機能要件と空間制約に基づく概念設計法*

小林 正 和** 鍋 野 正 幸*** 東 正 毅**

Conceptual-CAD Based on the Cooperation between Functional and Shape Designs

Masakazu KOBAYASHI, Masayuki NABENO and Masatake HIGASHI

This paper proposes a novel conceptual design method and support system named "Conceptual-CAD." Traditional conceptual design consists of functional and shape design stages and these stages are executed in order. During the first or the functional design stage, a functional structure is designed by breaking down product's overall function into the collection of elemental functions. During the second or shape design stage, each part that achieves each elemental function is then designed and built up into the product. The proposed method also consists of functional and shape design stages, but two stages are not executed sequentially but executed simultaneously and cooperatively. Using the proposed method, both functional and shape structures are simultaneously designed from top down by considering the relationships between function and shape at each level. Such simultaneous and cooperative design enables simultaneous pursuit of high-functionality, high-performance, space-saving and weight-saving. In order to enable such design, this paper proposes a novel conceptual design method that consists of functional and shape design systems plus a shape realization system that supports the process of designing shapes from functions and a supervisor that checks discrepancy between functional and shape structures.

Key words: Conceptual design, Functional design, Layout design, Function / Means tree and Face representation

1. 緒 言

近年の消費者ニーズの多様化を初めとした社会状況の変化により、企業はより良い製品や今までになかった新しい製品を、限られた開発期間とコストで実現することが求められている。これに対して、CAD、CAE、CAMの普及に伴う、設計活動のデジタルエンジニアリング化が積極的に推し進められている。

しかしながら、既存のCAD、CAEシステムは、詳細設計での使用を目的としたものであり、概念的・概略的かつ創造的なプロセスである概念設計には適さないため、既存のCAD、CAEシステムに代わる概念設計のための設計法が求められている。このような手法としてはこれまでに、機能モデリング手法¹⁾²⁾や、概略的な形状とレイアウトの設計手法³⁾⁻⁵⁾、FOA (First Order Analysis) と呼ばれる解析手法⁶⁾⁷⁾などが提案されているが、これらの手法は概念設計の各段階を個別に支援するための方法であり、概念設計の総合的な支援は難しい。

そこで本研究では、概念設計の主な作業である、機能構造の設計と、概略的な形状とレイアウトの設計を総合的に支援することのできる新しい概念設計法の提案を行う。本設計法を用いることで、機能的な構造と形状的な構造を相互に対応付けながら製品概念を具体化していくことが可能になり、これまでになかった優れた製品が期待できる。

2. 従来の概念設計法

概念設計は、詳細設計に先立って行われる設計の1段階であり、製品の機能構造を構築し、個々の機能を実現するための方策を考え、概略的な形状とレイアウトを作り上げるプロセス⁸⁾である。詳細設計とは異なり概念的・概略的な情報が扱われる

概念設計には、既存のCAD、CAEシステムは適さないため、これらに代わる概念設計のための設計法が研究されている。

概念設計は大きく機能構造の構築と、概略的な形状・レイアウトの具体化の2段階に分けられるが、前者に注目した手法としては、FBSモデリング¹⁾や機能/手段ツリーに基づく方法²⁾などが提案されており、後者に関しては、SPIFF³⁾⁴⁾や3D Layout Module⁵⁾等のように概略的な形状やレイアウトの設計を行うための方法や、FOA (First Order Analysis)⁶⁾⁷⁾と呼ばれる、概略的な形状・構造を対象に解析を行う方法が提案されている。前者については、複雑な機能を持った製品であっても、機能的に矛盾・不足のない機能構造を作り上げることが可能になるという利点があり、後者については、概略的な形状とレイアウトの検討と、空間的な動作によって実現される機能の解析が可能になるという利点がある。

しかしこれらの方法は、機能設計もしくは形状設計のいずれか一方のみを扱うため、前者には、言葉で表現される機能からは実際の製品のイメージが掴みにくい、得られた機能を実現する原理が実際に形状として設計できるかどうか分からない等といった問題があり、後者には、機能的な検討が全く行えない、もしくは空間的な動作によって実現される機能しか扱うことができない等といった問題がある。そのため、これらの問題を解決し、より効果的な概念設計を可能にするために、機能設計と形状設計を統合した新しい設計法が求められている。

