

## 静的線形有限要素解析のオブジェクト指向分析と設計

AN OBJECT ORIENTED ANALYSIS AND DESIGN OF  
STATIC LINEAR FINITE ELEMENT ANALYSIS

石田 栄介\* 新美 勝之\*\* 福和 伸夫\*\*\* 中井正一\*\*\*\*

Eisuke ISHIDA Katsuyuki NIIMI Nobuo FUKUWA Shoichi NAKAI

The object oriented analysis of finite element method is presented in this paper. First, the necessity of the object oriented approach to structural analysis is pointed out from a lot of view points. Then, the object oriented modeling technique (OMT), which consists of object model, dynamic model and functional model, is applied to the analysis of finite element method. By using OMT, finite element methods are analyzed from three points of view, which are static structure, control and data flow. Based on these analyses, the framework of finite element system for linear static analysis is designed. The pilot program is demonstrated in order to present the validity of the present approach.

**Keywords :** object oriented analysis, object modeling technique, finite element method, C++  
オブジェクト指向分析, オブジェクトモデル化技術, 有限要素法, C++

## 1. はじめに

ソフトウェア工学の発達に伴い、オブジェクト指向の考え方が今後のソフトウェア開発の中心になってきた。オブジェクト指向は、単なるシステム記述言語にとどまらず、問題の表現や分析にも利用されつつある。このオブジェクト指向分析を、既成の概念の再整理や統合化に利用する試みが多方面で始まっている。本論では、建築構造工学へのオブジェクト指向導入の手始めとして、構造解析における代表的解析手法である有限要素法を対象にオブジェクト指向分析を行い、有限要素構造解析システムをオブジェクトの集合体として設計し、パイロットプログラムの試作を通してその有効性を検証する。

オブジェクト指向の考え方は1960年代末のSimula<sup>1)</sup>の発表に始まり、Small Talk-80<sup>2)</sup>の登場とともに1980年代末から急速に利用が展開されてきた。当初は、構造化プログラミングを補う技術として、主にプログラム記述言語として利用されていたが、近年、オブジェクト指向の考え方をより早期の段階から適用し、問題を解決する手段として利用しようという立場で見直す動きが現われ

ている<sup>3)4)5)6)</sup>。ここでは、問題を解決する手段として、問題分析 (Object-Oriented Analysis : OOA)、設計 (Object-Oriented Design : OOD)、実装 (Object-Oriented Programming : OOP) の3つの段階からなる開発サイクルを踏むことが提案されている。問題解決の立場からは、分析、設計、実装、の順に重要度が高く、実装は設計モデルの翻訳でしかないとの見方がされている。従って、問題をいかに表現するかがポイントとなり、現実の問題をモデル化し表現する方法論が重要となる。Rumbaugh<sup>6)</sup>は、モデル表現の手段としてオブジェクトモデル化技術 (Object Modeling Technique:OMT) を提唱している。OMTは、オブジェクト構成を示すオブジェクトモデル、状態遷移を示すダイナミックモデル、データフローを示すファンクショナルモデル、の3つのモデルに対してグラフィカルな表記法を提供している。これらのモデルは直交する3つの側面から問題を表現し、互いに補完し合う関係となっている。

一方、有限要素法構造解析システムの開発に関しては、大量データの処理や解析の高速化を図るため、種々の研

\* 名古屋大学工学部建築学科大学院生・工修

\*\* 清水建設 大崎研究室・研究員・工修

\*\*\* 名古屋大学工学部建築学科助教授・工博

\*\*\*\* 清水建設 大崎研究室・主任研究員・工博

Dep. of Architecture, Faculty of Eng., Nagoya Univ., Undergraduate student

Ohsaki Research Institute, Shimizu Corporation, Res. Eng. MS.Eng.

Dep. of Architecture, Faculty of Eng., Nagoya Univ., Assoc. Prof., Dr. Eng.

Ohsaki Research Institute, Shimizu Corporation, Senior Res. Eng. Dr.Eng.

究が行われている。ベクトルプロセッサを利用したスーパーコンピューティングに関する研究<sup>7)</sup>も行われてきたが、1つのプロセッサでは処理速度に限界があるという立場から、最近では多数のプロセッサによるパラレルコンピューティングに関する研究<sup>8) 9)</sup>も盛んになっている。中には、近年のEthernetなどによるデータ転送速度やネットワーク利用環境の飛躍的向上を利用し、ネットワークを形成する複数の計算機を同時に使って並列に処理を行う試みも始められている<sup>10)</sup>。また、処理能力の向上に伴い、大規模な構造解析が頻繁に行われるようになってきていることから、これらの支援環境として、所謂プレポストのグラフィックユーザーインターフェース(GUI)の研究も盛んになっている<sup>11)</sup>。この背景には比較的安価なEWSの普及とEWSソフトのGUI環境の発達があると考えられる。このような計算機環境を活かした研究成果を積極的に取り入れていくためには、特定のハード/ソフトに依存することなく柔軟にシステムを融合していくことができるように有限要素構造解析システムの枠組みの見直しを始める必要がある。それに対し、従来より用いられてきた汎用解析コード<sup>12)</sup>は、システムがあまりに肥大化しており、利用者による機能拡張に限界があること、膨大なマニュアルを参照する必要があること、などから今後の計算機環境の進展を有効活用していくには、限界があると考えられる。

