



Title	分散協調型設計支援システムのためのプロセスモデルと操作モデル（第1報, プロセス構造についてのモデル）
Author(s)	藤田, 喜久雄; 菊池, 慎市
Citation	日本機械学会論文集 C編. 2002, 68(666), p. 657-665
Version Type	VoR
URL	https://hdl.handle.net/11094/2977
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

分散協調型設計支援システムのためのプロセスモデルと操作モデル (第 1 報：プロセス構造についてのモデル)*

藤田 喜久雄^{*1}, 菊池 慎市^{*2}

Computational Models for Concurrent Design Process Support (1st Report: Process Structure Model)*

Kikuo FUJITA^{*3} and Shin'ichi KIKUCHI

^{*3} *Department of Computer-Controlled Mechanical Systems, Osaka University,
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan*

This research proposes computational models for concurrent design process of large complicated systems as the formal fundamentals for the development of agent-based distributed design support systems. In this first report, we assume that an artifact is modeled with a set of lumped mass systems and that the number of variables is fairly huge to facilitate the discussion. Under these assumptions, the design phases, the types of design knowledge, and their characteristics such as granularity levels, operational cost, fidelity, dominatedness, etc. are formally discussed. Furthermore, the design model is primarily decomposed into the global model and domain models, and the horizontal and vertical decomposition of a design process are formulated. Following such decomposition structure, the conception of goal cascading is proposed as a model of coordination structure among the divided design tasks. The goal means the play on an intermediate variable shared among distributed tasks. The discussion deducts three types of coordination: inter, global and inner ones. These understandings are expanded to operational models and a computer-aided design support system in the succeeding papers respectively.

Key Words : Design Engineering, Design Process, Design Task Distribution, Goal-Driven Coordination, Computational Design Model, Concurrent Engineering

1 緒 言

大規模なシステムの設計は組織によってあらかじめ想定された期間内で遂行されている。このようなプロセスにおいて、様々なコンピュータ援用設計支援技術が重要なツールとなりつつあるものの、各種の内容は設計における個別的な処理を効率化するものであって、設計プロセスの全体からみれば、いささか補助的なものに留まっていると言わざるを得ない。これに対して、設計シンセシスにおける各種情報間の関連性やタスク間の構造に対応した支援システムの構築を目標として、いわゆるコンピュータ援用コンカレントエンジニアリング (Computer-Supported Concurrent Engineering) と称される研究が様々な視点から試みられつつある⁽¹⁾⁽²⁾。従来からの設計支援技術は設計における特定の局面を切り出した上で個別的に展開されるものであることに対して、一連の試みはプロセス全体

に向けての統合的な支援を構成しようとするものであり、これに関して、藤田ら⁽³⁾は、大規模な設計問題を部分問題に分割するための「分散化」、それらの間で並行処理を形式的に実現するための「並行化」、並行処理のもとで合理的な設計を展開するための「協調化」という 3 つの段階の必要性を指摘している。

本研究は、上述の動向に対応した分散協調型の設計支援システムの実現に向けて、大規模なシステムの設計プロセスにおける規模性に起因する複雑さに対応した設計支援のための計算による (computational) モデル⁽⁴⁾を確立することを目的としている。本報では、まず、それに向けて、設計問題の内容が大規模な集中定数系で表現できるものと仮定した上で、あらかじめ定められた時間の範囲内では、そのような問題の完璧な解を求めることはできず、制限された状況下での何らかの優れた満足解を導出することが求められているとの想定を導入し、それに基づいて、希求される設計プロセスの分散化における構造と協調処理の形式を導出する。なお、第 2 報⁽⁵⁾では、それを受けた設計操作のためのモデルの提案とプロセス管理についての検討

* 原稿受付 2001 年 5 月 7 日

^{*1} 正員, 大阪大学大学院工学研究科 (〒 565-0871 吹田市山田丘 2-1).

^{*2} 富士写真フイルム (株) (〒 250-0193 神奈川県南足柄市中沼 210)
(元: 大阪大学大学院工学研究科).

Email: fujita@mech.eng.osaka-u.ac.jp

を行い、さらに、第3報⁽⁶⁾では、一連のモデルをエージェント方式により実装した上で、設計支援システムのプロトタイプを航空機の基本設計へ展開した例を示して、本研究の妥当性を検証する。

2 大規模システムにおける設計問題の形式化

2.1 規模性に向けての形式化 上述のように、分散協調型設計支援システムは大規模なシステムについての設計プロセスの全体を統合的に支援しようとするものである。一般に、設計についての理論を展開するには、問題を記述する上での何らかの基盤を設ける必要がある。それに向けての有効な方策の一つは、対象とする要素やシステムの種類を具体的に限定した上で、それについての固有領域における術語や理論を基盤とすることである。しかしながら、設計における規模性は様々な領域を横断する多様な内容によるものであるために、上記の方策に従って対象領域を限定することは規模性に起因する課題を取り上げる上で適当ではない。一方、領域を固定せずに多様な内容を対象としたままで規模性による課題を取り上げようとするれば、多様性と規模性の両方を同時に取り上げる必要が生じることから、厳密な考察は現状では困難を極める。これらを踏まえて、本研究では、分散協調型の設計プロセスが必要となる様々な理由のなかでも、設計対象の規模そのものに起因する必要性に着目するために、次項以降に示す形式化を前提とすることにする^{†1}。