また近年、多くの製品で高機能・高性能であることと同時に、小型・軽量であることが求められている。このような製品の設計においては、既存の概念設計のように、初めに機能構造を構築し、その後に形状・レイアウトを具体化するのではなく、機能構造と形状・レイアウトを相互に考慮しながら両者を並行して具体化していくことができれば、両者をより高い水準で実現することが可能になる。

そこで本報では、概念設計の主要な作業である機能設計と形状設計を支援の対象とし、図1に示すように、機能と形状・レ

* 原稿受付 平成21年2月9日

** 正 会 員 豊田工業大学 (愛知県名古屋市中白区久方 2-12-1)

*** 学生会員 豊田工業大学 (現、トヨタ自動車(株); 愛知県豊田市トヨタ町1番地)

Fig.1 Proposed conceptual design procedures

Table1 Definition of keywords

Overall ↓ Detail	Overall shape	Overall function	Component
	Structure and Mechanism	Function	
	Element	Elemental function

Fig.2 Configuration of the proposed conceptual design system

アウトを相互に対応付けながら、最上位の全体機能と全体形状から、個々の要素機能と要素（部品）にまでトップダウンに詳細化していくことのできる総合的な概念設計法と、コンピュータ上に実装した支援システムである概念設計 CAD の提案を行う。図 1 に示すように、(1)形状設計を通常とは逆にトップダウンに実施する、(2)機能と形状・レイアウトを相互に対応付けながら機能設計と形状・レイアウト設計を平行して実施する、の 2 点が提案手法を用いた概念設計の特徴である。

手法の説明に先立って、機能と形状の階層構造に関わる用語を表 1 のように定義する。全体機能とはその製品の果たすべき役割であり、それ以上抽象化することのできない一つの最上位の機能を表す。全体形状は形状の最上位であり、製品の外形形状を表す。一方で、要素機能および要素はそれ以上分解することのできない、最下位の機能と形状を示す。形状の階層構造における要素は、製品を構成する部品と 1 対 1 の関係を持つ。また、階層間で機能や形状の上下関係を表す場合には、1 階層上の形状や機能については親と呼び、1 階層下の形状や機能については子と呼ぶ。また、機能や形状、階層の区別なく階層構造のある部位を指し示す場合には、コンポーネントと呼ぶ。

3. 概念設計法

3.1 概念設計法の概要

図 2 に本設計法の構成を示し、以下に各部の概略を述べる。

(1) 機能設計システムと形状設計システム

機能構造の設計と、概略的な形状・レイアウトの設計は、概念設計の主たる作業であり、本設計法の支援対象である。そこで、これらの作業を支援するために、機能／手段ツリー³⁾に基づく機能設計システムと、面ベースフィーチャモデル¹⁰⁾¹¹⁾に基づく形状設計システムを構築する。

(2) 形状導出システム

機能／手段ツリーに基づいて機能構造を表現する場合、機能

Fig.3 Global and local flows of the proposed conceptual design system

を実現するための方策として手段があり、形状はその手段を実体化したものとなる。そのため、機能から手段を求め、手段から形状を求めるのが最も基本的な設計の流れとなる。形状導出システムはこの一連の流れの中で、手段から形状を導出するプロセスを支援するものである。

(3) 機能／手段データベースと手段／形状データベース

一般的な製品設計においては、機能と形状の両面において多くのコンポーネントは再利用可能である。そこで、過去の設計で得られた機能構造を蓄積した機能／手段データベース（F/M データベース）と、手段と形状の対応関係を蓄積した手段／形状データベース（M/S データベース）を構築し、過去の設計情報を再利用する仕組みを提供する。

(4) 形状位相変化の監視システム

多くの機械部品では、その機能は形状の位相や構造によって実現される。ここで位相とは形状要素が持っている形状の特徴の構成であり、構造とは形状要素がどのように組み合わせられているかのことである。そのため、各コンポーネントの寸法が変更され、位相や構造が変化した場合、コンポーネントの機能が変化してしまう可能性がある。そこで本設計法では、このような位相や構造の変化を監視するシステムを導入する。

本設計法を用いた概念設計の流れを図 3 に示す。大域的には機能と形状を全体機能と全体形状という最上位のコンポーネントから、要素機能と要素にまで段階的に詳細化していくことで、機能構造と形状構造を構築するという流れがあり、これは図 3 においては、スパイラルが上から下へと回転しながら進んでいく様に相当する。一方で、局所的には機能から手段を求め、手段から形状を具体化する作業を繰り返すという流れがあり、これは図 3 のスパイラルの 1 回転に相当する。

