9.6 9章のまとめ

本章では、本研究で提案する問題解決支援の方法と関連する研究について実用的な視点からサーベイを行なった結果を報告した。最初に知的 CAD の研究としていくつかの研究を紹介した。特に知的 CAD の研究で扱うプロダクトモデルは本研究の形状表現と密接に関係している。また、工程設計やスケジューリングに関する研究について概観し、工程の抜いかたや最適化へのアプローチについて比較した。製造を考慮した設計というカテゴリーで本研究と類似したさまざまな研究が見られるが、一般性と実用性を合わせ持っている点で本研究と同等の研究は存在しなかった。またこの他に、コンカレント・エンジニアリングの研究を仮想生産の研究なども関連させながらサーベイした。以上のサーベイを通して、本研究の新規性が十分にいえることが確認できた。

第 10 章

結論

10.1 はじめに

本論文では、これまで、モノ作りを人間の問題解決の1つととらえ、この行為を新たな視点から再定義するとともに、計算機によって支援するための方法について論じてきた。特に、人間の身近な問題解決は、設計的な側面と計画的な側面を同時に含んでおり、計算機によってこれらを統合的に扱うことが、より柔軟な問題解決の支援につながることを示した。さらに、提案する枠組みを、実際の製造業における製品開発の事例に適用させた結果を報告し、その有効性を明らかにした。

本章では、これらの結果をもとに本論文の結論を述べる。まず、次節では、本研究の成果を、工学的な観点から総括する。ここでは、本研究において新たに明らかにした事実、開発した手法などをあげ、その新規性と有用性を主張する。また3節では、本研究の成果としての実用面での期待を述べる。提案する手法が、実際の製造業における製品開発に対して与えるであろう効果を予測する。最後に4節で、現時点では実現に至らなかった項目を挙げ、今後の課題とする。

10.2 工学的研究成果

まず、工学的な研究成果としては、以下の3点が挙げられる。

(1) 設計と計画を統合する知識の表現形式を設定

本研究の工学的な成果の第1は、設計と計画を統合する知識の表現形式を設定したことである。これまで提案されてきた多くの知識の表現形式は、個々の問題の性質に固有のものが多く、より一般的でかつ基本的な知識表現は存在しなかった。本研究で開発した知識の表現形式は、問題解決の状況や視点に依存しないプリミティブと、間

題固有の知識であるテンプレートを用いて行なわれる。特にプリミティブは、人間にとって、その知識を表現するための最小の単位であると同時に、計算機にとって、制約処理を基本とした計算あるいは推論を行なう単位でもある。従って、提案した知識の表現形式は、人間と計算機との協調的な問題解決を行なうにあたり、対象とする問題の構造を人間と計算機が互いに理解するための必要不可欠なしくみとなっている。

この知識の表現形式を用いる場合の最大のメリットは、設計という主に空間的な構造を扱う問題解決と、計画という主に時間的な構造を扱う問題解決を統合的に扱えることである。このことにより、従来個別に行っていた両者の問題解決を、より細かな単位でフィードバックし、結果としてより柔軟でかつ効率的な問題解決が可能となる。空間的な構造と時間的な構造を統合的に扱うために、提案した知識の表現形式は関係指向なものとなっている。ここでは、いわゆる“関係”という概念は、“実体”に従属する概念でもなく、“行為”に従属する概念でもない、という基本的な思想のもとに知識表現の単位を設定した。これは、従来のオブジェクト指向や、プロダクションシステムなどに対等する新しいアプローチの1つである。

(2) 時間概念を付加した制約処理アルゴリズムの開発

本研究の工学的な成果の第2は、従来の制約充足問題を中心とした制約指向のパラダイムに加えて、時間概念を取り入れた新たなアプローチを開発し、その有効性を示したことである。ここで時間概念とは、動的な時間つまりなんらかの行為による状態遷移を指す。従来の制約処理では、静的な時間つまりパラメータとして表せる時間を扱ってきた。本研究では、従来の静的な時間に加えて、動的な時間も同一のアルゴリズムの中で扱うことを可能とした。本研究における時間の概念は、オペレータというプリミティブを用いて表現される。この時間概念の性質としては、時間は半順序関係つまり束となること、行為の粒度に依存していること、そして実体との関係によって時間が進行することなどが挙げられる。

本研究では、この時間概念を制約処理における状態遷移計算の中で扱い、独自の計算方法によってこれを処理するアルゴリズムを提案した。従来の制約処理では、制約式あるいは述語によって表現された制約を充足させることにのみ着目するが、本研究ではこれに対し、パラメータの変更は状態遷移計算によって行なうしくみを付加した。これにより、計算機上の問題解決の経過と現実世界における時間の経過をある程度対応づけることを可能にした。提案するアルゴリズムは、従来の制約充足問題と、プランニングなどの探索問題との統合的な問題であるといえ、これを明確な知識表現のもとに形式的に行なっている研究は、現在のところ他に見当たらない。本研究のアプローチ

