The University of Osaka
Institutional Knowledge Archive

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）
Author(s)	藤田, 喜久雄; 菊池, 慎市; 南, 雄太郎
Citation	日本機械学会論文集 C編. 2002, 68(670), p. 1910-1918
Version Type	VoR
URL	https://hdl.handle.net/11094/3228
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

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

The University of Osaka

日本機械学会論文集 (C 編)
68 巻 670 号 (2002-6)

論文 No. 01-0553

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) *

藤田　喜久雄 *1，菊　池　慎　市 *2，南　　雄 太 郎 *3

Computational Models for Concurrent Design Process Support (3rd Report: Agent-Based Implementation and Application to Aircraft Design)*

Kikuo FUJITA*4, Shin'ichi KIKUCHI, and Yutaro MINAMI

*4 *Deptpartment 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 toward the development of agent-based distributed design support systems. This third report presents the agent-based implementation of the computational models for design process structure, design operation, and process management. In the implementation, the divided tasks are modeled as agents, design information shared among agents are managed through the exported and imported variables in both sides respectively, the task on a specific agent is split into the core and mediator tasks for asynchronous information exchange among agents. The design knowledge is represented and processed in an object-oriented manner in respective agents. Furthermore, the implemented prototype system of a distributed design support system is applied to preliminary aircraft design for ascertaining the validity of computational models proposed in this research.

Key Words : Design Engineering, Design Process, Design Task Distribution, Agent-Based Concurrent Engineering, Object-Orientation, Knowledge Engineering, Aircraft Design

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を有効に支援するためには，個別的な設計処理のコンピュータ化に加えて，設計プロセスの全体からみた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を構築する必要があり，コンピュータ援用コンカレントエンジニアリングと称される試みが多方面から行われつつある．

本研究(1)(2)は，上記の動向のもと，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着目した設計支援のためのモデルを確立することを目的としている．第 1 報(1)では，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ものとの想定のもと，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態を導出した．第 2 報(2)では，それらのプロセス構造のもとでの設計対象の操作モデルを提案し，プロセス管理の方法について論じた．具体的には，分散構造のもとでの各部分問題における設計操作において自身の設計と他の部分問題との間で定義されたゴールについての相互調整を段階的に進めるための支援機能を最適化計算に基づいて構成し，そのもとで設計プロセスを管理するための方策を論じた．

本報では，第 1 報(1)と第 2 報(2)で提案した一連のモデルをエージェント方式により実装する方法とその適用例を示すことにより，それらの妥当性を検証する．具体的には，エージェント方式による設計処理の分散化，各種設計情報の共有と交換，オブジェクト指向による知識表現と設計処理のための方法をそれぞれに提案し，さらに実装したプロトタイプシステムを航空機の基本設計問題に適用した事例を示す．

* 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 富士写真フィルム (株) 生産技術部 (〒 250-0193 南足柄市中沼 210) (元：大阪大学大学院工学研究科).
*3 日産自動車 (株) 総合研究所 (〒 237-8523 横須賀市夏島町 1) (元：大阪大学大学院工学研究科).
Email: fujita@mech.eng.osaka-u.ac.jp

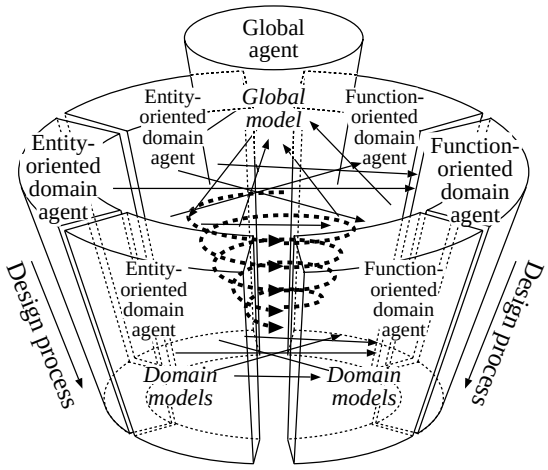


Fig. 1　Design process distribution structure

2　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

本節では準備として第 1 報[1]と第 2 報[2]の内容を本報に関連するものを中心として整理する．

2.1　設計プロセスの分散構造　設計は何らかの対象モデルをもとに展開される．なかでも，設計対象が大規模な場合には，対象モデルは対象全体の概略を代表する**大域モデル** (global model) とシステムの各部分についての具体的な内容を表す**領域モデル** (domain models) とに分割することができる．後者の一連の領域モデルは，背後にある技術領域の内容に対応して，主に設計対象の部分実体の内容についてのものと，主に設計対象が実現する個別機能に関するものとに大別することができる．なお，領域モデルへの分割は特定の設計問題の内容に依存するが，効率的な分散協調処理を実現するためには，個別内容の独立性のみならず，分割後の各モデルが担う内容量のバランスをも踏まえる必要がある．さらに，各領域モデルは，操作性と忠実性とのバランスのもとで，複数の**粒度** (granularity) **レベル** において多重に構成されているとすることができる．これらの対象モデルの構成に対して，設計プロセスにおいては，図 1にも示すように，まず，大域モデルに基づいて大まかな設計の方向が策定された後に，個別の領域モデルに基づいた部分設計が分散・協調・並行的に展開されて，粒度レベルのもとで段階的な詳細化を進めて設計が確定されていくものとすることができる．この際，部分問題相互のバランスを適切なものに維持するために，大域モデルに基づいた設計展開の大局的な方向付けについての補正も同時に行われる必要がある．