以降、本設計法を構成する各部についてその詳細を述べる。

3.2 機能／手段ツリーに基づく機能設計

3.2.1 機能／手段ツリー

機能／手段ツリー³⁾は機能分解手法の 1 手法であり、機能を実現するための手段を陽的に考慮し、機能と手段を交互に挙げながら機能構造を作成していく点に特徴がある。ここで手段とは機能を実現するための実体を持った方策であり、例えば“トルクを発生させる”という機能に対する手段としては“電気モーター”や“エンジン”などが考えられる。図 4 は機能／手段ツリーの一例であり、四角形が機能を、楕円形が手段を表す。

機能／手段ツリー作成の手順としては、(1)親機能・手段を実現するための機能を設定する、(2)その機能を実現するための手段を設定する、という 2 つの作業を 1 サイクルとして、これを繰り返しながら段階的に機能構造を詳細化していく。機能／手段ツリーでは機能と手段を交互にツリー構造に追加していくことから、作成される機能構造は図 4 に示すように、機能と手段

Fig.4 Function / Means tree

が交互に現れる。機能／手段ツリーでは、1つの機能に対して、複数の実現手段が存在することも考えられ、この場合、選択した手段に応じて以下の機能構造が変化していく。図4はその1例であり、ここでは“トルクを発生させる”という機能に対して“電気モーター”と“エンジン”という2つの手段が設定されている。ここで手段“電気モーター”を選択した場合、下位機能は図4(a)に示すように“電圧を起こす”と“電圧をトルクに変換する”の2つとなり、手段“バッテリー”、“モーター”がそれぞれの下位機能を実現する。一方で、手段“エンジン”を選択した場合は、下位機能は図4(b)に示すように“燃料を供給する”と“燃料をトルクに変換する”の2つとなり、それぞれ手段“燃料供給装置”、“エンジン”によって実現される。機能／手段ツリーを採用する利点としては、(1)機能を実現するための手段を陽的に考えることで、より現実的な機能構造が得られやすい、(2)手段という具体的な方策を考慮することで、機能と形状の関連を持たせやすい、の2点が挙げられる。

3.2.2 機能／手段ツリーに基づく機能設計の流れ

機能／手段ツリーに基づく機能設計では、機能の設定と手段の設定という2つのステップを1サイクルとして、このサイクルを繰り返し行うことで、全体機能から機能要素まで段階的に機能構造を詳細化していく。以下に各ステップの詳細と、具体的な機能／手段ツリーの記述方法について述べる。

ステップ1 機能の設定

設計者は親となる機能・手段を実現するための機能を設定する。本手法では、以下に述べる動詞と目的語の組み合わせにより機能を記述する。

・動詞

設計情報の検索と再利性を向上させるために、原則として、付録1に示した Function basis⁹⁾で整理、分類された動詞を利用する。コンポーネントを表現する適切な動詞がない場合は設計者が定義することができるが、できる限り一般化された最小限の動詞のみを追加するように留意する。

・目的語

目的語は動詞の働きを受ける対象であり、製品内部に流れるフローか他のコンポーネント、もしくは製品外部のオブジェクトを選択する。目的語は必ずしも1つだけとは限らず、「AをBに変換する」のように複数設定することもできる。

ここでフローとは、製品の機能を達成するために、製品内に流れる3種類の対象（物質、信号、エネルギー）であり、個々の機能の入出力に相当する。ただし、全ての機能が入出力を持つわけではなく、入出力の一方のみ、もしくは入出力とも持たない場合もある。本手法では、フローの種類として動詞と同様に Function basis⁹⁾のフローを用いる。詳細は文献9を参照されたい。目的語として選択可能なフローは、以下の2種類に大別することができる。