なお、問題の規模はアルゴリズムを構成する上での重要な視点であり、その効率が問題の規模に対して理論的に論じられることもある。しかしながら、ここでの設計問題についての規模性はその種の規模性とは異質なものである。すなわち、設計問題における規模に起因した困難性は、限定された時間の範囲内ではそれを克服して絶対的な意味での最適解を得ることのできないという種類のものであって、限定された状況のもとでは、その代わりに何らかの優れた満足解⁽⁷⁾を導出することが求められているというものである。

2.2 集中定数系としての表現 具体的な議論を展開するに当たって、本研究では、設計対象(と対応する設計問題)を以下のように表現できるものと仮定する。

あるシステムを設計する上で関係するすべて

の内容は一連の連続変数として表すことができ、そのような変数の総数は膨大である。また、関数として表現されるそれらの間の関係式はかなりなめらかである。

この仮定は、前項で述べた着眼点のもと、対象システムの内容についての記述を、個別領域に依存させることなく、高々有限個の連続変数とそれらの上に定義されるなめらかな関数というものに封じ込めようとするものでもあり、その上で、設計問題の規模や分割された部分問題の規模を変数の数によって測定できるようにして、規模性や関連する困難性が容易に扱えるようにするためのものである。つまり、仮定の形式は極めて単純であるものの、規模性は拡大可能であり、上述の目的に向けては適切なものであると考えられる。

なお、関数のなめらかさは、ある部分での設計変更が他の関連する部分に及ぼす影響が何らかの方法で、ある程度、予測できることを意味してしている^{†2}。これについても、大規模システムの具体的な状況を考えれば、比較的妥当なものであると考えられる。

2.3 変数のタイプとそれらの間の関係式 前項の仮定における一連の変数に対して、設計が要求される機能を実現する実体の像を生成する過程であることを認めれば、一連の変数を以下のように分類することができる^{†3}。

決定変数 (decision variables) ^{†4} … 設計対象であるシステムがどのようなものであるかを表す変数であり、設計において決定できる内容に相当する。 x と表す。

性能変数 (performance variables) … 設計対象であるシステムがどのように振る舞うかに対応する変数であり、直接的な操作の対象とはなり得ない。 z と表す。

媒介変数 (intermediate variables) … 設計が様々な部分に分割される場合には、決定変数と性能変数との間を結び付けている関係式も様々な関係式に分割される。その際に、個別の部分関係式の間に介在する変数を媒介変数と呼び、 y と表す。

以上の分類のもとで、アナリシスを規定されたシステムの振舞いを模擬することとして位置付ければ、それ

^{†1} ただし、大規模なシステムの設計問題における様々な性質と対峙するためには、具体的な内容に依存することも本質的である。それに向けては、まずは、特定の内容に依存した個別的な議論を避けた上で、一般的なモデルの導出を行った後に、将来課題として多様な内容を含む具体的な設計問題についての研究に向うことが妥当な道筋の一つであるとの視点が本研究の立場である。

^{†2} ここでの「なめらかさ」の意味は必ずしも厳密なものとはならない。例えば、第2報⁽⁵⁾では、設計操作についての対象モデルの線形化近似に基づいた支援法を提案するが、それを前提とすれば、「なめらかさ」の意味は諸関数が少なくとも線形化近似が有効であるための性質を満たしていることに対応するということもできる。

^{†3} この分類は文献⁽⁸⁾⁽⁹⁾での用語を参考にしたものである

^{†4} これを設計変数と呼ぶ場合もあるが、混乱を避けるために、本研究では敢えて決定変数という呼称を用いる。

に相当する処理は既知の決定変数 x に対して以下の計算式により性能変数 z を求める行為として規定することができる。

$$z = f(x) \quad (1)$$

さらに、媒介変数は決定変数と性能変数の間に位置するため、式(1)のアナリシにおける方向性は媒介変数に関して以下のように分解することができる。

$$\left. \begin{aligned} y_i &= f_{y_i}(x_{y_i}, y_{y_i}) \\ z_j &= f_{z_j}(x_{z_j}, y_{z_j}) \end{aligned} \right\} \quad (2)$$

ここで、 f_{y_i} は決定変数の部分ベクトル x_{y_i} と媒介変数の部分ベクトル y_{y_i} からある媒介変数 y_i の値を算出する部分関数、 f_{z_j} は決定変数の部分ベクトル x_{z_j} と媒介変数の部分ベクトル y_{z_j} からある決定変数 z_j の値を算出する部分関数となっている。つまり、逆に言えば、式(1)と式(2)は $f(x)$ が f_{y_i} や f_{z_j} などの集合から構成されていることを示している。

2.4 既存の設計理論と形式化との関連 本項では、議論の準備として、設計に関わる既存の理論や方法のもとで形式化の意味を改めて解釈する。

吉川による一般設計学⁽¹⁰⁾は、設計を与えられた機能を実現する実体の像を生成するという逆問題として位置付けた上で、概念設計におけるその状況を位相空間において定義したものである。本研究は、概念設計ではなく設計問題の規模性に関心があることから、集中定数系に着目しているが、逆問題という視点は式(1)において重要である。一般設計学では、完全な設計を行うためには存在し得るすべての設計結果である実体についての完全な知識が事前に必要であるとしており、これをさらに展開すれば、そのような知識は記述不可能であることから、絶対的な意味での完璧な設計は不可能であるということになる。この不可能性は、本研究での集中定数系による形式化のもとでは、 $z = f(x)$ が陽関数として完全に記述されているとしても、その逆関数である $x = f^{-1}(z)$ を有限の時間とコストで定めて設計解を得ることは不可能である、ということに転写される。なお、この帰結そのものは、一般設計学に拠らずとも、数学的な方程式の性質により、膨大な変数により記述される系においては自明ではある。