2.2　対象モデルの関係と内容　上記の内容に加えて，対象モデルの内容は大規模な集中定数系として表現できるとの想定を導入した場合，設計処理は，一連の**決定変数** (decision variable) $\boldsymbol{x}$ の決定とそれらから算出される**性能変数** (performance variable) $\boldsymbol{z}$ の調整として形式化することができる．両者の関係が $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$ という式により規定されるとするとき，領域モデルによる分割構造は，相互の間に介在する変数を**媒介変数** (intermediate variable) $\boldsymbol{y}$ とすれば，上式が決定変数の部分ベクトル $\boldsymbol{x}_{y_i}$ と媒介変数の部分ベクトル $\boldsymbol{y}_{y_i}$ からある媒介変数 y_i の値を算出する $y_i = f_{y_i}(\boldsymbol{x}_{y_i}, \boldsymbol{y}_{y_i})$ や決定変数の部分ベクトル $\boldsymbol{x}_{z_j}$ と媒介変数の部分ベクトル $\boldsymbol{y}_{z_j}$ からある決定変数 z_j の値を算出する $z_j = f_{z_j}(\boldsymbol{x}_{z_j}, \boldsymbol{y}_{z_j})$ などとして分解できることに対応する．なお，y_i の位置にあるものを**媒介出力変数** (intermediate output variable) $\boldsymbol{y}^{out}$，その他のものを**媒介入力変数** (intermediate input variable) $\boldsymbol{y}^{in}$ として区別する．また，粒度レベルのもとでの多重構造は，決定変数 $\boldsymbol{x}$ について，各粒度レベル g_i における内容 $\boldsymbol{x}_{g_i}$ が $\{\boldsymbol{x}_{g_1}\} \subset \{\boldsymbol{x}_{g_2}\} \subset \cdots \subset \{\boldsymbol{x}_{g_i}\} \subset \cdots$ という関係のもとで定められるときに，性能変数の内容が $\boldsymbol{z}^{(g_i)} = \boldsymbol{f}_{g_i}(\boldsymbol{x}_{g_i})$ として異なる操作性と忠実度のもとで多重に算出されるということに対応する．

2.3　ゴールとその内容　2.1項に示した設計プロセスを具体化するには領域モデルによる分割構造と粒度レベルのもとでの多重構造とに対応して構成された各部分問題を並行的に処理する必要がある．しかしながら，一連の部分問題は相互に干渉していて，本来は一方の設計が確定しないと他方の設計が進められない．さらに，そのような関係は重奏している．この依存関係を越えて並行的な処理を可能にするために，本研究では各種の設計内容についてある種のあそびとしての**ゴール** (goal) という考え方を導入している．

具体的なゴールの内容は，決定変数と媒介入力変数については，仮の確定値としての**決定値** (decided point)，それが最終的に確定されるべき範囲としての**決定範囲** (performed range)，他のモデルの内容に照し合せて希求されている範囲としての**推奨範囲** (recommended range) により，定義する．そのもとで，ある変数についてのそれぞれを θ_ℓ，$\bar{\theta}_\ell = [\theta_\ell{}^L, \theta_\ell{}^U]$，$\check{\theta}_\ell = [\check{\theta}_\ell{}^L, \check{\theta}_\ell{}^U]$ とするとき，$\theta_\ell \in \bar{\theta}_\ell$ と $\bar{\theta}_\ell \subset \check{\theta}_\ell$ という関係を維持しながら設計を展開していくものとする．性能変数と媒介出力変数については，対応する内容を**性能値** (performed point)，**性能範囲** (performed range)，**選好範囲** (preferred range) として定めて，同様の取り扱いを行うものとする．

2.4 設計操作とプロセス管理 以上のもとで，それぞれの部分設計は，決定変数 $\boldsymbol{x}$ についての初期値を何らかの知識に従って想定した後，媒介入力変数 $\boldsymbol{y}^{in}$ などと併せて前向き知識 $\boldsymbol{f}$ により性能変数 $\boldsymbol{z}$ や媒介出力変数 $\boldsymbol{y}^{out}$ の値を算出し，評価知識，$\boldsymbol{y}_{out}$ や $\boldsymbol{z}$ についてのゴール情報に照し合せてそれらの結果を評価した上で，$\boldsymbol{x}$ や $\boldsymbol{y}^{in}$ やそれらのゴール情報に修正を加えるという一連の操作を繰り返すことにより展開される．これらのうち，初期設計と前向き操作は明示的な知識によって行われる．一方，修正のための後向き操作のための明示的な知識は一般には存在しないため，その部分的な内容を前向き知識の内容などをもとにした最適化計算により提示する支援機能を提案している．また，評価については，個別項目の評価は明示的な知識によって行われるものの，それらを総合した判断は設計者に委ねるものとする．さらに，領域モデル内においては，以上の内容に加えて，粒度レベルの移行を随時行う必要があるが，これについても，時間限定性のもとである種の基準は提示するものの，最終的には設計者に委ねるものとしている．