- (1) 親機能から継承された入出力フロー
- (2) 親機能の内部フロー

Fig.5 Inheritance of the flows and definition of the internal flow

Fig.6 Connection features

本手法では、親機能が入出力を行う働きをするとき、親機能の目的語に相当するフローをシステムが子機能の目的語に対し選択可能なフローとして継承させ、また、必要に応じて親機能内部のフローを設計者が追加する。図5は親機能からのフローの継承と、設計者が追加する親機能の内部フローについて説明した図である。同図に示すように、親機能が複数の子機能に分割されるとき、親機能の入出力フローはいずれかの子機能が継承し、その入出力を繋ぐように内部フローが新たに設定される。また、それ以外にも親機能を実現するために追加のフローが必要となる場合は、それを内部フローとして定義し、目的語として使用することができる。

最後に、製品外部のオブジェクトとは、製品の外部に位置し、製品が動作する過程で製品に関わってくる、3種類の対象（物質、信号、エネルギー）である。自動車の最上位機能「人間を運搬する」の目的語「人間」はその一例である。

ステップ2 手段の設定

設計者はステップ1で設定した機能を実現するための手段を設定する。本設計法では過去の設計で用いられた機能と手段の対応関係を F/M データベースに蓄積しており、ステップ1で設定された機能をデータベースに蓄積された機能と照らし合わせ、対応する手段が存在する場合はそれを候補として設計者に提示する。設計者は提示された手段の中から適したものを選択するか、もしくは新たに手動で入力する。手段の記述形式としては、手段の設計要素名と手段のタイプを組み合わせる。

・設計要素名

例えば電気モーター、ガソリンエンジン、バッテリーなどのように、設計要素名は手段の名称を表す。

・手段のタイプ

手段のタイプは以下に記すコンポーネント、関係フィーチャ、接続フィーチャの3種類から選択する。

- (1)コンポーネント：下位のコンポーネントの組み合わせにより機能を実現する機構や構造。最下層ではそれ以上分解することのできない要素となり、個々の部品と1対1に対応する。
- (2)関係フィーチャ：コンポーネント間の自由度を拘束することで実現される機能に対応した、実体としての形状を持たない自由度拘束のみの特殊な手段のタイプ。多くの機構ではコンポーネント間の特定の自由度を拘束することで機能を実現

Fig.7 Relationships between functions and means in F/M database

する。そこで、このような自由度の拘束は実体としての形状は存在しないが手段の一種として扱う。

(3)接続フィーチャ：実体としての複数の形状と、その形状間に付与された自由度拘束により実現される機能に対応した特殊な手段のタイプ。接続フィーチャは、複数のコンポーネントと関係フィーチャの組み合わせにより実現されるフィーチャである。図6は接続フィーチャの例であり、同図(a)は2つのコンポーネントを特定の位置関係で固定する機能、同図(b)と(c)はそれぞれ回転運動と直線運動を支える機能を持つ。このような機能は機械製品ではよく用いられるため、独立した手段のタイプとして扱う。

3.2.3 F/M データベースとその活用

F/M データベースは、機能／手段ツリーの構築を支援するために、過去の設計で作成された機能と手段の対応関係を蓄積したデータベースである。F/M データベース内で機能が持つ情報としては、機能そのものを定義する(1)動詞と目的語の組み合わせと、(2)対応する手段の設計要素名である。(2)の対応する手段の設計要素名については複数持つことができる。手段が持つ情報としては、(1)設計要素名と、(2)手段のタイプ、(3)初期化用の幾何拘束（コンポーネント、接続フィーチャの場合）もしくは拘束する運動自由度（関係フィーチャの場合）に加えて、(4)1階層下の子機能との対応関係、具体的には対応する子機能の動詞＋目的語である。(3)は形状導出の過程で用いる情報であり、この項目には過去の設計において入力した幾何拘束や拘束情報を記録する。なお、この情報をどのように用いるかについては3.4.2項で述べる。また、(4)の子機能については複数持つことができる。図7にF/M データベースにおける、機能と手段が持つ情報と対応関係を示す。設計者が機能／手段ツリーに機能を追加すると、システムはデータベース内の機能を検索し、同一の機能が存在する場合は、その機能に対応付けられた手段を呼び出し、次の手段の候補として設計者に提示する。

F/M データベースでは、手段の情報として1階層下の子機能の情報を持つため、現在注目している階層から下位に向けて機能と手段の対応関係を順次検索していくことで、過去に設計された機能構造を再現し、再利用することができる。