チは、時間推論の研究の中での新しい試みとしても評価できる。

(3) 悪定義問題に対する効果的な問題解決方法の提案

本研究の工学的な成果の第3は、悪定義問題に対して、精緻化によるトップダウンな問題解決の一方法を示したことである。これまで、悪定義問題に対する工学的な研究が少なかったのは、形式的に記述できない矛盾を含んだ対象や、分節、粒度などが曖昧な対象が、なんらかの形で陽に扱えなかったことによる。本研究では、計算機によって知識表現の分節や粒度を管理し、人間の要求や事実を問題解決の進行に伴って段階的に表現することを可能とした。つまり、関係するさまざまな情報をまず計算機上で表現し、それらの情報を1つの問題として認識する過程そのものを支援の対象となっている点で新規性がある。

いったん問題として認識された構造は、制約処理によって解候補が求められるが、その解候補が要求するレベルに満たない場合は、問題構造の精緻化を行なう。精緻化では問題構造の粒度をさらに細かいレベルに展開するが、ここでは新たな要求や現実の追加も行なわれる。精緻化によるアプローチは、問題分割によるアプローチと比較して、部分問題の間での関係が密である場合には効率が良い。精緻化によつアプローチでは、問題構造における上位の要素と下位の要素との対応関係を考慮することにより、問題解決の粒度を状況に応じて部分的に変更することが可能となる。従って、提案した方法は、大規模複雑な問題に対する新しいアプローチとして期待できる。

以上の3つの工学的な成果は、例題を用いた原理的な説明と、実際の製品開発の過程における問題解決への適用によって実証することができた。従来の知識工学の研究の多くが、例題のみによる動作原理の説明に終始しているのに対し、本研究では、実際の製品開発の事例を扱っており、提案した方法の適用範囲が、現実規模の複雑な問題に対しても有効であることを示した。この点で、本研究は価値がある。特に近年の科学のパラダイムが複雑性をどのように扱うかというテーマに移行しつつあり、本研究は、人間の知能の複雑さの新たな探求における大きな足掛かりとなることが期待できる。

10.3 実用面での期待

本研究で提案した問題解決の方法およびその計算機援用システムは、実際の製造業における製品開発の過程に対して適用可能であり、計算機を用いない従来の方法と比較して、さまざまな点で効果が予想される。以下に、本研究の成果がもたらす製造業

における実用面での期待を列挙する。

(1) 既存知識の有効利用による設計作業の省力化

本研究で提案した知識表現および問題解決の方法の有効性としていえることは、まず、製品の新規開発における既存知識の有効利用による設計作業の省力化である。製品開発の現場では、たとえそれが革新的な製品であったとしても、非常に多くの部分は既存の知識によって構成されている。従って、いかに既存の知識を有効利用するかは、製品開発において非常に重要なテーマである。従来は、この種類の知識は経験やノウハウの中に隠されており、再利用の効率は非常に悪い。本研究で提案する枠組によってそれらの隠された知識を、彼/彼女ら自身の問題解決の過程の中から抽出し、計算機によって管理可能な形で処理することが可能となった。このような知識獲得と再利用に特に着目したアプローチは、製品開発におけるルーチンワークを省力化し、人間をさらに創造的な作業へと向かわせることが可能となる。

(2) ラフな製造プランの生成による問題の早期確認

提案した手法は、製品開発において、ラフな製造プランの生成による問題の早期確認にも有効である。コンカレントエンジニアリングでは、製品開発の各工程をできるだけ並行に進めることが1つの目標であるが、そのようなアプローチには自ずと限界がある。本研究で主張した製品開発へのアプローチは、現実的には従来の製品開発の流れを踏襲しつつも、各工程の大枠についてより上位のレベルで知識のつき合わせをおこない、ある程度の同意を得る。従来の方法では、特に工程や作業方法に関する知識を整理する環境が少なく、それらの多くが属人的なものであったために、製品の形状が明確でない開発の上流では、製造の立場からは意志決定が下せない場合が多かった。本研究では、それらの工程や作業方法に対応する知識をあらかじめ登録し、計算機によってそれらを状況に応じて再構成することで、この問題を解決した。従って、製品開発における非常に大きな問題であった製造計画の段階で判明した不具合による後戻りを、開発の上流において早期発見することで回避することができる。

(3) 製品開発の上流における計算機能力の有効利用

製品開発の過程において、計算機を活用するメリットには、知識の有効利用の他にも、計算機を持つ基本的な計算能力によるメリットがいたるところにある。例えば、製品設計における寸法の決定は、非常に単純な制約関係によって規定されている場合が多いが、製品図面の作成の多くの労力はこのような制約関係に対する対応が占めて

いる。また、製品を製造する場合のプランの生成は、さまざまな製造現場の制約が関係し、これらの情報をいかに効率良くさばくかが重要である。従って、製品開発における問題構造が計算機上で表現された場合には、計算機による基本的な処理である情報検索や制約処理や探索などを利用することで、結果として非常に大きなメリットが得られる場合が少なくない。本研究の実験で行なった形状パラメータの計算やプランの生成、そして設計変更におけるそれらの再計算などは、この典型的な例であるといえる。