以上のようにして分散構造のもとでゴールを介した協調処理を進めていけば有効な設計支援とすることができるということが本報の前提である．以下では，まず，それをコンピュータ上に展開するための方法について示す．

3 エージェント方式による分散化と情報の共有と交換

3.1 エージェント方式による分散化 2.1項に示した分散構造とそのもとでの並行処理を実現するために，本研究ではエージェント方式を導入する．エージェントの意味は分野やアプリケーションによって多様であるが，ここでは分散環境下で異なる計算資源に割り当てられた自律的な処理単位を意味する[3]．具体的なエージェントとしては，図 1にも示すように，対象モデルの分割に対応させて，以下の 2 つのタイプのものを導入する．

- **大域エージェント** (global agent) ··· 大域モデルに基づき，設計における起点としての全体の概略を定めた後，領域モデルによる詳細な設計展開に対してスーパーバイザーとしての管理を行う．
- **領域エージェント** (domain agents) ··· 各領域モデルに基づいて，それぞれの部分の設計を担う．

このようなエージェント群の構成のもと，大域エージェントの役割は前述の大域モデルの役割に基づいた補助的なものである．一方，対象モデルの分割に基づいて構成される一連の領域エージェントは，それぞれに個別の設計処理を分散的に展開しつつ，相互の依存関係を通じて協調処理を行う．

3.2 共有情報と隠蔽情報 上記の各エージェントはそれぞれに固有のモデルに基づいた設計を展開する．各モデルの内容は 2.2項にも示したように一連の変数と関連する知識から構成されている．各領域モデルを相互に連結するものは媒介変数であるため，領域エージェントの間では媒介変数についての情報が共有される必要がある．また，大域モデルに含まれる (大域) 決定変数や (大域) 性能変数も最終的な決定は領域モデルにおいて行われることから，それらについての情報は大域エージェントと領域エージェントの間で共有される必要がある．つまり，エージェントの間で共有されるべき情報は特定の変数についてのものであって，それ以外の内容，例えば各種の知識はそれぞれのエージェントに隠蔽されるべきものとなる．なお，以下では共有情報に関わる変数を**共有変数** (shared variable) と呼ぶことにする．

3.3 情報交換における方向性 エージェントの間で交換する情報は具体的には共有変数についてのゴール情報である．あるゴールについての媒介入力変数と媒介出力変数との関係が示すように，その共有と交換においては何らかの方向性が存在する．そこで，各エージェントの側からみた場合の共有変数を，向き操作の方向性に基づいて，以下のように区別する．

- **輸出変数** (exported variable) ··· 情報を供給する側にあるエージェントにおける共有変数．具体的には，領域エージェントにおける媒介出力変数，大域エージェントにおける大域決定変数，大域性能変数．
- **輸入変数** (imported variable) ··· 情報を参照する側にあるエージェントにおける共有変数．具体的には， 領域エージェントにおける媒介入力変数，大域決定変数と大域性能変数に対応する各変数．

なお，例えば，媒介変数に関して出力変数側での選好範囲を策定するために入力変数側での決定範囲を参照するなどして，逆方向に情報の供給と参照が行われる場合もある．

3.4 情報交換の非同期処理化 具体的な情報共有に向けて情報を交換するためにはエージェントの間で何らかの通信を行う必要がある．エージェント間での並行処理を効率的に実現するためには，相互の通信を非同期化することが不可欠である．一方，それぞれのエージェントには固有の設計処理があって，単純にそれに専念していたのでは非同期的に送信されてく

る共有情報を処理することはできない．そこで，エージェントを実装する計算資源におけるマルチタスク機能のもと，あるエージェントにおいては以下の二つのタスクを並行的に展開するものとする．

中核タスク (core task) ··· エージェントに固有の設計処理を行う．その過程で自らが更新したゴール情報を必要に応じて他のエージェントに送信する．

仲介タスク (mediator task) ··· 他のエージェントから送信されてくる情報をもとに該当する共有変数のゴール情報を更新する．

なお，ゴール情報の参照・更新にあたっては両タスクによる並行処理において各種変数の変更をそれぞれに相互排除的に行うためのロック機能(4)を導入する必要もある．

以上の非同期化によって，各エージェントにおける処理は自律性の高いものとなる．

4　エージェント内での知識表現と設計処理

4.1　オブジェクト指向による知識表現　前節のようにしてカプセル化される各エージェントにおける設計処理を実現するにあたっては，オブジェクト指向による知識表現と処理方法(5)を導入する．本研究では2.2項でも述べたように対象モデルの内容を集中定数系として記述できるものと想定していることから，それにおける一連の変数と相互の関係式 (知識) を表現する必要がある．また，付随して，粒度レベルを管理できたり，各変数については 2.3項に示したゴール情報をも同時に表現する必要がある．