3.3 面ベースフィーチャモデルを用いた形状設計

形状設計のための形状モデリング手法として、本設計法では面ベース表現¹⁰⁾¹¹⁾に基づく面ベースフィーチャモデルを用いる。面ベース表現は、一般的な稜線ベースの境界表現法とは異なり、面を中心としてその回りの隣接関係について、面名を用いて記号的に表現する表現法である。具体的には、隣接する面を反時計回りに並べた面名のリストとして表すことにより、面をノー

(a) Face name

Face name	Adjacent-Faces List	Normal vector, Distance
a	f[2], s3, s2, s1, f[1], d, e, b	{{1, 0, 0}, 10}
b	f[2], a, e, c	{{0, 1, 0}, 10}
c	f[1], s1, s2, s3, f[2], b, e, d	{{-1, 0, 0}, 10}
d	f[1], c, e, a	{{0, -1, 0}, 10}
e	a, b, c, d	{{0, 0, -1}, 10}
f[1]	a, s1, c, d	{{0, 0, 1}, 10}
f[2]	c, s3, a, b	{{0, 0, 1}, 10}
s1	a, s2, c, f[1]	{{0, 1, 0}, 3}
s2	s3, c, s1, a	{{0, 0, 1}, -6}
s3	c, s2, a, f[2]	{{0, -1, 0}, -7}

(b) Face Connection Table (c) Face Data Table
Fig.8 Face representation

ドとするグラフ構造を表現する。さらに、頂点名、稜線名や干涉計算における交点名を、面名のリストとして表すことにより、その名前から接続関係や位相の拘束関係を求め、更に幾何情報を加えることで形状を表現する。図8に面ベース表現の例を示す。面接続表が位相構造を示し、面データ表が幾何情報を示す。面ベース表現を用いることで、ボスやリブといったフィーチャの位相構造を記号操作によって求めることができる。このように位相拘束を記号操作によって容易に操作できるのが面ベース表現の特徴であり、位相の変化を含む大きな形状変化が生じやすい概念設計での形状表現法として適している。

上記の説明では多面体を表現しているため、面データ表には平面の方程式として、各平面の単位法線ベクトルと原点からの距離を格納しているが、これらの代わりに曲面の方程式をFDTに格納することで、曲面を含んだ形状についても取り扱うことができる。しかしながら、本設計法における形状設計は、概略的な形状とレイアウトの設計が目的であることから、計算の効率化のために、立方体を基本形状とし、これにボスやリブといったフィーチャを付与することで全てのコンポーネントの形状を表現する。機械製品では円柱や歯車のような立方体以外の形状が多く含まれるが、これらについても、面ベースフィーチャモデル内部ではこれらを包絡する多面体として計算を行い、画面上でのみ円柱や歯車としての形状を表示することとする。

3.4 形状導出部：手段を実現する形状の導出

本設計法では、機能／手段ツリーで設定された手段から形状を導出するにあたって、過去の設計で得られた手段と形状の対応関係を蓄積した手段／形状データベース(M/S データベース)を利用する。

3.4.1 M/S データベース

M/S データベースは、手段から形状を導出するプロセスを支援するために、過去の設計で得られた手段と形状の対応関係を蓄積したデータベースである。M/S データベース内には、形状の情報として、(1)対応する手段の設計要素名と、(2)面ベース表現により記述された形状の位相情報を持つ。

3.4.2 形状導出の流れ

機能設計に際して、設計者がシステムにより提示された手段のいずれかを選択するか、もしくは新たに手動で手段を入力したとき、手段のタイプがコンポーネント、もしくは、接続フィ

ーチャであった場合は、システムはその手段に対応した形状を以下の3ステップに従って作成する。

ステップ1: M/S データベースの検索

システムは手段の設計要素名をキーとして、M/S データベースから同一の名称を持つ形状位相を検索する。手段が設計者によって新たに入力された新規の手段である場合は、このステップを省略する。

ステップ2(a): 初期形状の実体化

ステップ1の検索の結果、手段に対応した形状位相が存在する場合は、データベースから対応する形状位相を読み出して形状モデラに送り、初期形状を実体化する。形状の実体化には幾何拘束が必要となるが、ここではF/M データベース内に手段の情報として記録されている初期化用の幾何拘束を使用する。この値は具体的には、前回の設計で用いられた形状の寸法である。なお、この時点では形状は空間内の原点に配置される。

ステップ2(b): 形状位相の設定

ステップ1の検索の結果、手段に対応した形状が存在しなかった場合、もしくは新しい形状位相を与えたい場合は、設計者が手で形状位相と幾何拘束を入力し、形状を実体化させる。