(4) 製造環境の変化に対する製品開発のダイナミックな対応

通常の製品開発は、フォーマルには設計から製造への情報伝達の流れが主であり、その逆は歓迎されない。しかし、昨今の製造業をとりまく環境は変化が激しく、このような製造環境の変化にどこまで柔軟に対応できるかがそれぞれの企業の競争優位の源泉となっている。本研究で提案した製品と計画の統合的な問題解決は、このような製造環境の変化に対して設計そのものがダイナミックに適用するための基本的な情報インフラを提供している。新しい製造技術やノウハウの導入があったとしても、それらが最大限に発揮されるためには、それなりの製品設計が行なわれなければならないが、これは設計から製造への一方向の情報伝達では不可能である。従来はこれを人間のインフォーマルなコミュニケーションによって対応してきたが、これらを情報システムとして扱うためには、知識表現を含めた広義のソフトウェアのインフラが不可欠である。そのような意味で、本研究で提案した方法は、これからの製造業における基幹的な情報システムの基盤として、組織の内部に偏在したそれぞれの知識を相互に有効利用しあう仕組みを提供可能である。

(5) 創造的な製品開発のための環境整備への貢献

最後に、創造的な製品開発のための環境を製造業がめざす場合に、本研究で提案した方法がどのように位置付けられるかについて考えたい。一般の企業では製品開発はトップダウンで行なわれる。これは、問題解決の過程においてトップダウンである方が処理効率がよいからである。しかしながら、多くの創造的な製品開発の過程は、このような定型的な製品開発の方法そのものを否定している場合がある。つまり、創造的な製品開発における核となる部分は、ボトムアップに形成される。このようなインフォーマルなモノ作りの過程と、トップダウンに行なわれる全体でのフォーマルな製品開発の過程とをいかに整合させるかが、創造的な製品開発のための重要なテーマである。本研究では、製品開発を問題解決の一貫としてとらえ、そこには個人的な問題

解決、グループによる問題解決、そして組織全体による問題解決という垣根を特に設けていない。また、部分的な問題解決が他の問題解決と無矛盾でなければならないという制限もない。つまり、計算機システムは、問題解決のための方法については深く立ち入ってはならず、あくまでもそのための知識処理環境を提供しているにすぎない。従って、提案する枠組は、創造的な製品開発を阻害する要因はなく、逆に、異なる視点の知識間のインタラクションを誘発する可能性を秘めており、これは創造的な製品開発にとってプラスの要因となるであろう。

10.4 今後の課題

本研究の対象である問題解決という行為は、さまざまな観点からのアプローチが可能であり、本研究のアプローチはその中の有効な一つに過ぎない。しかし、問題解決という行為そのものが非常に奥が深く、本研究において現在の段階で明らかになったことは、そのうちのほんの一部であるといえる。本論文では、以下に示すいくつかの項目について今後の課題とした。

- 要求展開の自動化：

問題解決の過程において、要求をいかに分節し展開するかについては、本研究では問題解決者にすべてを任せるという立場をとっている。しかし、実際の問題解決の過程では、ある程度ルーチン化した要求展開のパターンのようなものが存在し、それらの知識を利用した展開の自動化が望まれるだろう。本研究の知識表現におけるテンプレートを、問題構造の一場面としてではなく、問題解決行為そのものにも適用できるようなしくみを導入することによってこのことが可能となるが、これについては今後の課題とした。

- テンプレート知識の抽出：

提案した枠組を実用化する場合に、そのシステムの利便性を決定する最も大きな要因は、登録されている知識の内容である。特にテンプレートとして経験的な知識がどこまで現実的に整備されてあるかが、最終的にはクリティカルな要素となる。本研究では現在のところ、テンプレートはあらかじめ与えられるものとして進めてきた。しかし、これらのテンプレートは過去の問題解決事例からある程度自動抽出することによって、初めて提案する問題解決の枠組が完成する。過去の事例の中から共通する構造をあらたな知識として抽出する手法については、多くの関連する研究があり、それらをふまえて今後取り組んでいきたい。

- 深い探索への対応：

本研究で提案した制約処理は、その探索の効率という点を現時点では考慮していない。通常 AND-OR 木の探索は、そのままでは非常に多くの計算資源を必要とするために、問題が複雑になるにつれて、ヒューリスティックな探索が必要となってくる。また同時に、個々の部分問題の計算量的な考察も必要であり、それに合わせて柔軟に探索の戦略を使い分けるといったアプローチが必要となるだろう。現時点では、探索の過程は人間と計算機が協調して行なっているが、より複雑な工程を考慮する場合や、自動化の割合を高める場合には、さらに多くの検討が必要である。

- CAD/CAM, CAPP との連動：

提案した問題解決の枠組を製品開発の中できちんと位置付けるためには、従来の CAD/CAM システムや、CAPP(Computer-Aided Process Planning) などの連携をとる必要がある。本研究で開発した PICCSS は、これらの既存の情報システムにとって代わるものではなく、それらを補完するものである。製品開発の上流における意思決定に対して、提案する問題解決の枠組によって支援を行ない、さらに問題が具体的な形になった時点で従来の個々のシステムに受け渡すという形式をとる。本研究では、問題解決結果を製品形状や製造プランのための基本データという形で提供することができたので、その情報を実際の関連システムが対応可能な形にするためのインタフェースの開発が今後必要となるだろう。

- グループ問題解決への拡張：

問題解決には、個人で行なう問題解決、グループで行なう問題解決、そして組織によって行なう問題解決などがあり、それぞれがお互いに関係しあっている。特にグループや組織における問題解決ではそれぞれ視点の異なる知識の相互伝達が重要であり、その上で共通の問題構造を確認しながら1つの解を生成する。本研究では、基本的に問題解決者が1人である状況について提案する方法の有効性を検討したが、おそらくこれをグループや組織による問題解決に適用する場合には、新たないくつかの問題点が出てくると予想される。しかし、特に製造業における製品開発では、このようなグループや組織による問題解決の方が一般的であり、それに対応するように今後拡張することが強く望まれる。