以上の内容やその他の内容を表現するにあたっては，図 2に示す一連のクラスを用意する．以下にそれらのなかでも主要なものについて説明する．

granularity-level ··· 領域エージェントにおいて粒度レベルを管理するためのオブジェクトのクラス．

variable ··· 各種の変数を表現するためのオブジェクトについてのクラス． 2.2項や 3.3項の内容に応じて各種のサブクラスを用意する．なお， 2.3項に示したゴール情報はそのスロットに保持される．

parameter ··· 設計対象に関する属性値であるものの，設計の過程で変更されないものを表現するためのオブジェクトについてのクラス．設計条件や各種の定数がこれに属する．

procedure ··· 設計知識のうち明示的なものを記述するためのオブジェクトについてのクラス．初期設計知識・前向き知識・個別評価知識がこれに属し，知識に相当する関係式をスロット *form*に保持する．また，各関係式の引数となる変数やそれが値を与える変数との間に関係[†1]を定義する．さらに，前向き知識についてはそれが属する粒度レベルに関係付ける．

backward-chaining ··· 後向き操作のための最適化計算による支援機能を実現するための各種のオブジェクトについてのクラス．

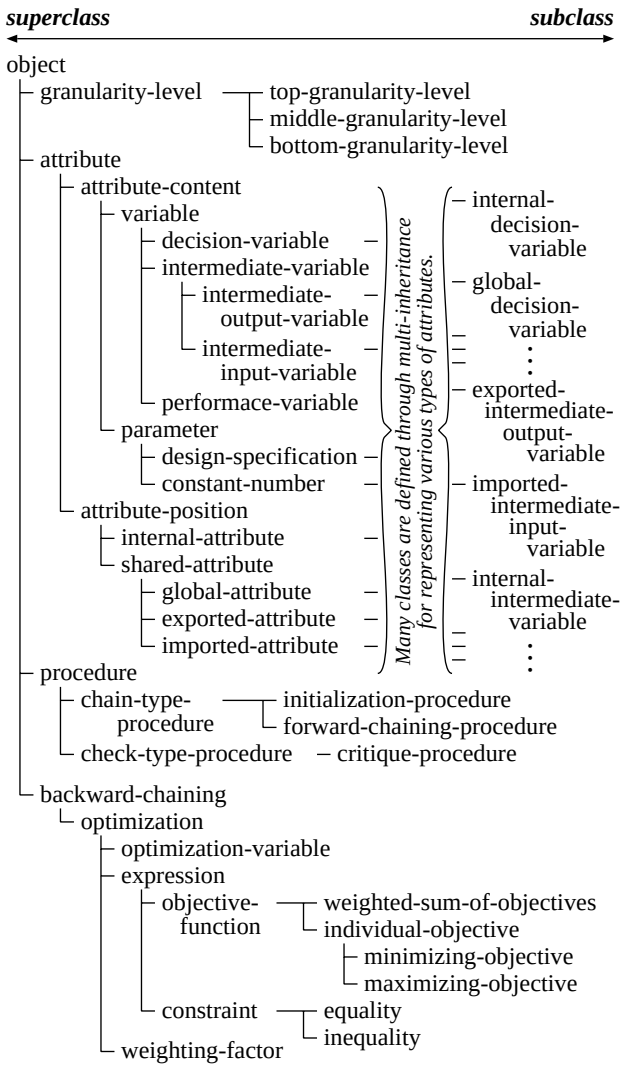


Fig. 2　Class hierarchy for knowledge representation

なお，個別エージェント内で前向き知識を細分化して記述する必要上，個別知識を連結する媒介変数が必要となるが，これについては**内部媒介変数** (internal intermediate variable) と呼ぶことにする．

以上の各クラスのもとで定義される様々なオブジェクトは相互に連結されて，エージェントにおける対象モデルの内容を表す．図 3はある領域エージェントに

[†1] 本報では，通常のオブジェクト指向に対する拡張としてオブジェクト間の方向性を伴った関係を表現するための仕組みを導入しており，‘association’ と呼んでいる．