ステップ3: 配置と幾何拘束の調整

設計者は、空間的な制約を配慮しながら、実体化された形状の配置と幾何拘束を調整する。配置や幾何拘束の設定方法としては、値を直接入力する方法と、他のコンポーネントとの相対的な関係で指定する方法がある。ここで相対的な関係とは、2つのコンポーネントの面と面が接する場合や、他のコンポーネントからある距離に位置する場合である。相対関係を用いる場合は、システムが全体の整合性を取るように値を設定する。

一方、手段のタイプが接続フィーチャであった場合、システムはF/M データベース内の該当手段の(3)拘束する運動自由度の項目を確認する。拘束する運動自由度が記録されている場合はその情報を読み出し、記録されていない場合は新たに拘束する運動自由度を設計者が選択する。そして、その拘束を付与するコンポーネントを設計者が選択することで、形状モデラ内に実体化されているコンポーネント間の運動自由度が拘束される。

3.5 形状位相変化の監視

機械製品では多くの機能が形状位相、特にボスやリブ、スロットのようなフィーチャによって実現されているため、幾何拘束の調整などにより形状位相が変化すると、その機能は失われてしまう。例えば図6(c)では、RidgeとSlotの噛み合いによりスライダとしての機能が実現されているが、仮に幾何拘束の変更によりSlotがOpen-Stepに変化したとすると、スライダとしての機能を満たさなくなる。

そこで本設計法には、形状位相の変化を監視し、設計者に警告する機能を持たせる。具体的には、設計者が形状の幾何拘束を変更する場合や、他のコンポーネントの形状変更に伴って全体の整合性を取るために形状が変更されるような場合には、それらの変更により形状位相の変更が生じるか否かをシステムが確認し、変更が生じる場合は、警告を発する。

4. 概念設計CAD

3章で述べた概念設計法を実際の設計活動で用いるために、コンピュータ上にプロトタイプシステムを構築する。本報ではこのシステムを概念設計のためのCAD、すなわち概念設計CADと呼ぶ。使用言語は基本的にSun Microsystems Java2およびJava3Dであり、面ベースフィーチャモデラのみWolfram

Research Mathematica 5.1を使用する。MathematicaとJavaの接続にはJ-Linkを用いる。また、データベースの記述にはXMLを用いる。図9に概念設計CADの構成を示す。

概念設計CADのGUIは実行例の図10を参照されたい。GUIは2つのサブウィンドウから構成されており、左側のウィンドウに機能/手段ツリーが表示され、右側のウィンドウに形状モデルが表示される。

5. 実行例

提案手法と概念設計CADを用いた設計の流れを示すために、1人乗りの小型自動車を設計対象とした簡単な実行例を示す。

5.1 設計の流れ

5.1.1 全体機能の入力

ここでは小型自動車の目的を「人間を移動させる (Transport Human)」と考え、これを全体機能として入力する。手順としては、まず最初にオブジェクト設定ウィンドウで“Human”を定義する。このウィンドウは各種物質やエネルギーフローといった機能の目的語として用いられるオブジェクトを定義するためのものである。次に機能設定ウィンドウで“Transport”という動詞をプルダウンメニューから選択し、先に定義した“Human”を目的語として選択することで“Transport Human”という機能を定義する。その結果、図10左側のウィンドウに“Transport Human”が表示される。

5.1.2 全体機能に対する手段の入力と全体形状の実体化

設計者により機能が入力されると、システムは入力された機能をF/M データベースに蓄積された機能と照らし合わせ、対応

Fig.9 Construction of conceptual CAD

Fig.10 Result of the first cycle

する手段が存在する場合はそれが提示される。ここでは“Transport Human”に対応する手段がデータベースに存在しなかったため、手段入力ウィンドウを用いて、手動で手段を入力する。手段入力ウィンドウでは、手段が実現する機能と、手段のタイプ、フィーチャタイプを設定する。ここでフィーチャタイプは手段を実体化するにあたっての基礎となる形状位相を示し、現時点では立方体、円柱、歯車の3種類が選択可能である。また、これらの形状に更にフィーチャを付加することで複雑な形状を作成することもできる。本例では、手段のタイプとしてコンポーネント、フィーチャタイプとして立方体を選択する。次に形状作成ウィンドウを用いて形状の名称と寸法、親形状を入力し、関連付けウィンドウで手段と形状の関連を定義することで、右側のウィンドウに形状モデルが表示され、左側のウィンドウに表示された機能／手段ツリーに手段が追加される。本例では、小型自動車の名称として“smallCar0”，幾何拘束として小型自動車の寸法値を入力し、親は存在しないため最上位を示す ROOT を選択する。そして，“Transport Human”と“smallCar0”を関連付けることで、図 10 に示すように手段と形状が表示される。以上が、概念設計 CAD を用いた概念設計の最初の 1 サイクルである。