また、解析理論における現状を考えると、約30年間に渡る蓄積の結果、膨大な理論体系が築かれている<sup>13)</sup>が、今もなお、新たな解析理論、解析スキームは絶え間なく研究開発されている。特に、大型宇宙構造物や原子力発電施設の振動解析などのように、特殊な構造物の振動性状を把握するためには、種々の解析手法の特徴を踏まえた融合化を通して、新たな解析手法を開発していくことが必要である。

そこで、本論文では、近年のソフト・ハード両面に渡る計算機環境の有効活用、最新の研究成果を反映しやすい柔軟な構造解析システムの実現を目指して、有限要素法をオブジェクト指向分析し、オブジェクトの集合体として有限要素解析システムを設計する試みを行う。オブジェクト指向は、従来のアルゴリズム主体の考え方から大きく異なり、概念を素直に表現できる多くの機能を有している。従来は、データの集合である実体とアルゴリズムの集合である解析が遊離していたのに対し、オブジェクト指向では、共通の階層構造を有するオブジェクトを用いて、対象物のデータ表現から物理・数理モデルの作成、解析、評価までを統一的に表現できる。また、プログラミングを意識せずに有限要素法を分析することができ、有限要素法概念の素直な記述がシステム構築へつながっていく。

オブジェクト指向の構造解析への適用としては、著者らの試み<sup>14) 15) 16)</sup>の他には、CとPascalを用いたFordeらによる有限要素解析システムの設計<sup>17)</sup>、Smalltalkを用いた反復法によるトラスの構造解析<sup>18)</sup>や線形振動有限要素解析<sup>19) 20) 21)</sup>、C++を用いた有限要素法クラスライブラリーの基本設計<sup>22)</sup>、有限要素データのデータ抽象化に関する研究<sup>23)</sup>、有限要素解析の有限要素データのデータ変換ならびにグラフィックスへの適用に関する研究<sup>24)</sup>、EWS環境を有限要素モデリングに積極的に利用する立場でオブジェクト指向を導入した研究<sup>25)</sup>、などここ数年の間に急速に研究が展開されている。また、平成5年度の応用力学連合講演会、機械学会主催の計算力学講演会などでオブジェクト指向の特別セッションが設けられる<sup>26) 27) 28)</sup>など、計算力学分野においてもオブジェクト指向の重要性が認識されつつある。

このような背景のもと、本論では、Rumbaughら<sup>9)</sup>のOMT手法を用いて有限要素法のオブジェクト指向分析を行う。この分析結果に基づいて、オブジェクトモデルを設計し、著者らの想定する有限要素解析システムのクラス階層を明示する。また、設計された有限要素解析システムに対し部分構造を追加することにより、部分法への拡張を行う。この拡張を通じて、本開発アプローチの拡張性に関する考察を行うとともに、階層構造の素直な表現が可能な本解析システムの柔軟性について述べる。本有限要素システムのねらいは、種々の解析手法を高度に組み合わせて行う解析を実現し、システムの成長を容易にする枠組みを提供することにある。将来は、EWSのGUI環境を利用すると共に、ネットワークで接続されたEWSによる並列処理を念頭においている。そこで、構造物を部分構造に分割することにより、各部分構造で最も適切な解析手法を選択し、かつ複数のEWSで分散処理できるようにすると共に、クラスライブラリーの設計を通してシステムの拡張性や再利用性を高めることを進める。このためには、実現象である構造物、有限要素法概念、ならびに解析システムの各階層構造を1対1対応させるとともに、十分吟味されたオブジェクトモデルを作成することが肝要となる。本有限要素システムが対象とする問題は、静的ならびに動的問題、線形・材料非線形・幾何学的非線形問題であり、用いる動的解析手法は直接時間積分法・モード重畳法・周波数応答解析法とし、複数の解析手法を融合化する手段としては部分構造法を前提とする。既往の汎用解析コードは、システムがプロセッサ的な体系であるので、このような柔軟性を保持することは難しく、解析手法の自由な組み合わせはオブジェクト指向の導入によって実現が容易となる。なお、本論においてオブジェクトモデルの分析に重点をおいているのは、オブジェクト指向では、システムの分析ならびに設