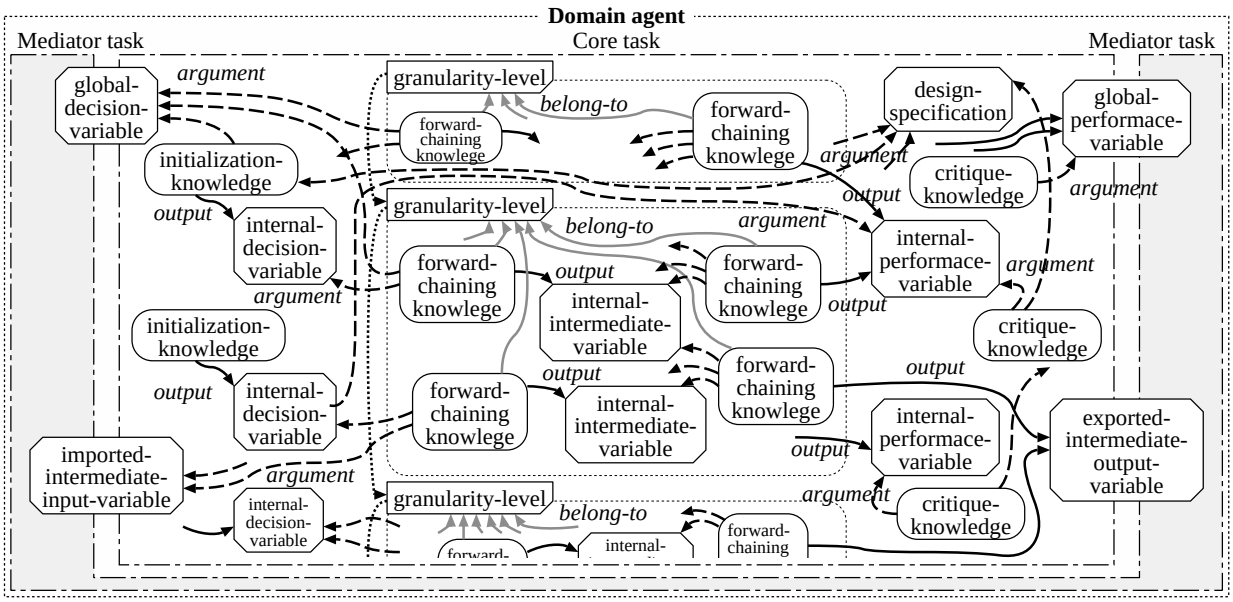


Fig. 3　Objects and associations in a domain agent

おける連結の状況を模式的に示したものである．図中の関係 *argument*は関係式の引数を， *output*は関係式の算出を， *belong-to*は知識が属する粒度レベルを，それぞれ指し示すものである．なお，大域エージェントにおける状況は図 3から粒度レベルによる多重性を排除したものにほぼ対応する．

4.2　領域エージェントでの粒度レベルの切り換え

領域エージェントで何らかの処理を行うには，対象となる粒度レベルを定めて，それに関与する変数と知識を抽出する必要がある．これについては，まず，図 3の状況のもとで，各オブジェクトのスロット *active?*にそれぞれの参照可能性を保持するようにする．その上で，対象となる粒度レベルのオブジェクトから *belong-to*という関係を介して関連する前向き知識を列挙し，さらに *argument*と *output*という関係をたどって関連する変数を列挙して，列挙された一連のオブジェクトにおける *active?*の値を false から true に，随時，変更することにより，参照可能性を管理する．

4.3　オブジェクト指向のもとでの設計処理

大域エージェントと領域エージェントにおける設計処理は，初期設計・前向き操作・評価・後向き操作・移行，の 5 つから構成される．これらのうち，初期設計・前向き操作・評価における個別評価知識の操作は明示的に記述された知識 (関係式) に基づくものである．それらについては，一連の変数と知識が *argument*と *output*という関係を介して連鎖してネットワークを構成していることに基づいて，関係式における依存関係に従って個別のメソッドを起動するためのメッセージを再帰的に送信することにより，処理を行う[5]．例えば，決定変数や媒介出力変数の値を算出する処理は， *output*からたどれる知識のオブジェクトにメッセージを送信し，それを受けたオブジェクトが *argument*からたどれる変数にその値を要求するメッセージを送信し，返ってくる結果に基づいてスロット *form*の内容に対応する計算を行って，その結果を返すことを繰り返すことにより行われる．

4.4　後向き操作のための支援処理

上述の 5 つの処理のうち，個別の評価結果を総合した上で後向き操作を行う一連の局面は逆問題としての設計における中核であり，本研究では最適化計算による支援機能を提案している[2]．一連の処理は図 2に示した backward-chaining 以下のクラスに属する様々なオブジェクトを用いて行うが，それらは知識ベースをロードした際に事前に生成しておく．その上で，各スロットの情報を各時点での状況に応じて書き換えつつ，以下の手順に従って支援のための処理を行う．

(i) その時点での前向き操作による結果や個別評価の結果，さらに他のエージェントから通知されているゴール情報に従って，各オブジェクトの内容を更新する．その際には，最適化計算における各設計変数の上下限値なども設定する．

(ii) 各制約条件や目的関数についての具体的な式を前向き知識の内容とそれらから得られている結果とから構成し，さらに，一連の内容に対して線形化近似や単調性解析などによる簡略化処理を施す．

(iii) 設計者の判断に基づいて，考慮すべき制約条件や

目的関数を取捨選択したり，最適化計算における重み付けの係数を調整する．

(iv) 以上の内容から 2 次計画モデルにおける一連の係数を定めて，最適化計算を実行する．

(v) 得られた結果を設計者に提示し，具体的な設計修正へと進むか，あるいは支援のために設定したモデルに不具合があった場合には，(iii) に戻って，別の設定での修正案を探求する．

4.5 設計者との関係 本研究での枠組みはあくまでも支援システムであって，総合評価，後向き操作に向けての支援機能における方向設定，具体的な設計修正の実行，さらに，粒度レベルの移行やゴールの縮小などの判断は設計者に委ねられることになる．それらを円滑に行えるように，後出のプロトタイプシステムにおいては，各種のインターフェース機能を用意することにする．

5 プロトタイプシステムの実装

分散協調型設計支援システムのプロトタイプは，各エージェントを個別のワークステーションに実装した上で，それらの間の通信により情報の交換が行えるようにすることによって，構成する．

5.1 個別エージェントの実装 個別エージェントの実装に当っては Allegro Common Lisp (ACL)[6]をシステム構築のための基本言語として用いる．個別の設計処理を実現する上で，Lisp の記号処理機能と CLOS (Common Lisp Object System)[7]によるオブジェクト指向プログラミングは有用であり，3.4 項で述べた非同期化を実現するには ACL のもつマルチタスク機能は必須である．また，後向き処理のための双対法の計算は Fortran によるサブプログラムを呼出して行い，ユーザインターフェースは CLIM (Common Lisp Interface Manager) を用いて構築した．