5.1.3 子機能の入力

次に機能“Transport Human”と手段“smallCar0”に対して子機能を入力する。ここでは“Generate Drive force”，“Increment Drive force”，“Change Drive force into Acceleration”，“Stabilize Human”の4つの子機能を用いて，“Transport Human”を実現することを考える。子機能の入力にあたっては、まず初めにエネルギー入力ウィンドウを用いて製品の内部を流れるエネルギーである“Drive force”と“Acceleration”を入力する。ここで設定するエネルギーは、機能を定義する目的語の一つである、エネルギーフローとして用いられる。次に、全体機能の入力と同様に機能入力ウィンドウを用いて子機能を入力するが、親となる機能と手段を設定することで、子機能は機能／手段ツリーの適切な場所に配置される。子機能の目的語としては、上位機能から継承したフローに加えて、新たに入力したエネルギーフロー、およびこれまでに入力された手段を選択することができる。本例では、親から継承した“Human”，新たに入力したエネルギーフローである“Drive force”と“Acceleration”，および前項で入力した手段である“smallCar0”を選択することができる。

この子機能に対して手段を入力し、またその子機能を入力するという手順で機能と手段の階層構造を段階的に詳細化し、また、それに応じて個々の手段の形状を導出していくのが、概念設計 CAD を用いた概念設計の基本的な流れである。

5.1.4 F/M データベースと M/S データベースの利用

機能／手段ツリーの作成に際して、設計者の入力した機能に対応する手段が F/M データベースに存在する場合は、それを再利用することができる。本例では、前項で入力した4つの機能に対応する手段が存在したため、図 11 に示すように、それぞれに対応する機能の下側に候補として表示されている。この例のように対応する手段が F/M データベースに複数存在する場合には、それらの手段は全て表示され、設計者はどれを使用するかを選択する。候補の段階では、手段は白地に黒色の文字で表示され、選択された手段は色が反転する。さらに、選択した手段の下位構造が F/M データベースに存在する場合はこの下位構造を再利用することもできる。本例では、手段“Engine”と“Gears”に過去に設計された下位構造が存在したため、それを再利用す

Fig.11 Reuse of means defined in previous projects

Fig.12 Reuse of previously designed shape structures

Fig.13 Result of case study

る。図 14 に手段“Engine”と“Gears”の下位構造を読み込んだ結果を示す。さらに、再利用する手段に対応した形状が M/S データベースに存在する場合は、これらを再利用することもできる。図 12 には手段“Engine”とその下位構造に対応する形状を M/S データベースから読み出した直後の状態を示す。図 12 の段階では Engine の配置が適切でないため、この後に配置の調整や寸法の調整を行う。このように、F/M データベースと M/S データベースを利用することで、概念設計を効率的に進めることができる。

5.2 考 察

本例で作成した小型自動車の形状を図 13 に示し、図 14 に機能／手段ツリーを示す。この図の段階ではまだ小型自動車は完成していないが、以上が概念設計 CAD を用いた概念設計の基本的な流れである。

最後に概念設計 CAD の利点をまとめる。

- ・ 機能的な構造と形状的な構造を相互に考慮しながらトップダウンに製品概念を作り上げていくことができる。
- ・ 機能／手段データベースと、手段／形状データベースという2つのデータベースを用いることで、機能設計と形状設計の両段階で過去の設計情報の再利用が可能となる。
- ・ システムが形状位相の変化を監視しており、設計者の変更

Fig.14 Reuse of previously designed F/M trees

によって位相が変化する場合には警告が出るため、モデルが複雑になっても、形状変更時の整合性が保ちやすくなる。

6. 結 言

本報では、これまでにない優れた機械製品を実現するために、従来の CAD・CAE システムの支援が及ばない概念設計段階に注目し、概念設計の主な作業である、機能構造と形状・レイアウトの設計を総合的に支援することのできる新しい概念設計法の検討を行った。以下に本報の成果をまとめる。