10.5 おわりに

人間が行なう問題解決という行為は、人間の持つ知能のひとつの表れである。本論文では、この問題解決について、モノ作りという具体的な行為を例にとって考えた。モノを作るという行為は、現実世界を作り変えるということを前提としている点で、記号化された世界の中での操作のみに注目してきた従来の記号主義的な問題解決とは一線を画する。現実世界を作り変えるということは、見方を変えれば、現実世界の中での自分自身の位置付けを再構成することである。つまり、モノ作りとは、人間の自己を特徴づける要求と環境との相対的な関係を再構成することによって新しい自己を獲得する過程である、ということができる。本研究では、このような基本思想のもと、現在のモノ作りのあり方を原点にもどって再度検証したかった。この試みは、ある程度達成できたように思う。

現在の製造業は、知識と技術が集積され、より効率的な進化を遂げたモノ作りのための組織である。そこで行なわれている非常に高度なモノ作りは、その断片は人々の目の前で現に行なわれているにも関わらず、それが何であるかを説明することができない。ましてや、それらの断片的な行為の集大成としての製造業のマクロな行為は、いったい何であり、なぜであり、どうなっているのか？ 昔も今もそしておそらく将来も、きちんと説明するとはできないのではないか。ことばのもつ表現能力の限界は、以外と身近かなところで露呈している。本研究の背景にある1つの速大なテーマに、計算機がメディアとして、このことばの持つ限界を越えることができるかというものがあった。現時点では残念ながら、まだ回答を得るに至っていない。

脱工業化ということばが使われるようになって久しい。日本では産業の空洞化、若者の製造業離れ、技術やノウハウの伝承困難など、製造業にとって明るい話題は少ない。モノ（ハードウェア）に価値があった時代からコト（ソフトウェア）へ価値が移行しつつある時代において、当然の成行きとしてモノ作りのあり方そのものを見直すことが必要となってくる。本研究では一貫してモノ作りを人間の問題解決ととらえてきたが、ここで、モノ作りにおける問題は、誰にとっての問題であるかを再検討する必要があるそうである。問題を生産者と消費者という構図の中でとらえるならば、それは消費者の問題でもあり生産者の問題でもある。また、問題を組織と個人という構図の中でとらえるならば、それは組織の問題でもあり個人の問題でもある。

日本や米国を中心とする経済先進数カ国は、確かにモノ余りの時代にある。物質的充足の中で、精神的になにか満たされない社会を人々は生きている。モノを持つことが憧れの対象であった時代から、モノそのものには何の意味も存在しないことが明らかになりつつある時代に移り、今あらためて価値というものが問われている。「創造

的なモノ」を作るよりは、創造的な「モノ作り」を行なうことの方が、結果的により大きな価値を生み出すのではないだろうか。モノ作りという行為そのものに、作る人間にとっての何らかの自己表現の喜びを体感できるようなしくみを取り入れる必要があり、そのための科学と技術をできるだけ早急に確立しなければならない。本論文を締めくくるにあたり、本研究の成果が、次世代のモノ作りのあり方にあたらな視点を与え、もっと人間っぽさを感じさせる人工物が増えることで、現在の“高度消費社会”がよりモノを大切に扱う社会へと移行していくことを願いたい。

参考文献

- [Alan93] Alan, M.D.: Software Requirements, Prentice Hall, NJ. (1993)
- [Allen84] Allen, J.F.: Towards a General Theory of Action and Time, *Artificial Intelligence*, 23, pp.123-154 (1984)
- [Alting89] Alting, L. Zhang, H.: Computer Aided Process Planning: the state-of-the-art survey, *Int. Journal of Production Research*, Vol.27, No.4, pp.553-585 (1989)
- [Ando89] Ando, H., Suzuki, H. and Kimura, F.: A Geometric Reasoning System for Mechanical Product Design, Computer Applications in Product and Engineering (Kimura, F. and Rolstad, A. eds.), Elsevier Science Publishers, pp.131-139 (1989)
- [Boothroyd94] Boothroyd, G.: Product Design for Manufacture and Assembly, *Computer-Aided Design*, Vol.26, No.7, pp.505-520 (1994)
- [Carbonell86] Carbonell, J.G.: Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition, *Machine Learning II* (eds. Michalski, R.S. et al), Morgan Kaufmann, pp.371-1392 (1986)
- [Chen94] Chen, C.L.P., LeClair, S.R.: Integration of Design and Manufacturing: Solving Setup Generation and Feature Sequencing Using an Unsupervised-learning Approach, *Computer-Aided Design*, Vol.26, No.1, pp.59-75 (1994)
- [Checkland90] Checkland, P. and Scholes, J.: Soft Systems Methodology in Action, (妹尾堅一郎監訳, ソフトシステムズ方法論), John Wiley & Sons (1990)