計が完全に行われることが重要であり、実装は単純なプログラミング作業であるとの立場に立っているからである。また、本報告では、静的線形解析を中心に有限要素解析システムの構造を分析/設計に議論を行うこととし、動的解析<sup>29)</sup>ならびに非線形解析<sup>30)</sup>に関しては次報以降に報告することとする。これは、静的解析を通して、有限要素法の基本的なデータ構造を分析できると考えたためである。

## 2. オブジェクト指向による問題解決

従来の手続き型言語による設計では、「手続き」を決定することに目標が置かれていた。しかし、手続きは解析のねらいが変わるごとに大きく変化し、その度に再構築を強いられる。それに対し、オブジェクト指向では、「オブジェクト」の決定に目標が置かれている。オブジェクトは、「属性」と「振る舞い」を持ち、実現象を表現するのに適している。図-1の例において、はり(beam)や節点(node)はオブジェクトであり、オブジェクトの集合によって対象とする物理モデルを素直に写像できることを示している。それに対し、手続き表現である「流れ図」は、解析方法が確定するまで決定することができない。

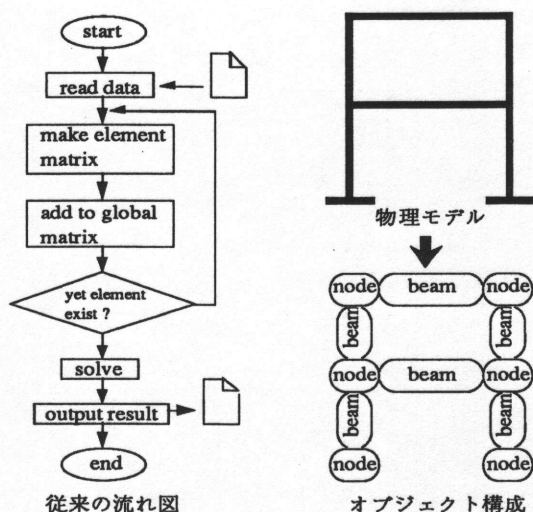


図-1 従来法との比較

オブジェクト指向の概念をサポートするプログラミング言語では言語上でオブジェクト表現ができるため、これまでは、直接オブジェクト指向プログラミング言語を介して議論を進めることが多かった<sup>29)</sup>。しかし最近では、分析が言語に依存しないようにすること、より大規模なシステムに対しても直感的な表現ができるようにすること、などの理由から図化表現が推奨されるようになってきた<sup>30)</sup>。その代表として、OMT<sup>®</sup>では、オブジェクトの構造およびオブジェクト同士の静的な

関係を記述するオブジェクトモデル (Object Model)、システム内で起こるイベントやそれに伴う状態の変化を記述する動的モデル (Dynamic Model)、システム内のデータ変換関係を示す機能モデル (Functional Model)、という3つの異なるモデルを用いて問題を表現する。これらのモデルは、それぞれオブジェクト図、状態図、機能図によって表記される。文献6において表記法の詳細が記されているが、図-2に OMT 方法論の3つのモデルタイプとそれぞれのモデル図の簡単な表記方法を示す。

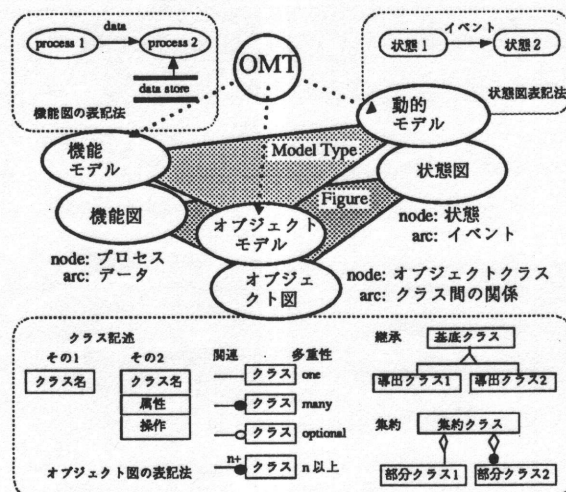


図-2 OMTにおける3つのモデルタイプ

OMTでは、分析、設計、実装の3つの段階を踏むことを提唱しており、個々の概念の本質、システム内での位置付け、計算機上での表現を厳格に区別している。図-3にオブジェクト指向システム開発の流れを示す。