Simon は設計についての基本的な性質として「満足化 (satisfying)」と「準分解可能性 (nearly decomposable systems)」という2つの概念を示している⁽⁷⁾。前者は、設計において最適解を求めることは不可能であって、設計解は評価における様々な評価指標間の調整による満足解として求められるものであり、さらに

は、設計要求として与えられる設計目標もあくまで一時的なものであって、満足解に向けての調整の過程で目標そのものも調整されるということを示している。後者は、システムについての内容は一連の部分に分割することができるが、それらの間の相互関係を完全に消し去ることはできないことを示しつつ、優れた満足解を得るためには、システムに潜む複雑さを緩和できるシステムの分割を適切に導入することが重要であることを示している。これらについても、設計解は限定合理性 (limited rationality)⁽¹¹⁾のもとでは満足解にならないが、式(1)を適切に式(2)の形式に分割して利用することができれば、より優れた設計解を得ることができる、ということに転写することができる。

これらは何れも、設計における問題解決においては時間限定性が重要な観点であることを意味している。

3 時間限定性のもとでの設計知識のタイプと性質

3.1 設計局面による設計知識の分類 設計プロセスを典型的な局面に分けて考えることは有益であり、Dym は設計プロセスを以下の4つの局面の繰返しであるとしている⁽¹²⁾。また、前節での変数の分類に関連づけると、それぞれの内容は以下のように解釈することができる。

設計 (design) … 決定変数 x を (仮に) 設定する。

検証 (verify) … 評価のための個別項目である性能変数 z の値を求める。

評価 (critique) … 求めた性能変数 z の値について何らかの総合的な判断を行う。

修正 (modify) … 評価の結果に基づいて、決定変数 x の値を調整する。

さらに、この分類を式(1)の関係式における $f(x)$ の方向性と関連させれば、設計に関連する知識を以下のように分類することができる。

前向き知識 (forward-chaining knowledge) … 検証の局面において $f(x)$ を計算することに対応する知識であり、いわゆるアナリシスのための知識となる。

後向き知識 (backward-chaining knowledge) … 設計と修正の局面で用いられる知識であり、形式的には $f^{-1}(z)$ に対応する。しかしながら、前述のように $f^{-1}(z)$ の陽関数としての表現を明示的には得ることはできない。

評価知識 (critique knowledge) … 評価の局面で用いられる知識であり、性能変数 z が相互の調整に向けて適切であるかどうかを判定するために用いられる。いわゆる満足化において z を洗練化する上で不可欠なものであるが、その内容は $z = f(x)$ とい

う関係式に関連づけられるものではない。

設計におけるシンセシスは、以上の3種類の知識のすべてを必要とする。例えば、シンセシスのなかでも、もっとも簡単な形式である設計最適化は数理計画法のアルゴリズムとアナリシスモデルから構成されている。前者は後向き知識に相当し、後者は前向き知識そのものであるが、加えて、計算の結果に対して何らかの評価を行う必要があることも自明のとおりである。

3.2 決定変数から性能変数への写像における性質

まず、上述の知識分類のうち前向き知識、つまりアナリシスの性質について考える。アナリシスにおいて物理的な現象をより良く理解するためには、それに対応するより精密なモデルを導出して用いることが必要であり、高い精密さを希求すれば、それに対応した資源や時間が必要になることとなる。これらの内容を設計シンセシスにおける決定変数から性能変数への写像のもとで考えた場合、逆写像が明示的ではない状況のもとでの順写像におけるモデル操作での効率性に着目すれば、精密なモデルのみならず、そうではないモデルにもその意義を見いだすことができる。つまり、あらかじめ定められた時間の枠内で設計を行うことを考えた場合、広範囲な内容についての大域的な設計を進める上では精密ではない知識が有用である一方で、設計解の精度を保証するためには部分に関する精密な知識が不可欠となる⁽¹³⁾ということになる。

3.3 決定変数と設計知識における粒度 上記の内容は、粒度(granularity)⁽¹⁴⁾という概念を持ち込むことにより、より明確となる⁽¹³⁾。式(1)の形式における決定変数ベクトル x については、その要素の全体集合が対象システムを精密に表現するためのものである一方で、何らかの粒度レベルを設定すれば、そのレベルでの内容を代表する一連の要素に対応する部分変数ベクトルを定義することができるはずである。このとき、一連の部分変数ベクトルを粒度レベルの粗いものから詳細なものへと順に $x_{g1}, x_{g2}, \dots, x_{gi}, \dots$ と記述すれば、相互の関係は以下ようになる。

$$\{x_{g1}\} \subset \{x_{g2}\} \subset \dots \subset \{x_{gi}\} \subset \dots \quad (3)$$

なお、 $\{\cdot\}$ は変数ベクトル \cdot の要素の集合を表す。

さらに、対象表現が各部分変数ベクトルによって多重に規定された場合には、対応して、性能を求めるための前向き知識 $z = f(x)$ も各粒度レベルにおいて多重に存在することになる。そこで、 $x_{g1}, x_{g2}, \dots, x_{gi}, \dots$ のそれぞれに対する前向き知識を、それぞれに $f|_{x_{g1} \rightarrow z}, f|_{x_{g2} \rightarrow z}, \dots, f|_{x_{gi} \rightarrow z}, \dots$ と記述することにし、さらに、それぞれの算出する z の値を $z^{(g1)}, z^{(g2)},$