- [Clark91] Clark, K.B., Fujimoto, T.: Product Development Performance (邦訳 製品開発論 ダイヤモンド社), Harvard Business School Press, Boston (1991)
- [Collinot88] Collinot, A., Pepe, C. K. and Pinoteau, G.: "SONIA: A Knowledge-based Scheduling System", *Artificial Intelligence in Engineering*, Vol.3, No.2, pp.86-94, (1988)
- [Cutkosky93] Cutkosky, M.R., et al.: PACT: An Experiment in Integrating Concurrent Engineering Systems, *IEEE Computer*, Vol.26, No.1, pp.28-37 (1993)
- [deKleer86] de Kleer, J.: An Assumption-based TMS, *Artificial Intelligence*, 28, pp.127-162 (1986)
- [Detand90] Detand, J., Kruth, J.-P., Kempnaers, J., Pinte, J. and Kreuzfeldt, J.: The Generation of Non-Linear Process Plans, Preprints of the 22nd CIRP International seminar on manufacturing systems, (1990)
- [Dighe93] Dighe, R., Jakiela, M.J. and Wallace, D.R.: Structural Synthesis Under Manufacturability Constraints: A CAD System for the Design of Injection-Molded Product Housings, *Research in Engineering Design*, Vol.5, pp.185-201 (1993)
- [Elmagraghy93] Elmagraghy, H.A.: Evolution and Future Perspectives of CAPP, *Annals of the CIRP*, Vol.42, No.2, pp.739-751 (1993)
- [ESPRIT93] ESPRIT Consortium AMICE (eds.) CIMOSA: Open System Architecture for CIM, Springer-Verlag (1993)
- [Eversheim93] Eversheim, W., Schneewind, J.: Computer-Aided Process Planning - State of the Art and Future Development, *Robotics & Computer-Integrated Manufacturing*, Vol.10, No.1-2, pp.65-70 (1993)
- [Fikes71] Fikes, R.E. and Nilsson, N.J.: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Artificial Intelligence*, Vol.2, pp.189-208 (1971)

- [Finger89a] Finger, S., Dixon, J.R.: A Review of Research in Mechanical Engineering Design. Part I: Descriptive, Prescriptive, and Computer-Based Models of Design Process, *Research in Engineering Design*, Vol.1, pp.51-67 (1989)
- [Finger89b] Finger, S., Dixon, J.R.: A Review of Research in Mechanical Engineering Design. Part II: Representations, and Design for the Life Cycle, *Research in Engineering Design*, Vol.1, pp.121-137 (1989)
- [Gaddam93] Gaddam, A. and Kaldor, S.: Process-Driven Engineering - A Key Element in Integrating Design and Manufacture, *Annals of the CIRP*, Vol.42, No.1, pp.155-158 (1993)
- [Gui94] Gui, J.-K., Mantyla, M.: Functional Understanding of Assembly Modelling, *Computer-Aided Design*, Vol.26, No.6, pp.435-451 (1994)
- [Harel87] Harel, D.: STATECHARTS: A Visual Formalism for Complex Systems, *Science of Computer Programming*, Vol.8, pp.231-274 (1987)
- [Hori94] Hori, K.: A System for Aiding Creative Concept Formation, *IEEE Trans on Systems, Man, and Cybernetics*, Vol.26, No.6, pp.882-894 (1994)
- [Houten91] Houten, v.: PART: A Computer Aided Process Planning System, (1991)
- [Houten92] Houten, V.: Manufacturing Interfaces *Annals of the CIRP*, Vol.41, No.2, pp.699-710 (1992)
- [Hsu93] Hsu, W., Lee, C.S.G. and Su, S.F.: Feedback Approach to Design for Assembly by Evaluation of Assembly Plan, *Computer-Aided Design*, Vol.25, No.7, pp.395-410 (1993)
- [Iwata90] Iwata, K., Fukuda, Y. and Sugimura, N.: A Proposal of Knowledge Base Structure for Integrated Process Planning System,

- Preprints of the 22nd CIRP International seminar on manufacturing systems, (1990)
- [JIS93a] JIS 用語辞典 機械加工編, 日本企画協会, (1993)
- [JIS93b] JIS 用語辞典 機械要素編, 日本企画協会, (1993)
- [Kim89] Kim,S.H., Lee,K. An Assembly Modelling System for Dynamic and Kinematic Analysis, *Computer-Aided Design*, Vol.21, No.1, pp.2-12 (1989)
- [Kimura90] Kimura,F., Suzuki,H., Inui,M. and Ranta,M.: Product Feature Modelling for Integrating Product Design and Process Planning, Preprints of the 22nd CIRP International seminar on manufacturing systems, (1990)
- [Kimura91] Kimura,F., Suzuki,H. and Tanaka,I.: A Pattern-Directed Design System for Machine Assembly, *Annals of the CIRP*, Vol.41, No.1, pp.127-130 (1991)
- [Kimura93] Kimura,F.: Product and Process Modelling as a Kernel for Virtual Manufacturing Environment, *Annals of the CIRP*, Vol. 42, No. 1, pp.147-150, 1993
- [Kimura94] Kimura,F.: Product and Process Modelling as a Basis for Rapid Product Development, Int Conf on Rapid Product Development, Stuttgart, Germany, pp.92-102 (1994)
- [Ko87] Ko,H., Lee,K.: Automatic Assembling Procedure Generation from Mating Conditions, *Computer-Aided Design*, Vol.19, No.1, pp.3-10 (1987)
- [ISO95] National Institute of Standards and Technology: Product Data Representation and Exchange - Part 214, ISO CD 10303-214 (TC 184/SC4 N335), (1995)
- [Kolodner93] Kolodner,J.: Case-Based Reasoning, Morgan Kaufmann, (1993)