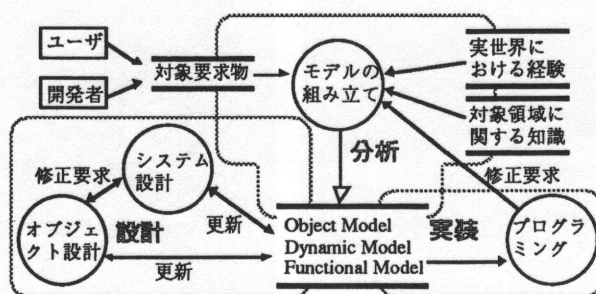


図-3 問題分析・設計・実装へ至るプロセス

設計は、分析されたモデルの各システムへの適合行為であり、実装は、設計されたモデルの忠実な翻訳でしかない。即ち、永続性、再利用性の観点から言えば、分析>設計>実装の順で重要度が高く、それぞれ前段階からのモデルの変化が少ない、もしくは規則的であるほど、永続性、再利用性の高いシステムが構築され

ることになる。また、同じ分析概念に従い、規則性を持って分析→設計→実装への段階を踏んだ複数のシステムは高い融合性を有する。

### 3. 有限要素法のオブジェクト指向分析

有限要素法による静的線形構造解析問題を対象に、構造物、要素、節点、荷重などの有限要素法に関するオブジェクト指向分析を行う。まず、構造物に関する分析を図4に示す。構造解析問題において、構造物は受けた荷重に対して変形性状を示すオブジェクトと見なすことができる。構造物は、「釣合状態を得る」という振る舞いを持ち、荷重が加わると構造物全体が変形することにより釣合状態を得る。この構成関係、状態推移、データ流れをそれぞれ、オブジェクト図(a)、状態図(b)、機能図(c)を用いて表記した。

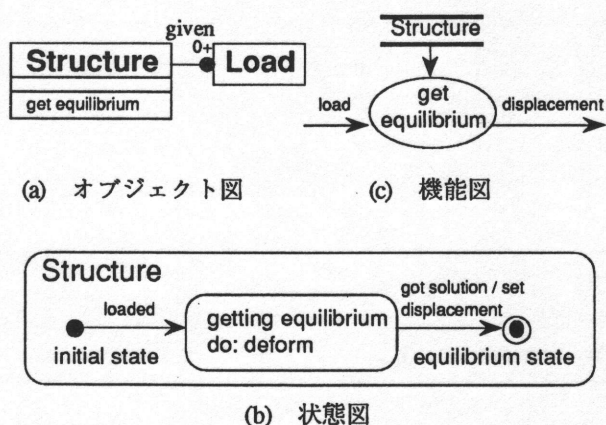


図4 構造物に関する分析

次に構造物の内部構成に着目すると、有限要素法では構造物は有限な自由度を持つ要素の集合体として定義され、構造物内における各要素相互の位置関係は節点を介して決定される。これらの空間構成関係を図5に示す。

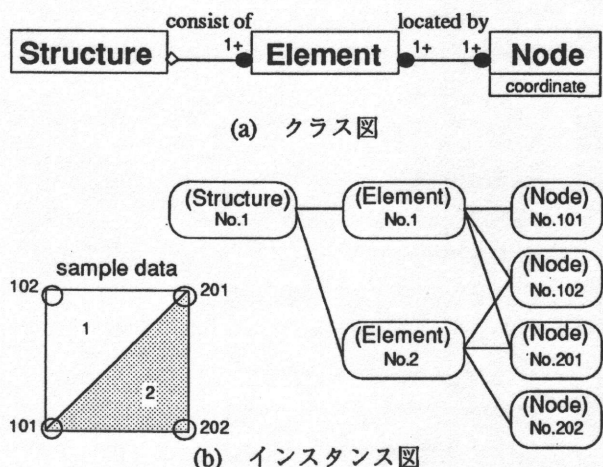


図5 構造物内の空間構成

ここから、有限要素法において構造物が荷重に対して釣合状態を得るために必要な諸概念に関する分析および設計を行う。有限要素法では、要素ごとに、内挿関数、形状、材料定数を用いて、構成節点に関する釣合方程式が導かれ、共有節点における連続条件から全体での釣合方程式が組み立てられる。この釣合式を用いることにより、図4の釣合状態を得ることが可能となる。静的線形構造解析問題では、式(1)によって釣合方程式が表される。

$$Ku = f \quad \text{.....(1)}$$

ここに、 $K$ : 剛性、 $u$ : 変位、 $f$ : 外力、である。従って、静的線形構造解析問題での釣合方程式に関するオブジェクト図および機能図は図6のようにになる。