$\dots, z^{(gi)}, \dots$ として区別することにする。

3.4 前向き知識におけるコストと忠実度 前向き知識の多重性を踏まえれば、それらの運用や結果に相違が生じるはずである。まず、処理コストに関しては以下の関係が成り立つものと考えることができる。

$$C(f|_{x_{g1} \rightarrow z}) < C(f|_{x_{g2} \rightarrow z}) < \dots < C(f|_{x_{gi} \rightarrow z}) < \dots \quad (4)$$

なお、 $C(\cdot)$ は処理 \cdot を行うためのコストを表す。

また、式(1)が示すように、性能変数は前向き知識による帰結であるため、前提となる決定変数の記述能力の相違に対応して、結果における忠実度(fidelity)も相互に異なったものになるはずである。その際の性能変数 z_j の多重な値における忠実度についての関係は、以下になるものと想定することができる。

$$\Phi(f|_{x_{g1} \rightarrow z_j}) < \Phi(f|_{x_{g2} \rightarrow z_j}) < \dots < \Phi(f|_{x_{gi} \rightarrow z_j}) < \dots \quad (5)$$

ここで、 $\Phi(\cdot)$ は \cdot の値がどの程度信頼できるかを、つまり、忠実度を示すものである。

3.5 後向き知識の可能性 次に、後向き知識について考える。前述のように、式(1)の関係からすれば、後向き知識は形式的には前向き知識の逆関数である。一般に、最も粗い粒度レベルにおける $f|_{x_{g1} \rightarrow z}$ の内容は、経験的な設計公式やマクロな物理モデルに基づいているものとすることができ、簡便な代数式の形態をとる。これに対して、ある程度以上の詳細な粒度レベルにおける $f|_{x_{gi} \rightarrow z}$ の内容は、数値シミュレーションや極端な場合は実験によることになる。このような前向き知識における相違は後向き知識において決定的となる。つまり、操作性と忠実度を両立させた前向き知識の逆関数としての後向き知識の存在を期待できることは極めて稀であり、後向き知識は設計プロセスにおいて何らかの試行錯誤的な処理として暗黙的に形成されるということになる。

3.6 後向き設計知識における支配性 効率的な試行錯誤を実現するために、 $f^{-1}(z)$ に相当する内容が何らかの代替(例えば、近似式)によって知識化されたと考えることにはあまり矛盾はない。仮に、代替的な後向き知識が存在するとして、 $f^{-1}(z)$ に対する代替関係式を $\hat{f}^{-1}(z)$ として表すことにする。

そのような代替知識の内容や形態は、航空機の設計問題についての検討結果⁽⁶⁾に従えば、ある粒度レベル g_i における後向き知識に関わる決定変数 x_{gi} を $\{\hat{x}_{gi}\} = \{x_\ell | x_\ell \in \overline{\{x_{gi-1}\}} \cap \{x_{gi}\}\}$ という関係のもとで x_{gi-1} と \hat{x}_{gi} という2つの部分に分割した場合に、 $\hat{f}_{gi}^{-1}|_{z_j \rightarrow x_{gi}}$ の具体的な内容は、一般に、 $\hat{f}_{gi}^{-1}|_{z_j \rightarrow x_{gi-1}}$

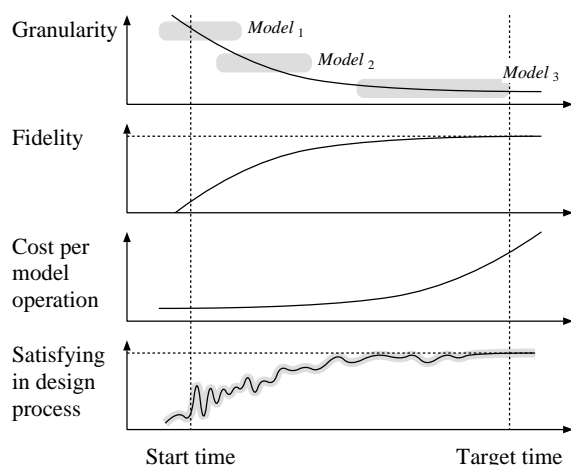


Fig. 1 Time-boundness of design process

と $\widetilde{f_{g_i}^{-1}}|_{z_j \rightarrow \widehat{x_{g_i}}}$ に分けて整備されていることがわかる。より厳密には、 $\widetilde{f_{g_i}^{-1}}|_{z_j \rightarrow \widehat{x_{g_i}}}$ を用いるためには、前もって $x_{g_{i-1}}$ の値が必要であり、 $\widetilde{f_{g_i}^{-1}}|_{z_j \rightarrow \widehat{x_{g_{i-1}}}}$ は $\widetilde{f_{g_i}^{-1}}|_{z_j \rightarrow \widehat{x_{g_i}}}$ に対して支配的である (dominate) とすることができる^{†5}。つまり、 $x_{g_{i-1}}$ の値は $\widehat{x_{g_i}}$ についての操作に先立って決定しておかなければならないということになる。

3.7 時間限定下での設計プロセス 以上を総合すれば、時間限定下での設計プロセスは図1のようなものであると期待することができる。すなわち、まず、設計の初期段階では、設計知識の支配性から支配的な決定変数を他の変数に先立って決定する必要があるため、粒度レベルは粗く性能変数の忠実度は低く設定される。これによって、設計要求を正確に満足しているかどうかについてはやや曖昧となるものの、低い設計操作のコストのもとで多くの試行錯誤が行えるようになり、設計解を大域的に探索することが可能になる。これに対して、設計の最終段階においては設計解が設計要求を満足することが十分に高い忠実度で判定される必要があることから、設計の進行に伴って徐々に粒度のレベルを詳細なものにしていき、忠実度を向上させていく必要がある。それと同時に、ある段階を過ぎると設計操作のコストが大幅に増加するため、早期に、例えば中間段階までに、個別の不具合に対応するための小規模な設計変更の余地を残しつつ、設計解の大局的な傾向を確定させておくことも必要となる。