- [Laakko93] Laakko,T., Mantyla,M.: Feature Modelling by Incremental Feature Recognition, *Computer-Aided Design*, Vol.25, No.8, pp.479-492 (1993)
- [Li93] Li, R.-K., Shyu, Y.-T. and Adiga, S.: "A Heuristic Rescheduling Algorithm for Computer-based Production Scheduling Systems", *Int Journal of Production Research*, Vol.31, No.8, pp.1815-1826, (1993)
- [Mantyla88] Mantyla,M.: An Introduction to Solid Modelling, Computer Science Press (1988)
- [Mantyla94] Mantyla,M., Lagus,K. and Lakko,T.: Application of Constraint Propagation in Part Family Modelling, *Annals of the CIRP*, Vol.43, No.1, pp.129-132 (1994)
- [Marefat92] Marefat,M., and Kashyap,R.L.: Automatic Construction of Process Plans from Solid Model Representations, *IEEE Trans on Systems, Man, and Cybernetics*, Vol.22, No.5, pp.1097-1114 (1992)
- [Marefat93] Marefat,M., Banerjee,P., Kashyap,R.L. and Moodie,C.L.: Capturing Intelligence as a Reusable Framework for Manufacturing Decision Process, *Int Journal of Production Research*, Vol.31, No.8, pp.1767-1795 (1993)
- [Martino94] Martino,T.D., et al.: Feature-based Modelling by Integrating Design and Recognition Approaches, *Computer-Aided Design*, Vol.26, No.8, pp.646-653 (1994)
- [McCarthy69] McCarthy,J. and Hayes,P.J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence, *Machine Intelligence*, Vol.4, pp.463-502 (1969)
- [McCarthy80] McCarthy,J.: Circumscription: A Form of Non-monotonic Reasoning, *Artificial Intelligence*, Vol.13, pp.27-39 (1980)
- [McDermott80] McDermott,D. and Doyle,J.: Non-monotonic Logic I, *Artificial Intelligence*, Vol.13, pp.41-72 (1980)

- [McDermott82] McDermott,D.: A Temporal Logic for Reasoning About Processes and Plans, *Cognitive Science*, 6, pp.101-155 (1982)
- [Minsky81] Minsky,M.: A Framework for Representing Knowledge, *Mind Design* (Haugeland,J. ed.), MIT Press, pp.95-128 (1981)
- [Mitchell85] Mitchell, T.M., Mahadevan,S. and Steinberg,L.I.: LEAP. A Learning Apprentice for VLSI Design, *IJCAI-85*, pp.573-580 (1985)
- [Miyashita94] Miyashita, K. and Sycara, K. P.: "A Framework for Case-Based Revision for Schedule Generation and Reactive Schedule Management", *人工知能学会誌*, Vol.9, No.3, pp.426-435, (1994)
- [Nau95] Nau,D.S., Gupta,S.K. and Regli,W.C.: AI Planning Versus Manufacturing-Operation Planning: A Case Study, 14th *IJCAI*, pp.1670-1676 (1995)
- [Newell63] Newell,A. and Simon,H.A.: GPS: A Program That Simulates Human Thought, *Computers and Thought*(Feigenbaum,E.A. and Feldman,J. eds.), McGraw-Hill, New York (1963)
- [Ohsuga93] Ohsuga,S.: How can Knowledge-based Systems Solve Large-Scale Problems?: Model-based Decomposition and Problem Solving, *Knowledge-based Systems*, 6(1), pp.38-62 (1993)
- [Onosato93] Onosato,M., Iwata,K.: Development of a Virtual Manufacturing System by Integrating Product Models and Factory Models, *Annals of the CIRP*, Vol.42, No.2, pp.475-478 (1993)
- [Ow88] Ow, P. S., Smith, S. F. and Thiriez, A.: "Reactive Plan Revision", *AAAI-88*, pp.77-82, (1988)
- [Parsaei93] Parsaei,H.R., Sullivan,W.G.(eds.): *Concurrent Engineering*, Chapman & Hall (1993)
- [Reddy93] Reddy,Y.V.R., Srinivas,R. et al.: Computer Support for Concurrent Engineering, *IEEE Computer*, Vol.26, No.1, pp.12-16 (1993)