5.2 情報交換のための通信方式 エージェント間の通信には，通信サーバを経由してそれらの間で TCP/IP ソケットによる通信を行なう方式[3]を用いる．各エージェントにおいては，C 言語による通信処理サブプログラムを用い，送信内容は中核タスクから処理する一方，受信内容は仲介タスクで処理することで，通信処理によるデッドロックを回避する．一方，通信サーバは C 言語で実装されており，各エージェントから送信されてくるメッセージをタグに示されたエージェントに単純に転送する機能のみを持っている．

6 航空機の基本設計への適用

本節では，以上の内容の妥当性や有効性を検証するために，航空機の基本設計[†2]への展開例を示す．

6.1 設計における分散協調構造 航空機の基本設計における要求は乗客数，航続距離，離着陸距離，巡航速度，巡航高度などとして与えられ，それに対する設計処理の内容は対象システムのモジュール性や機能評価における領域性に対応した分散構造を伴っている．プロトタイプシステムの展開においては，それを踏まえて，領域エージェントとして，胴体，主翼，水平尾翼，垂直尾翼，エンジン，揚力・抵抗，航続性能，離着陸性能，安定性，のそれぞれについての 9 つのものを構成する．また，様々な設計知識は文献[8]~[11]などから抽出したものを用いる．

6.2 設計知識の階層性 表 1は，設計知識のなかでも，主に実体に関与する領域モデルにおける決定変数の内容とそれらの粒度レベルについての区別を示したものである．これに対応して，性能変数に関わる様々な内容は異なる忠実度の知識により推定されることになる．例えば，主翼の質量 W_w [kg] を推定するための前向き知識は第 1 粒度レベルでは次式で算出するものとしている[12]．

$$W_w = 0.0067\, B_w{}^{0.75}\left(1+\frac{1.38}{B_w}\right)^{2.457} \times \left\{\frac{B_w{}^2\,(C_{r_w}+C_{t_w})}{t_{r_w}\,W_{to}^a}\right\}^{0.3} W_{to}^a \quad (1)$$

ここで，B_w は主翼翼幅 [m]，C_{r_w} は主翼翼根翼弦長さ [m]，C_{t_w} は主翼翼端翼弦長さ [m]，t_{r_w} は主翼翼根厚さ [m]，W_{to}^a は推定離陸質量 [kg] である．

これに対して，第 2 粒度レベルでは，主翼の内部構造を考慮することから，上式の内容を次式で算出するものとしている[13]．

$$W_w = k\,(W_{sk_w}+W_{sp_w}+W_{str_w}+W_{rib_w} +W_{edge_w}+W_{lef}+W_{trf}) \quad (2)$$

ここで，k は航空機の形態に依存した係数であり，エンジンの取付け位置が主翼であり降着装置の取付け位置も主翼である場合には 1.3 となる．W_{sk_w} は主翼外板質量 [kg]，W_{sp_w} は主翼桁質量 [kg]，W_{str_w} は主翼ストリンガ重量 [kg]，W_{rib_w} は主翼リブ質量 [kg]，W_{edge_w} は主翼エッジ質量 [kg]，W_{lef} は前縁フラップ質量 [kg]，W_{trf} は後縁フラップ質量 [kg] である．これらのうち，例えば，W_{sk_w} は，主翼翼端厚さを t_{t_w} [m]，主翼面積

†2 ここでは，実用的なものではなく，研究目的に照し合せて簡略化したものを取り上げている．

Table 1　Decision variables in different granularity levels

Domain	Decision variables in the first granularity level $\boldsymbol{x}_{g_1}$	Decision variables supplementary in the second granularity level $\widehat{\boldsymbol{x}_{g_2}}$
Fuselage	Length^G, Breadth^G	The number of stringers, The number of frames
Main wing	Span^G, Root chord, Tip chord, Root thickness, Tip thickness, Angle of dihedral, Angle of sweep^G, Angle of incidence, x-coordinate of mount^G, z-coordinate of mount	The number of stringers, Location of front spar, Spar interval, The number of ribs
Horizontal tail wing	Span, Root chord, Tip chord, Root thickness, Angle of sweep, Angle of incidence, x-coordinate of mount^G, z-coordinate of mount	Tip thickness, The number of stringers, Location of front spar, Spar interval, The number of ribs
Vertical tail wing	Span, Root chord, Tip chord, Root thickness, Angle of sweep, x-coordinate of mount^G, z-coordinate of mount	Tip thickness, The number of stringers, Position of front spare, Spar interval, The number of ribs
Engine	Length^G, Diameter^G, x-coordinate of mount^G	y-coordinate of mount, z-coordinate of mount

^{G} :　These are the global decision variables as well.