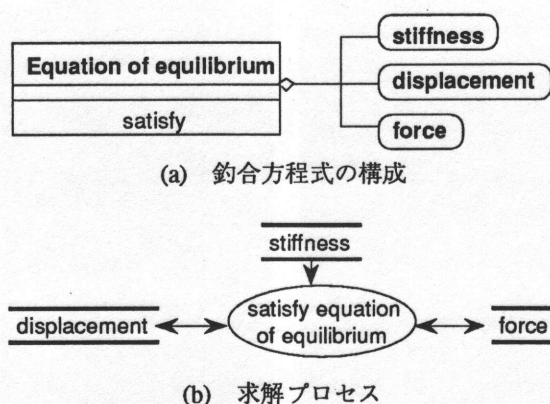


図6 静的線形構造解析問題における釣合方程式

各要素は、内挿関数、材料定数、構成節点から剛性を作成する。また、要素が物体力を受ける場合は、内挿関数などを用いて等価な節点荷重を作成する。各要素で作成された釣合式は、節点における適合条件により全体の釣合式へと組み込まれる。その際、各節点の変位方向成分に対応する自由度(Degree of freedom: Dof)の概念により、正確な足し込みが行われる。要素の釣合式と構造物の釣合式の間を関係を図7に示す。また、要素から構造物への釣合式の集積プロセスを図8に示す。

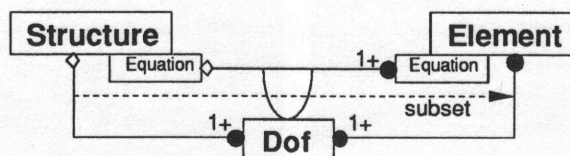


図7 構造物-要素間の釣合式の関係

このように分析段階では、有限要素解析問題の記述から始め、登場する各オブジェクトによる有限要素解析問題の忠実な表現を行なう。この時、「どのようにすべきか」ではなく、「何をすべきか」を表現することが肝要

となる。問題記述が一度に完全なものになることは殆んどないので、有限要素解析問題に関する専門家と分析者の間で綿密な意見交換を繰り返しながら分析を進めることが、完全な問題記述に近付くための大切な要因となる。そういった観点から、良い分析モデルとは、プログラマに分かり易いモデルではなく、有限要素解析問題に関する専門家が理解でき評価できるモデルを指す。

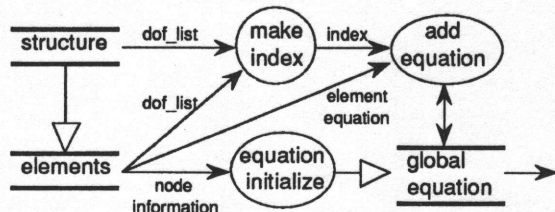


図-8 釣合方程式集積プロセス

#### 4. 有限要素解析システムのオブジェクト指向設計

設計段階では、分析モデルをもとに将来の拡張性を考慮しながら、より具体的な設計を行う。ここでは、静的線形問題を対象とする。図-9に解析システムのオブジェクト構成を示す。要素と全体構造は、共に自由度に関連する釣合方程式を持つことから、それらの共通点に対し抽象クラスFEOObjectを設計した。FEOObjectは、自由度と釣合方程式を持つ。要素は、内挿関数および材料定数、形状定数から釣合式を作成する方法を内部で定式化しており、メッセージ "give me Equation" に対して適切な処理を行い釣合式を渡す。構造物は、すべての構成要素に対し、メッセージ "give me Equation" を送り、要素から得た釣合式を適切に足し合わせていくことにより、自身の釣合式を組み立てる。足し合わせの際には、要素の持つ自由度リストと自分の持つ自由度リストを照らし合わせるにより、インデックスを作成する。自由度は、節点により管理され、構造物や要素、荷重から参照される。

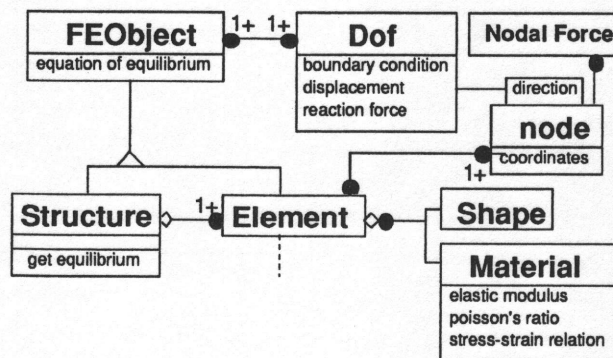


図-9 有限要素法システムのオブジェクト図

有限要素クラスElementは、すべての要素に共通する

要素の本質のみを定義した抽象クラスである。Element以下のクラス階層を図-10に示す。実際にオブジェクトを生成するはり、トラス、シェル、ソリッドなどは、すべてこのElementを継承して定義する。