- [Reiter80] Reiter,R.: A Logic for Default Reasoning, *Artificial Intelligence*, Vol.13, pp.81-132 (1980)
- [Roy89] Roy,U., Banerjee,P. and Liu,C.R.: Design of an Automated Assembly Environment, *Computer-Aided Design*, Vol.21, No.9, pp.561-569 (1989)
- [Sacerdoti74] Sacerdoti,E.D.: Planning in a Hierarchy of Abstraction Spaces, *Artificial Intelligence*, Vol.5, pp.115-135 (1994)
- [Salomons93] Salomons,O.W., Houten,F.J.A.M. and Kals,H.J.J.: Review of Research in Feature-Based Design, *Journal of Manufacturing Systems*, Vol.12, No.2, pp.113-132 (1993)
- [Sandewall72] Sandewall,E.: An Approach to the Frame Problem and Its Implementation, *Machine Intelligence*, Vol.7, pp.195-204 (1972)
- [Shah91] Shah,J.J.: Assessment of Features Technology, *Computer-Aided Design*, Vol.23, No.5, pp.331-343 (1991)
- [Sche95] スケジューリング・シンポジウム '95 講演論文集, 京都 (1995)
- [Shoham87] Shoham,Y.: Temporal Logics in AI: Semantical and Ontological Considerations, *Artificial Intelligence*, 33, pp.98-104 (1987)
- [Sodhi94] Sodhi,R., Turner,J.U.: Towards Modelling of Assemblies for Product Design, *Computer-Aided Design*, Vol.26, No.2, pp.85-97 (1994)
- [Sriram89] Sriram,D. et al.: Knowledge-Based System Applications in Engineering Design: Research at MIT, *AI Magazine*, pp.79-96 (1989)
- [Sugimoto94] Sugimoto,M., Hori,K. and Ohsuga,S.: Method to Assist The Building and Expression of Subjective Concepts and Its Application to Design Problems, *Knowledge-Based Systems*, Vol.7, No.4, pp.233-238 (1994)
- [Suh90] Suh,N.P.: *The Principles of Design* (設計の原理, 畑村洋太郎監訳, 朝倉書店), Oxford University Press (1990)

- [Tanik89] Tanik, M.M.: Rapid Prototyping in Software Development, *IEEE Computer*, Vol.22, No.5, pp.9-10, (1989)
- [Tijerino93] Tijerino, Y.A., 池田満, 北橋忠宏, 溝口理一郎: タスクオートロジーと知識再利用に基づくエキスパートシステム構築方法論, *人工知能学会誌*, 8(4), pp.476-487 (1993)
- [Tomiyama93] Tomiyama, T., Umeda, Y. and Yoshikawa, H.: A CAD for Functional Design, *Annals of the CIRP*, Vol.42, No.1, pp.143-146 (1993)
- [Tonshoff89] Tonshoff, H.K., Beckendorf, U. and Anders, N.: FLEXPLAN - A Concept for Intelligent Process Planning and Scheduling, *CIRP International Workshop on Computer Aided Process Planning (CAPP)*, pp.21-22, (1989)
- [Tonshoff90] Tonshoff, H.K., Beckendorf, U., Anders, N. and Detand, J.: A Process Description Concept for Process Planning, Scheduling and Job Shop Control, *Preprints of the 22nd CIRP International seminar on manufacturing systems*, (1990)
- [Tsatsoulis88] Tsatsoulis, C. and Kashyap, R.L.: A Case-based System for Process Planning, *Robotics & Computer-Integrated Manufacturing*, Vol.4, No.3-4, pp.557-570 (1988)
- [Watatani92] Watatani, Y. and Fujii, S.: "A Study on Rescheduling Policy in Production System", *Japan/Usa Symposium on Flexible Automation (ASME)*, pp.1147-1150, (1992)
- [Wong93] Wong, A., Sriram, D.: SHARED: An Information Model for Cooperative Product Development, *Research in Engineering Design*, Vol.5, pp.21-39 (1993)
- [Zweben94] Zweben, M., Fox, M.(eds.): *Intelligent Scheduling*, Morgan Kaufmann Publishers, San Francisco, (1994)
- [安部 90] 安部憲広: プランニング, *人工知能学会誌*, 5(6), pp.737-747 (1990)

- [安藤 90] 安藤公一, 吉川弘之: 設計対象の機能記述に基づく設計・製造の統合化精密工学会誌, Vol.56, No.2, pp.133-139 (1990)
- [泉 94] 泉英明: 生産工学, 日刊工業新聞社 (1994)
- [乾 88] 乾正知, 鈴木宏正, 木村文彦, 佐田登志夫: 形状特徴操作に基づく機械加工工程設計の自動化, *精密工学会誌*, Vol.54, No.10, pp.1903-1908 (1988)
- [市田 81] 市田崇, 牧野鉄治: デザインレビュー, 日科技連 (1981)
- [伊藤 90] 伊藤公俊, 曾布川托司: 機械のアセンブリ計画の自動生成 (第1報) - 問題の定式化 -, *精密工学会誌*, Vol.56, No.7, pp.1262-1267 (1990)
- [伊藤 91] 伊藤公俊, 曾布川托司: 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)
- [茨木 94] 茨木俊秀: "スケジューリング問題と計算の複雑さ", *オペレーションズ・リサーチ*, Vol.39, No.10, pp.541-546, (1994)
- [岩田 88] 岩田一明, 福田好朗: 多品種少量生産のための動的工程設計の提案, *精密工学会誌*, Vol.54, No.12, pp.126-131 (1988)
- [上原 91] 上原邦昭: 要求獲得の知的支援, *人工知能学会誌*, 6(2), pp.170-172 (1991)
- [梅田 92] 梅田靖: 機能冗長に基づく自己修復機械設計方法論, 第10回設計シンポジウム講演論文集, pp.155-164 (1992)
- [萱場 95] 萱場孝雄, 加藤康司: *機械工作概論*, 理工学社 (1995)
- [岸 95] 岸義樹, 柴田和人: 仕様充足問題における初期値推定法, *人工知能学会誌*, Vol.10, No.5, pp.92-100 (1995)
- [木村 93] 木村文彦: コンカレントエンジニアリングと仮想生産, *電気学会誌*, Vol.133, No.3, pp.183-188 (1993)
- [桐山 95] 桐山孝司, 富山哲男: 定性物理を用いた設計対象モデル統合化の研究, *人工知能学会誌*, Vol.10, No.4, pp.111-117 (1995)