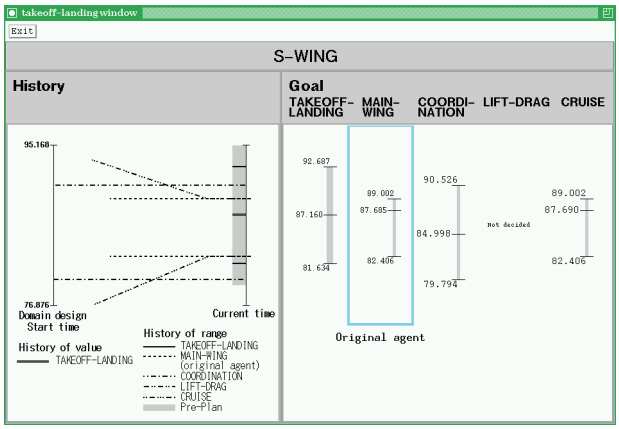
を S_w [m^2]，主翼外板厚さを t_{sk_w} [m] とするとき，$4.2\{(C_{r_w}+C_{t_w}+t_{r_w}+t_{t_w})B_w-0.2\,S_w\}\,t_{sk_w}$ により与えられる．

以上のように，式 (1) と式 (2) の間では操作性も忠実度も異なっている．なお，各エージェントにおける変数と個別の設計知識の数は，大域エージェントでは変数が 33 個，設計知識が 67 個である．領域エージェント群については，9 つの合計で変数が 265 個，設計知識が 368 個である．後者を粒度レベルごとに分類すれば，第 1 粒度レベルにおいて扱う変数が 141 個，設計知識が 142 個であり，第 2 粒度レベルにおいて加わる変数が 124 個，扱う設計知識が 226 個となる．

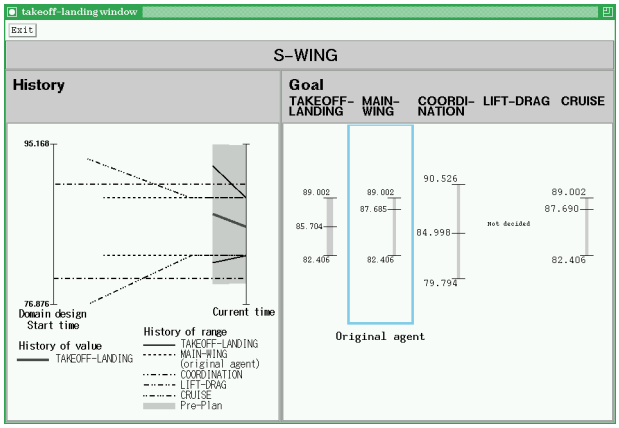
6.3　設計支援の事例　図 4からは図 6はプロトタイプシステムを用いて設計を進めていく過程で表示されるウインドウのいくつかを示したものである．

図 4は離着陸性能エージェントでのウインドウであり，領域間協調のための処理を主翼面積についてのゴール情報に着目して示している[†3]．(a) はある時点での状況を他の領域エージェントでのゴール情報に関連させて示したものである．(b) では，そのもとで，他のエージェントでのゴール情報を参照しつつ最適化計算に基づく支援機能を利用しつつ，主翼面積についての決定範囲 (主翼エージェントにおける選好範囲に対応する) を (a) の約 60% にまで絞り込んでいる．

図 5は大域エージェントでのウインドウであり，大域的協調に向けての処理を離陸推力についてのゴール情報に着目して示している[†4]．(a) は大域的協調を行う以前の状況であり，エンジンエージェントでの範囲が離着陸性能エージェントや航続性能エージェントでの



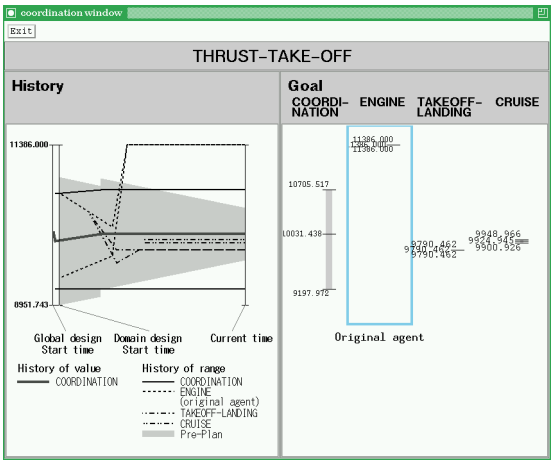
(a)　Before optimization-based refinement



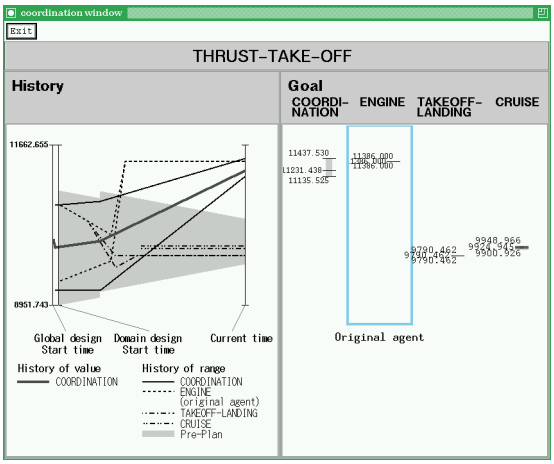
(b)　After optimization-based refinement

Fig. 4　Transition of goal information on main wing area at the takeoff and landing agent as an example of inter coordination