- 機能と形状・レイアウトの両階層構造について、相互に対応を取りながら、トップダウンに詳細化していく仕組みを構築した。これにより、機能と形状・レイアウトの両面を高い水準でバランスさせることが可能になる。
- F/M データベースと M/S データベースという2つのデータベースにより、過去の設計情報を蓄積・再利用する仕組みを構築した。通常の製品設計においては、多くの機能・形状が再利用できるため、データベースを用いることで効率的・効果的な設計が可能になる。
- 提案手法の実設計での使用を可能にするために、コンピュータ設計支援システムのプロトタイプを構築し、概念設計 CAD と名付けた。

今後の展望としては以下の2点が挙げられる。

- 現状では、F/M データベースからの検索、もしくは設計者によって行われている機能や手段の設定を、過去の設計情報の分析や、動作解析によって求められるようにする。
- データベースに蓄積された設計情報を再利用するにあたって、最適な機能、手段の選択を可能にするために、最適化手法を導入する。

参 考 文 献

- Y. Umeda, T. Tomiyama and H. Yoshikawa: Proposal of Function-Behavior-state Modeling to Support Function Design, Journal of the Japan Society of Precision Engineering, **63**, 6(1997)795. (in Japanese).
- W. Sören: Function- and constraint-based conceptual design support using easily exchangeable, reusable principle solution elements, Artificial intelligence for engineering design, analysis and manufacturing, **19**, 3 (2005), 201.
- W. F. Bronsvort, R. Bidarra, M. Dohmen, W. van Holland and K. J. de Kraker: Multiple-view feature modelling and conversion. In: W. Strasser, R. Klein and R. Rau, Editors, Geometric modeling: theory and practice—the state of the art, Springer, Berlin (1997), 159.
- W. F. Bronsvort and A. Noort: Multiple-view feature modelling for integral product development, Computer-Aided Design, **36**, 10 (2004) 929.
- A. Csabai, I. Stroud, and P. C. Xirouchakis: Container spaces and functional features for top-down 3d layout design, Computer-Aided

Design, **34**, 13 (2002) 1011.

- H. Nishigaki, S. Nishiwaki, T. Amago, and N. Kikuchi: First Order Analysis for Automotive Body Structure Design, Proceedings of DETC2000, ASME, (2000) DETC2000/DAC-14533.
- H. Sugiura, E. Nishigaki, S. Nishiwaki and Y. Kojima: Application of First Order Analysis to Suspension Design, Proceedings of Design & Systems Conference, 2001, 10(2001)124 (in Japanese).
- G. Pahl, W. Beitz, J. Feldhusen, and K. H. Grote: "Engineering Design: A Systematic Approach", Springer-Verlag, (2007), 125.
- J. Hirtz, R. B. Stone, D. A. McAdams, S. Szykman, and K. L. Wood: A Functional basis for Engineering Design: Reconciling and Evolving Previous Efforts, Research in Engineering Design, **13** (2002) 65.
- M. Hosaka and T. Sata: Integrated CAD/CAM system, Ohmsha, (1994) (in Japanese).
- M. Higashi, H. Senga, H. Nakano and M. Hosaka: Parametric Design Based on Topological and Geometrical Constraints, Journal of the Japan Society of Precision Engineering, **67**, 2(2001)229 (in Japanese).

Appendix1 Functional basis reconciled function set⁹⁾

Class (Primary)	Secondary	Tertiary
Branch	Separate	Divide
		Extract
		Remove
		Distribute
Channel	Import	Diffuse
		Export
		Transfer
		Transport
Connect	Couple	Transmit
		Guide
		Translate
		Rotate
Control	Actuate	Allow DOF
		Join
		Link
		Add
Magnitude	Regulate	Mix
		Actuate
		Regulate
		Change
Convert	Convert	Increase
		Decrease
		Increment
		Decrement
Provision	Store	Shape
		Condition
		Stop
		Prevent
Signal	Sense	Inhibit
		Convert
		Store
		Contain
Support	Stabilize	Collect
		Secure
		Constrain
		Position
Support	Align	Align
		Align
		Align
		Align