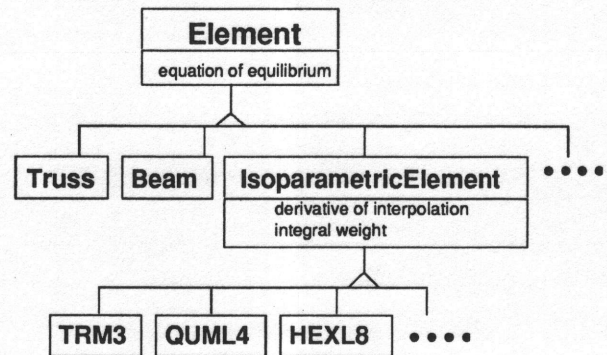


図-10 有限要素クラス階層

要素剛性は、歪み-変位関係、構成関係、内挿関数をもとに体積積分を行うことにより求められるが、この計算には、数値積分が用いられるのが一般的である。その中でもアイソパラメトリック要素は、内挿関数を仮想関数としておくことにより、その派生クラスにおいて各種アイソパラメトリック要素を定義することができる<sup>13)</sup>。それに対し、はりやトラスなど、要素剛性行列を陽に求められる要素は、直接Elementを継承して、それぞれ固有の手続きで要素剛性を求める。アイソパラメトリック要素の剛性作成プロセスを図-11に示す。

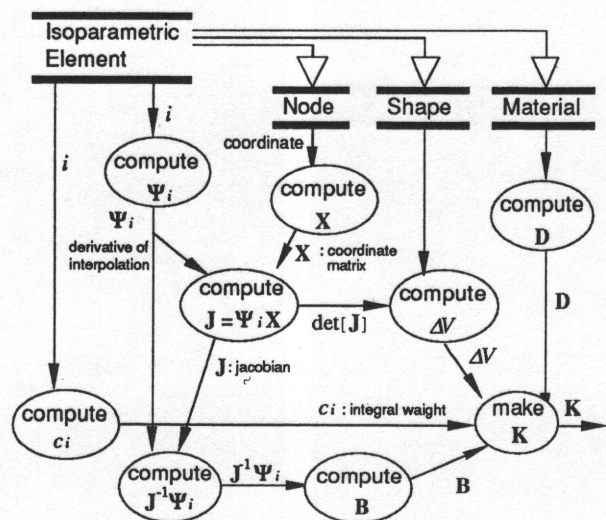


図-11 IsoparametricElementの剛性行列作成プロセス

このように設計段階では、有限要素解析システムの全体に渡る高レベルな決定を行ない、問題を解決していく上での戦略を決定する。また、分析モデルに基づいて全体システムからサブシステムへ問題を分化し、組織化を

行なっていく。有限要素解析システムの場合、釣合方程式を解くのは構造物であるが、構造物の釣合方程式は、分化された要素によって導かれた釣合方程式を集積することにより構築される。また、要素からクラスを導出することにより各種有限要素を定義していくという拡張性に関する戦略、FEObject を構造物と要素から抽象化し、概念共有を行なう等のシステム構成に関する戦略を明示している。これらの設計モデルの明示は、複数の開発者によって開発を行なう場合においても、開発戦略の一貫性を保ち、優れた融合性をもつ有限要素解析システムを構築していく上で有効である。

## 5. 部分法への拡張

これまでの設計モデルに部分構造を組み込むことにより、部分法への拡張を行い、本研究におけるアプローチの有効性を検証する。部分法は、全体構造を幾つかの部分構造に分け、部分構造ごとに性状を把握するとともに、内部節点を縮約することにより、結合節点における釣合式を用いてその特性を代表させる手法である<sup>31)</sup>。内部に構造を持つ点においては、全体構造と概念を共有しており、部分であること、単独では釣合式を解くことができないこと、という点においては、要素と概念を共有している。そこで、図-12に示すような設計を行い部分構造を組み込んだ。

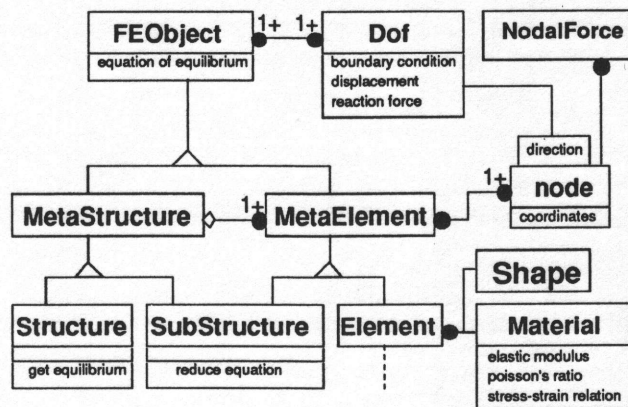


図-12 部分構造を組み込んだ解析システム構成

全体構造と部分構造の共通概念について抽象クラス MetaStructure (集合構造)、要素と部分構造の共通概念について抽象クラス MetaElement (集合構造の構成要因) を設計し、両者を多重継承することにより、SubStructure (部分構造) を設計した。これにより、部分構造は2つの釣合式を持つことになる。1つは、MetaStructure から継承した釣合式であり、これは、内部構造の各要素から釣合式を集積した部分構造内全節点に関する釣合式に当たる。もう1つは、MetaElement から継承した、全体構造からのメッセージ "give me Equation" に対して返される釣