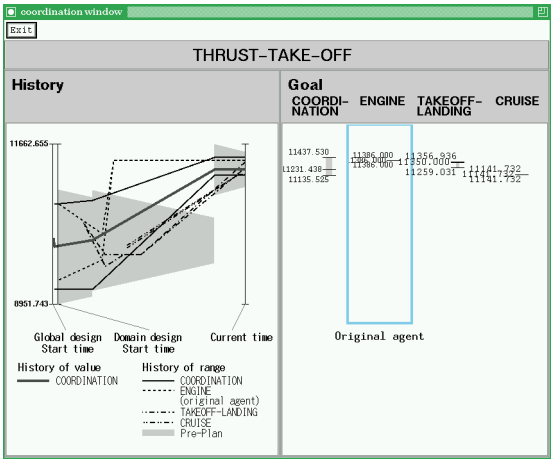
†3　第 2 報[(2)]での図 2 の①局面に対応する．
†4　第 2 報[(2)]での図 2 の③局面に対応する．



(a)　Before optimization-based refinement



(b)　After optimization-based refinement



(c)　After propagation of refined goals to domain agents

Fig. 5　Transition of goal information on takeoff power at the global agent as an example of global coordination

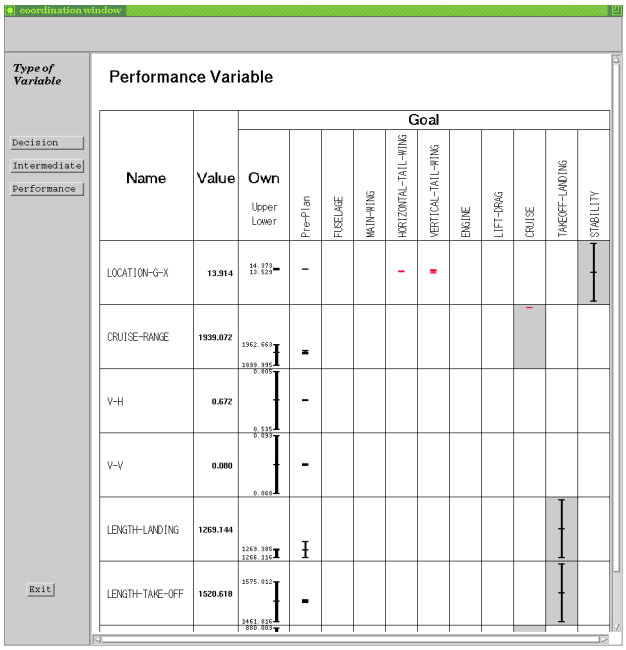


Fig. 6　Design result summarized at the global agent

範囲を上回っていて，共通する範囲のないことを示している．これに対して，(b) では，大域モデルに従いつつ最適化計算に基づく支援機能を利用してその選好範囲をエンジンエージェントで算出されている性能値を含む比較的狭いものに定めており，結果的にエンジンエージェントでの範囲に近いものとなっている．(c) では，それを受けて各領域エージェントがそれぞれのゴール範囲を修正し，結果として相互の矛盾が解消されたことを示している．

図 6は大域エージェントでのウインドウであり，以上の状況を経た設計の最終局面においていくつかの性能変数について各領域エージェントのゴール情報を比較したものであり，相互の矛盾も解消され，満足な設計結果が得られつつあることが確認できる．

以上のような航空機の基本設計への適用により，実装した対象知識の内容は現時点ではやや不足しているものの，本研究で提案した分散協調型設計支援システムに向けてのプロセスモデルと操作モデルの基本的な妥当性を示すことができたと考えている．

7　結　言

本報では，第 1 報[1]と第 2 報[2]で提案した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に対する実装方法と航空機の基本設計問題への適用を通じて，それらの妥当性と有効性を示した．一連の内容は集中定数系による形式化に依存するものではあるが規模性に基づく設計問題の困難性に対峙するた

めの方向を具体的に示すものである．今後は，協調処理をより積極的に支援していくための方策について検討を進めるとともに，対象モデルの内容が形式化からは外れる場合の状況についても何らかのモデルを構成していく必要があると考えている．

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) 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 1 報，プロセス構造についてのモデル), 日本機械学会論文集 C 編, Vol. 68, No. 666, (2002), pp. 657-665.
(2) 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), pp. 666-674.
(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) 例えば， Brawer, S., (大森 訳), 並列プログラミングの基礎 — マルチプロセッサを指向した応用プログラム構成法 —, (1990), pp. 70-83, 丸善.
(5) 赤木・藤田, オブジェクト指向に基づく設計エキスパートシステムの研究，日本機械学会論文集 C 編，Vol. 54, No. 500, (1988), pp. 1017-1025.
(6) *Allegro CL User Guide, version 4.3*, (1996), Frantz Inc.
(7) Steele, G. L. Jr., *Common Lisp — The Language* (*2nd edition*), (1990), Digital Press.
(8) Torenbeek, E., *Synthesis of Subsonic Airplane Design*, (1976), Delft University Press.
(9) Raymer, D. P., *Aircraft Design: A Conceptual Approach*, (1989), AIAA Education Series.
(10) 山名・中口, 飛行機設計論, (1985), 養賢堂.
(11) 鳥養・久世, 航空機の構造設計 — その理論とメカニズム —, (1992), 日本航空技術協会.
(12) 文献 (8) の p. 280.
(13) 文献 (8) の pp. 452-455.