^{†5} このようなモデルの階層的な支配性についての考え方は、忠実度 (fidelity) と関連しつつ、設計最適化において大域的な探索のための粗いモデルと精密な探索のために詳細なモデルを同時並行的に用いようとする考え方⁽¹⁵⁾とも通じるものである。

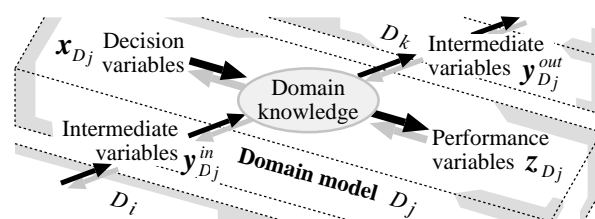


Fig. 2 A domain and its surroundings

4 設計プロセスの分割における構造

4.1 大域モデルと領域モデル 図1のような設計プロセスの状況を明確にするためには、さらに設計知識全体の構造について吟味する必要がある。一般に、大規模なシステムの設計においては、設計対象の全体を粗く代表する少数の変数が存在し、それらは主要目と呼ばれる。航空機設計の場合⁽⁶⁾には、胴体長、主翼幅、抗力係数、最大離陸荷重などが主要目に相当する。一方、それ以外の内容は個別的にそれぞれの技術領域に依存した内容になっている。両者の性質が大きく異なっていることから、一連の変数と関係式を以下の2つに分割することにする。

大域モデル (global model) … 主要目とそれらに関わる内容。これは設計における起点となるものであり、個々の設計問題において唯一のモデルが存在する。

領域モデル (domain models) … 大域モデル以外の内容。その内容は膨大であるため、設計における問題解決を分散させるのであれば、背後にある技術領域に対応して適切なサイズの部分に分割することになる。

なお、ここでのモデルとは一連の変数とそれらの間の関係式としての設計知識とをまとめたものである。また、後者の領域モデルが技術領域に基づいて分割されるという考え方は、いわゆる準分解可能性⁽⁷⁾に対応するものでもある。

4.2 領域モデルの水平分割 領域モデルはそれぞれの部分での内容が内部において相互に強い関連性を持っていることから、それぞれのもとで自律的な設計が行えることが望ましい。そのためには、各領域モデル D_j が、図2にも示すように、独自の権限で操作できる決定変数 x_{D_j} とそれらに主に依存して定まる性能変数 z_{D_j} を含んでいる必要がある。一方、分割を行っても、領域間には何らかの関連性が残ることから、領域モデルの境界にはそれに対応した媒介変数が介在していることになる。図にも示すように、媒介変数には、決定変数の側で用いられるもの $y_{D_j}^{in}$ と性能変数の

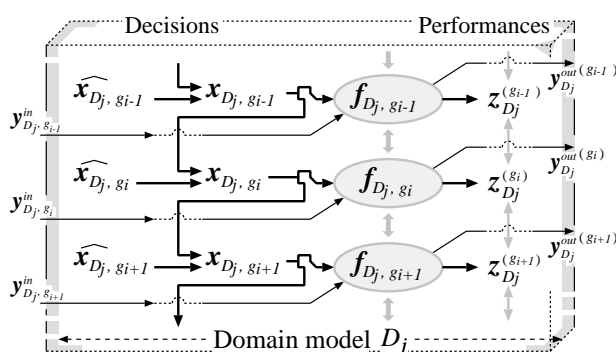


Fig. 3 Hierarchical linking among models

側で定められるもの $y_{D_j}^{out}$ とがある．ここでは，前者を媒介入力変数，後者を媒介出力変数と呼ぶことにする．このとき，ある領域における媒介出力変数は他の領域において媒介入力変数となっており，ある領域における媒介入力変数は他の領域において媒介出力変数として定められるものに対応することになる．

なお，個別の領域は，大まかに，設計対象の実体としてのある部分についての内容を決定するためのものと，設計対象の発現するある機能項目についての内容を保証するためのものとに大別することができる．航空機設計の場合⁽⁶⁾には，主翼，胴体などについての領域が前者のタイプのものであり，航続性能，推進性能などについての領域が後者のタイプのものとなる．

4.3 設計プロセスの垂直分割 設計知識における粒度と支配性は，領域による水平分割とは別に，個別の領域モデルが複数の層に分割できることを示している．図3はある領域モデル D_j における分割の状況を前向き知識に着目して模式的に示したものである．

一連の設計操作においては，まず，粒度レベル g_{i-1} において支配的な決定変数 $x_{D_j, g_{i-1}}$ を決定し，対応して，前向き知識 $f_{D_j, g_{i-1}}$ により z_{D_j} に対する推定値となる $z_{D_j}^{(g_{i-1})}$ を求めた後，設計操作の対象を次の粒度レベル g_i に移行させる．そのレベルでは，まず， $z_{D_j}^{(g_{i-1})}$ や $x_{D_j, g_{i-1}}$ の内容をもとに，新たに決定変数に加える $\widehat{x_{D_j, g_i}}$ の内容を決定し， $x_{D_j, g_{i-1}}$ と合わせることにより x_{D_j, g_i} を形成する．続いて，前向き知識 f_{D_j, g_i} を x_{D_j, g_i} と関連する媒介入力変数 y_{D_j, g_i}^{in} に対して適用し， $z_{D_j}^{(g_i)}$ と媒介出力変数 $y_{D_j}^{out(g_i)}$ の値を計算する．この結果，得られる $z_{D_j}^{(g_i)}$ は $z_{D_j}^{(g_{i-1})}$ に比べてより高い忠実度を持つことになる．このような粒度レベル g_i における処理は，次の粒度レベル g_{i+1} においても，同様に継続されることになる．