- [倉賀野 89] 倉賀野哲造, 佐々木伸夫, 菊池敦: 意匠的に特徴ある自由形状の設計およびその工具経路自動生成システムの開発, 精密工学会誌, Vol.55, No.12, pp.2169-2174 (1989)
- [黒田 95] 黒田充: "生産スケジューリングの方法—体系化と状況適合モデル構築の試み—", 経営システム, Vol.5, No. 1, pp.5-14, (1995)
- [小林 92] 小林重信: 計画問題と人工知能: 外観, 人工知能学会誌, 7(6), pp.965-969 (1992)
- [椎塚 92] 椎塚久雄: 実例ベトリネット, コロナ社 (1992)
- [清水 94] 清水周一, 沼尾雅之: 幾何制約に基づく3次元形状の設計, 人工知能学会誌, Vol.9, No.1, pp.129-138 (1994)
- [鈴木 92] 鈴木宏正: 機械設計 CAD と幾何推論, 人工知能学会誌, Vol.7, No.2, pp.219-228 (1992)
- [鈴木 93] 鈴木宏正: STEP における製品モデルの構造, 精密工学会誌, Vol.59, No.12, pp.11-18 (1993)
- [寺野 91] 寺野隆雄: 知識システム開発方法論に関する研究—水力銅構造物評価診断システムの開発に則して—, 東京工業大学博士論文, (1991)
- [富山 94] 富山哲男: Knowledge Incentive Engineering の提案 日本機械学会第4回設計工学・システム部門講演論文集, pp.346-351 (1994)
- [長坂 94] 長坂旬二: CAD データ交換のための標準化活動について, 自動車技術, Vol.48, No.11, pp.54-59 (1994)
- [長澤 89] 長澤勲: 設計支援のパラダイムとインテリジェントCAD, インテリジェントCAD(上), 朝倉書店, pp.113-141 (1989)
- [中島 90] 中島秀之: 状況を対象とした推論, 人工知能学会誌, Vol.5, No.5, pp.588-594 (1990)
- [西岡 94] 西岡靖之: "状況対応型スケジューリング問題へのアプローチ", 計測と制御, Vol.33, No.7, pp.751-754, (1994)
- [畑村 88] 畑村洋太郎: 実際の設計, 日刊工業新聞社, p.4 (1988)

- [畑村 92] 畑村洋太郎: 続・実際の設計, 日刊工業新聞社 (1992)
- [服部 94] 服部雅一, 田中利一, 末田直道: 事例ベース推論による機械設計, 人工知能学会誌, Vol.9, No.1, pp.82-90 (1994)
- [福田 93] 福田好朗: "生産情報の知能化", 精密工学会誌, Vol.59, No.11, pp.1784-1790, (1993)
- [堀 94] 堀浩一: 発想支援システムの効果を議論するための一仮説, 情報処理学会論文誌, Vol.35, No.10, pp.1998-2008 (1994)
- [松下 93] 松下敏広: レンズ付きフィルムのリサイクル設計, 精密工学会 D&C セミナー論文集, pp.61-69 (1993)
- [松田 88] 松田三知子, 乾正知, 鈴木宏正, 木村文彦: 製品設計と工程設計と統合する形状特徴モデリング, 精密工学会誌, Vol.54, No.12, pp.120-125 (1988)
- [松田 89] 松田三知子, 乾正知, 木村文彦: 製品特徴記述にに基づく切削加工情報の自動生成, 精密工学会誌, Vol.56, No.7, pp.1217-1222 (1989)
- [松原 87] 松原仁, 山本和彦: フレーム問題について, 人工知能学会誌, 2(3), pp.266-272 (1987)
- [松原 89] 松原仁, 橋田浩一: 情報の部分性とフレーム問題の解決不能性, 人工知能学会誌, 4(6), pp.695-703 (1989)
- [松本 94] 松本和男: "生産スケジューリングへの熱い期待", 計測と制御, Vol.33, No.7, pp.531-532, (1994)
- [水野 78] 水野滋, 赤尾洋二: 品質機能展開, 日科技連 (1978)
- [溝口 92] 溝口理一郎: 設計問題の知識工学的考察, 人工知能学会誌, Vol.7, No.2, pp.229-236 (1992)
- [溝口 94] 溝口理一郎: 知識の共有と再利用研究の現状と動向, 人工知能学会誌, Vol. 9, No. 1, pp.3-9, 1994
- [元田 93] 元田浩, 溝口理一郎, 西田豊明: 知識の共有と再利用のワークショップ報告, 人工知能学会誌, Vol.8, No.5, pp.666-671 (1993)

- [吉川 85] 吉川弘之: 一般設計学, 機械の研究, 37(1), pp.108-116 (1985)
- [吉川 87] 吉川弘之: 一般設計学, 人工知能学会誌, Vol.2, No.3, pp.273-279 (1987)
- [吉川 92] 吉川弘之: 人工物工学の提唱, *ILLUME*, 4(1), pp.41-56 (1992)