合式であり、これは、部分構造内全節点に関する釣合式を結合節点に縮約した釣合式である。この関係を図-13に示す。

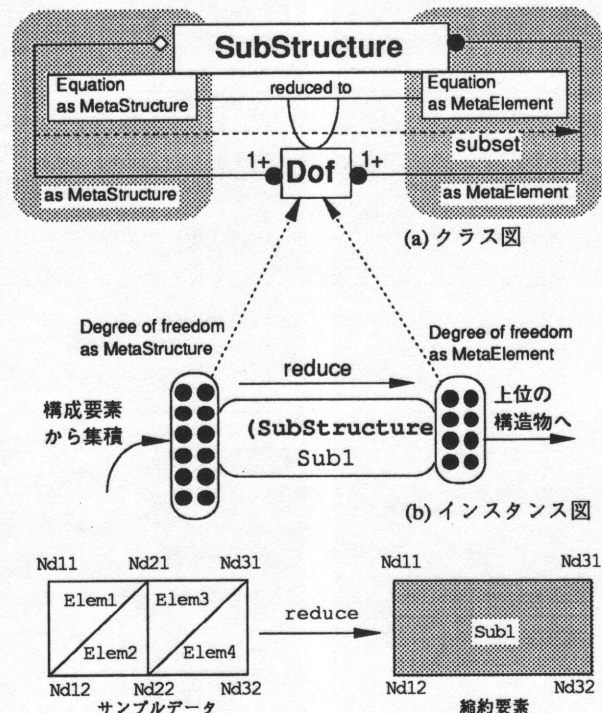


図-13 部分構造内の2つの釣合式の関係

自由度への参照リストも釣合式に伴って2つ持つことになる。即ち、内部全自由度と要素としての外部自由度であり、縮約はその2つの自由度情報を用いて行なわれる。多くの有限要素解析システムでは、自由度は節点の属性として定義されることが多いが、本システムでは、図7でも示したように要素および構造物との関連クラスとして分析されている。この分析の差が、部分法への拡張において大きく現れている。即ち、節点が自由度を管理している場合、何階層にも渡る階層構造の縮約を柔軟に行なうことは困難であるのに対し、本システムにおける分析に基づけば、それぞれの部分構造が内部全自由度と外部自由度を比較することにより縮合に関する情報を得ることができる。プログラム作成を主体とした開発を行なっている場合、このような、改善には、大きな労力を割くことになるが、OMT 法の場合は、分析/設計段階に時間をかけるので、実装に入る前段階でこれらの改善案が挙げられ、改善が容易となる。

今回の拡張における変更/追加項目は、

- ・集約に関する性質の抽出
- ・多重定義による部分構造の実装

の二項目である。多重継承を行ったことにより、部分構

造で新たに定義する基本操作は、縮約に関するもののみとなる。この部分構造からさらに導出クラスを派生することにより、各種部分構造を設計していくことができる。また、今回の拡張では、抽象レベルの高い位置での継承関係は変更したが、StructureおよびElementの性質は何も変えていないので、StructureおよびElementの導出クラスにはほとんど影響を与えていない。このように、オブジェクト指向の継承やカプセル化などの機能を用いてシステムを構築することにより変更などに対し柔軟で頑丈なシステムとすることができる。

## 6. 有限要素解析システムの実装

これまで述べてきた分析/設計をもとに、有限要素解析システムの実装をおこなった。実装には、演算効率や移植性、将来性などを考慮に入れてオブジェクト指向言語C++<sup>32)</sup>を用いている。C++の選定理由としては、

- ・ 現在最も多くの環境で実現可能
- ・ クラスライブラリ、開発ツールが充実
- ・ 手続き的な記述に関してはCと同等の処理速度を有する
- ・ C, FORTRANとの互換があり、これまでの研究成果を有効利用できる

等が挙げられる。Smalltalk等のインタプリタ系の言語に比べるとFast prototypingによるプログラム開発は困難になるが、OMT法では分析/設計に時間をかけるので実装にかかる労力は大きくならない。

実装段階まで至ったオブジェクトモデルを図-14に示す。これは、図-12の設計オブジェクト図に詳細を加えた形になっているが、ほぼこのままの形で、C++のヘッダファイルへマッピングされる。この他に、状態図やプロセス図に詳細を加えていくことにより、実装は自動化に近付いていく。現時点では、静的線形問題に関する実装を行っており、要素については、はり要素、トラス要素、二次元三角形要素、四角形要素、部分構造については、静的縮約部分構造について実装されている。現在実装されている要素のクラス階層を図-15に示す。