以上の操作は粗いレベルから詳細なレベルへと再帰的に連鎖することになるが，最も粗い粒度レベルでの

操作は大域モデルと連結されており，最も詳細な粒度レベルの操作は設計の終了に対応するものとなる．

5 分割された設計タスク間でのゴールの連鎖

5.1 評価における悪構構性 前節の内容は前向き知識とそのもとでの後向き知識に着目したものであるが，設計においてはむしろ評価の局面が本質的である．設計は悪構問題と呼ばれることがあるが，前出の満足化⁽⁷⁾とデザインスパイラル⁽¹⁶⁾はそのもとでの評価局面の意義をよく表している．いわゆる設計目標は，要求条件として与えられるものの，目標が過剰であって元来満足することが不可能であったり，相互のバランスが事前には明らかではなかったりするために，具体化された設計解の状況を検討しつつ目標を緩和するなどの洗練化が不可欠であるとされている．また，洗練化は試行錯誤的なものであるとされており，設計における時間限定性，それに付随する設計知識の多重性や操作性，領域間での設計知識の関連性のもとでは，図1にも示したように，様々な粒度レベルを通じて段階的に行われることになる．これらのことは，それぞれの性能変数が様々な粒度レベルに渡って管理されるのみならず，相互の間での調整をも含めて協調的に管理される必要があることを意味している．

5.2 段階的な設計プロセスのためのゴール概念

上記の管理を実現するためには，性能変数における希求値(目標)と算出値の相違や曖昧性を一時的に許容しつつ管理する必要がある，さらには，各種項目間の相互の関係における最終的な状況を予測したり，決定変数の修正や設計要求の満足化に向けて何らかの中間的な指標を設定することなどが必要となる．

それに向けては，本研究では，管理の対象を性能変数の取り得る値の範囲であるとした上で，範囲やその関連情報を一括して「ゴール(goal)」と呼ぶことにする^{†6}．なお，範囲は，設計プロセスの進展を通じて段階的に絞り込まれて，最終的には数値としての設計解に収斂する必要がある．

ここまでの議論をまとめれば，各性能変数 z_ℓ についてのゴールは以下の情報を含んでいる必要がある．

- 各粒度レベル g_i において算出された値 $z_\ell^{(g_i)}$ ．
- 目標としての希求される結果についての選好範囲(preferred range)．以下では \tilde{z}_ℓ と表記する．本研

^{†6} この呼称はゴールプログラミング⁽¹⁷⁾からの借用である．また，Wardらはコンカレントエンジニアリングにおける視点のもとでトヨタ自動車での製品開発プロセスに対する分析を行っている⁽¹⁸⁾が，その論点は，設計の早い段階で各設計項目の取り得る範囲を集合として公開していくことにより，後続する設計処理をより早い段階で着手できるようになっているというものである．

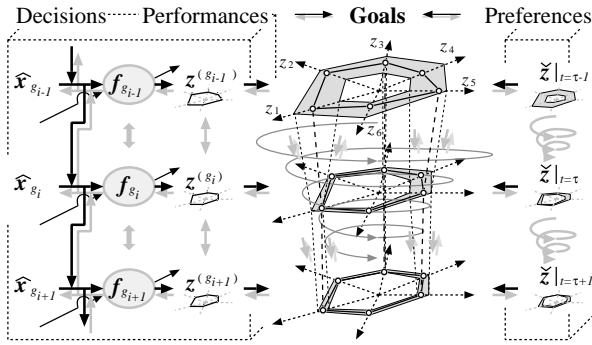


Fig. 4 Squeezing goals

究では、集中定数系における各関数がなめらかであるという前提のもとで、 $\tilde{z}_\ell = [\tilde{z}_\ell^L, \tilde{z}_\ell^U]$ という形式で定めることができるものとする。

ここで、 $z_\ell^{(g_i)} \in \tilde{z}_\ell$ という関係は満たされるべきものではあるが、必ずしも満たされているとは限らない。

5.3 ゴールの段階的な絞り込み 図4は図3に示した複数の粒度レベルのもとでのゴールの管理についての状況を模式的に示したものである。

まず、ある段階 $t = \tau - 1$ において、ある粒度レベル g_{i-1} のもと、仮に選好範囲を $\tilde{z}_{|t=\tau-1}$ として設定した後、それに基づいて決定変数 $\widehat{x}_{g_{i-1}}$ を決定し、それらを合わせることで $x_{g_{i-1}}$ を定める。これによって $f_{g_{i-1}}$ から $z_{g_{i-1}}$ を求めることができるようになり、続いて、評価知識により $z_{g_{i-1}}$ と $\tilde{z}_{|t=\tau-1}$ とを比較することになる。この決定内容に対する評価結果が次のレベルでの設計が首尾よく行える見通しを与えるものであれば、次の段階 $t = \tau$ では、選好範囲を $\tilde{z}_{|t=\tau}$ へと絞り込んだ上で、粒度レベルを詳細なものへと遷移させることになる。一方、そのような見込みが得られないなど、評価結果に不足があれば、 $\widehat{x}_{g_{i-1}}$ を修正することになる。以上の処理は、後続する段階において $\tilde{z}_{|t=\tau}$ から $\tilde{z}_{|t=\tau+1}$ へと同様に展開され、再帰的に繰り返される。

上記の操作は段階的な詳細化であることから、性能変数のゴールに関しては以下の関係式が成り立つことが望ましいと考えられる。

$$\cdots \supset \tilde{z}_{|t=\tau-1} \supset \tilde{z}_{|t=\tau} \supset \tilde{z}_{|t=\tau+1} \supset \cdots \quad (6)$$

なお、図4においてゴールを通じて先導される一連の操作が不適切な結果に陥ることは完全には否定することはできない。このため、場合によっては、設計プロセスの早い段階、つまり、粗い粒度レベルに戻って選好範囲を再検討することが必要となることもある。この意味において、設計操作の対象となる粒度レベル

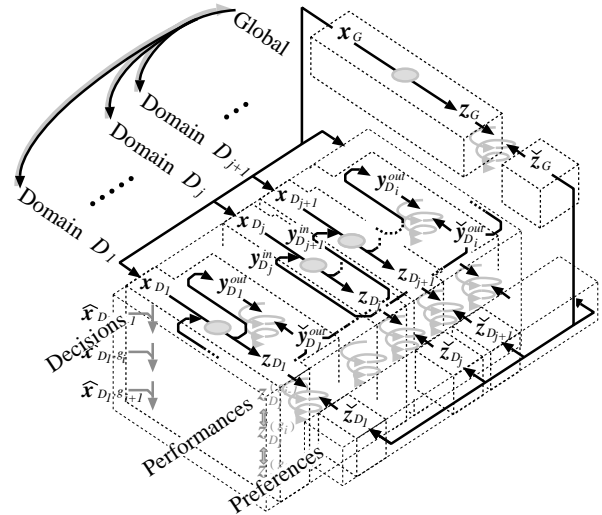


Fig. 5 Infinitely cascaded goals

は詳細にしたり粗くしたりすることを繰り返しながら遷移していくことになる。

5.4 領域間における媒介変数の役割 前項の内容は領域内における設計評価についてのものであるが、分散協調型の設計プロセスに対応するためには図2に示した領域分割と領域間での並列性にも対応する必要がある。領域間相互の媒介変数を介した依存関係のもとでは、ある領域においては、前向き処理を行うためには媒介入力変数の値が必要となり、後向き処理を行うためには媒介出力変数について希求される選好範囲が必要となる。しかしながら、これらは他の領域から与えられるものであるため、分散化のもとではいわゆるデッド・ロックを生じることになる。これを解消して並列性を実現するために、それぞれの処理において前提条件となる情報を、随時、暫定的に開示していく必要がある。そこで、媒介変数についても性能変数と同様にゴールの概念を導入して、相互のあそび(余裕)として機能させることを考える。つまり、ゴールという概念を媒介変数にまで拡張する^{†7}。

5.5 ゴールの連鎖 図5は分散協調型の設計プロセスにおいて性能変数と媒介変数についてのゴールが連鎖している様子を模式化したものである。すなわち、まず、設計条件が大局的に与えられた後、それに対応するゴールが \tilde{z}_G として設定される一方、対応して x_G が決定された後、そのような内容が各領域に伝達される。これを受けて、それぞれの領域 D_j では \tilde{z}_G をもとにして \tilde{z}_{D_j} が想定され、 x_{D_j} についての設計が行われる。この領域 D_j における操作は、図2の状況のも

^{†7} ゴール概念は第2報⁽⁵⁾でより明確に定義されるとともに、性能変数と媒介変数のみならず決定変数にまで拡張されることになる。

と、 $y_{D_j}^{in}$ と $y_{D_j}^{out}$ を通じて他の領域と相互に作用し合っている。例えば、 x_{D_j} の決定は z_{D_j} の内容を実現すべく行われるが、その決定は $y_{D_j}^{in}$ が適当なものであることを前提としている。一方、 $y_{D_j}^{in}$ についての前提は他の領域 D_i ($i \neq j$)での $y_{D_i}^{out}$ についての選好範囲 $y_{D_i}^{out}$ になっている。したがって、領域 D_i は同時に独自の性能変数についての選好範囲 z_{D_i} を持っていることから、ここでの操作は z_{D_i} と $y_{D_i}^{out}$ の両方に対して行われる必要があるということになる。以上の状況はすべての領域において生じるものであるため、選好範囲は無限に連鎖していることになる。この無限連鎖を現実的に回避して限定された時間のもとで設計を遂行するためには、各種の変数に対してある種のあそびとしてのゴールを導入しつつ、図1に示したように粒度レベルを遷移させながら設計プロセスを展開する必要があるということが本報での重要な帰結である。

6 ゴールに駆動された協調処理の様相

6.1 協調処理の形態 導出されたゴールの考え方と分割されたタスク間での協調処理の必要性は個別的な状況での協調処理の形態を暗示している。その形態は以下の3つに類別することができる。

領域間協調 (inter coordination) … 分割された領域モデルは相互に媒介変数を介して連結されていることから、直接的に連結された各領域は相互に協調することが必要である。

大域的協調 (global coordination) … 領域間協調のみでは、全体としてのバランスに優れた設計解を得ることを保証することはできない。このため、大域的な視点から全体的な方向付けを行ったり、競合の解消を導いたりすることも必要である。すなわち、何らかのスーパーバイザー機能によって設計解の全体を調整する必要がある。大域モデルは設計解の初期設定のみならずこのような方向でも用いられるべきものである。

領域内協調 (inner coordination) … 個別の領域 D_j においては、決定変数と対応する領域モデルが粒度レベルに対応して多重に定められていることから、相互の関係を内部で管理する必要がある。つまり、どのようなタイミングでどのような根拠によって、操作の対象となる粒度レベル、すなわち \hat{x}_{g_i} を切り替えて、設計処理を展開するかを管理する必要がある。

6.2 設計プロセスの管理 以上の各協調処理を具体化するためには、まずは、分割されたタスク間での何らかの交渉メカニズムが必要である。それに加え

て、粒度レベルやゴールの考え方のもとでは、以下の性質を伴ったプロセス管理のためのメカニズムも必要となる。

- 設計があらかじめ想定された時間で完了するように、各領域における粒度レベルの移行を管理するメカニズムが必要となる。このスケジューリングは、設計知識や個別の設計状況の伴っている曖昧性に対応すべく、動的かつ頑強に行われることが求められる。
- スケジューリングにあたっては、粒度レベルの遷移に応じて、ゴールについての情報がどのように変化していくか、例えば、 $f(x_{D_j, g_{i-1}})$ と $f(x_{D_j, g_i})$ との差異のもとで $z^{(g_{i-1})}$ に対して $z^{(g_i)}$ がどのようなものになるかなどを予測するとともに、個別の操作に対してその結果を得るために要するコストをあらかじめ見積もることも必要である。

7 結 言

本報では、分散協調型設計支援システムを実現する上での基盤となすことを目的として、大規模なシステムの設計プロセスに潜む構造について論じた。ここでの議論は設計対象を大規模な集中定数系として表現できるとする形式化に基づいたものであるが、この想定の導入により設計プロセスにおける分散化や協調操作における典型的な状況や形態を抽出することができた。なお、続報⁽⁵⁾⁽⁶⁾では以上を基盤として、設計操作のためのモデルとプロセス管理における要件、具体的な設計支援システムの構築方法について論じる。

なお、本研究の一部は日本学術振興協会未来開拓学術研究推進事業(96P00702)の援助によるものである。

文 献

- (1) Wong, A. and Sriram, D., SHARED : An Information Model for Cooperative Product Development, *Research in Engineering Design*, Vol. 5, No. 1, (1993), pp. 21-39.
- (2) Park, H., Cutkosky, M. R., Conru, A. B. and Lee, S. H., Agent-Based Approach to Concurrent Cable Harness Design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM)*, Vol. 8, No. 1, (1994), pp. 45-61.
- (3) 藤田・赤木, システム構造に着目したエージェント方式による分散並行型設計支援システムの構成方法, 日本機械学会論文集 C 編, Vol. 65, No. 630, (1999), pp. 813-820.
- (4) 赤木・藤田, 設計におけるコンピュータ利用とシンセシス, 日本機械学会論文集 C 編, Vol. 60, No. 579, (1994), pp. 3591-3601.
- (5) 藤田・菊池, 分散協調型設計支援システムのためのプロセスモデルと操作モデル(第2報: 設計操作のためのモ

- デルとプロセス管理), 日本機械学会論文集 C 編, Vol. 68, No. 666, (2002), (論文番号 01-0552).
- (6) 藤田・菊池・南, 分散協調型設計支援システムのためのプロセスモデルと操作モデル (第 3 報: エージェント方式による実装と航空機の基本設計への展開), 日本機械学会論文集 C 編, (投稿中: 論文番号 01-0553).
 - (7) Simon, H. A., *The Science of Artificial (3rd Edition)*, (1996), MIT Press, [邦訳: 稲葉・吉原, システムの科学, 第 3 版, (1999), パーソナルメディア].
 - (8) Kusiak, A. and Wang, J., Dependency Analysis in Constraint Negotiation, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 9, (1995), pp. 1301-1313.
 - (9) Sreeram, R. T. and Chawdhry, P. K., A Unified Scheme for Conflict Negotiation in a Multi-Agent Decision Process, *Advances in Concurrent Engineering CE99 — Proceedings of 6th ISPE International Conference on Concurrent Engineering*, (1999), pp. 129-141.
 - (10) 吉川, 一般設計学序説 — 一般設計学のための公理的方法, 精密機械, Vol. 45, No. 8, (1979), pp. 906-912.
 - (11) Russell, S. and Wefald, E., *Do The Right Thing — Studies in Limited Rationality*, (1991), The MIT Press.
 - (12) Dym, C. L., *Engineering Design — A Synthesis of Views*, (1994), Cambridge University Press.
 - (13) 藤田・赤木, 形態の段階的詳細化と視点管理による製品設計の枠組とその展開 (第 1 報: 基本概念と実装方法), 日本機械学会論文集 C 編, Vol. 62, No. 600, (1996), pp. 3347-3354.
 - (14) Hobbs, J. R., Granularity, *Proceedings of the Ninth IJCAI*, (1985), pp. 1-4.
 - (15) Knill, D. L., Giunta, A. A., Baker, C. A., Grossman, B., Mason, W. H., Haftka, R. T. and Watson, L. T., "Response Surface Models Combining Linear and Euler Aerodynamics for Supersonic Transport Design," *Journal of Aircraft*, Vol. 36, No. 1, (1999), pp. 75-86.
 - (16) Buxton, I. L., *Engineering Economics and Ship Design*, (1971), British Ship Research Association (BSRA) Report.
 - (17) Lee, S. M., *Goal Programming for Decision Analysis*, (1972), Auerbach Publishers.
 - (18) Ward, A., Sobek, D. K. II, Cristiano, J. J. and Liker, J. K., Toyota, Concurrent Engineering, and Set-Based Design, *Engineered in Japan*, (1995), pp. 192-223, Oxford University Press.