また、行列クラスに関するオブジェクト設計を図-16に示す。有限要素構造解析では、従来より、対称性やバンド性を利用してメモリの効率化が図られているが、index計算を隠蔽することにより、統一的オペレーターが提供できる<sup>33)</sup>。本システムでは、それらに加えて、参照カウントによるメモリ管理機構を追加している。

集合処理については、アイデンティティとして整数型のID番号を持つオブジェクトIdNumberを設計し、IdNumberを制御するリスト構造としてIdListを設計した。要素、節点、物性、形状、荷重、自由度など集合的扱い

が必要なものは、IdNumberを継承しIdListの制御を受けることによりシステムの運営を円滑にしている。IdListは、既存のListクラスを継承して実装される。それらの関係を図-17に示す。

また、NIH<sup>34)</sup>などの基本クラスライブラリの活用については、移植性や効率を考慮しながら検討しているが、現時点では自作のクラスライブラリを活用している。また、図形クラスライブラリ<sup>35)</sup>との融合は行っており、クラスブラウザなどの開発ツールは、フリーウェアを利用している。図-18にプログラムの作成例、図-19にサンプルデータの実行例を示す。

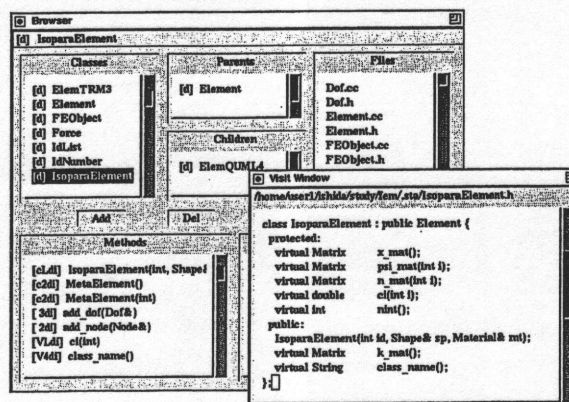


図-18 プログラム作成例

| Problem   |                 | shrink base      |                        |
|-----------|-----------------|------------------|------------------------|
| Dimension | DOF Number/Node | 1                | 2                      |
| 3         | 3               | { X Y Z }        | { X Y Rz }             |
| Node      |                 |                  |                        |
| 1         | 0.000e+00       | 0.000e+00        | { 1[1] 2[1] 3[1] }     |
| 2         | 3.500e+00       | 0.000e+00        | { 1[0] 2[0] 3[0] }     |
| 3         | 7.000e+00       | 0.000e+00        | { 7[0] 8[0] 9[0] }     |
| 10        | 7.000e+00       | 3.500e+00        | { 10[0] 11[0] 12[0] }  |
| Element   |                 |                  |                        |
| 4         | Beam            | NOE = { 1, 3 }   | Shape = 1 Material = 1 |
| 5         | Beam            | NOE = { 8, 9 }   | Shape = 1 Material = 1 |
| 6         | Beam            | NOE = { 6, 10 }  | Shape = 1 Material = 1 |
| Shape     |                 |                  |                        |
| 1         | Beam            | Area = 9.999e+03 | Isent = 3.000e+00      |
| Material  |                 |                  |                        |
| 1         | IsElastic       | E = 7.000e+00    | Nu = 0.000e+00         |
| Structure |                 |                  |                        |
| DOF       |                 |                  |                        |
| 4         | Node = 3        | Direct = 0       | E.C. = 0               |
| 5         | Node = 3        | Direct = 1       | E.C. = 0               |
| 6         | Node = 3        | Direct = 5       | E.C. = 0               |
| 7         | Node = 6        | Direct = 0       | E.C. = 0               |
| 8         | Node = 6        | Direct = 1       | E.C. = 0               |
| 9         | Node = 6        | Direct = 5       | E.C. = 0               |
| 10        | Node = 10       | Direct = 0       | E.C. = 0               |
| 11        | Node = 10       | Direct = 1       | E.C. = 0               |
| 12        | Node = 10       | Direct = 5       | E.C. = 0               |
| 1         | Node = 1        | Direct = 0       | E.C. = 1               |
| 2         | Node = 1        | Direct = 1       | E.C. = 1               |
| 3         | Node = 1        | Direct = 5       | E.C. = 1               |

図-19 テストプログラムの実行例

## 7. まとめ

有限要素法に関する概念に対してOMT手法を用いて分析、設計を行い、構造工学に対するオブジェクト指向方法論の適用方法を示した。これにより、全体構造、要素、節点、など概念的に意味のあるまとまりに関して、分析、設計、実装に渡る統一的表現が可能になり、問題に対する解析システムの対応が容易に把握できるようになった。また、部分法への拡張を行った場合、変更箇所を最小限に押さえることができた。今後、これらの拡張として、動的問題、非線形問題の分析・設計を通して、柔軟な動的構造解析システムの構築を進めていく。

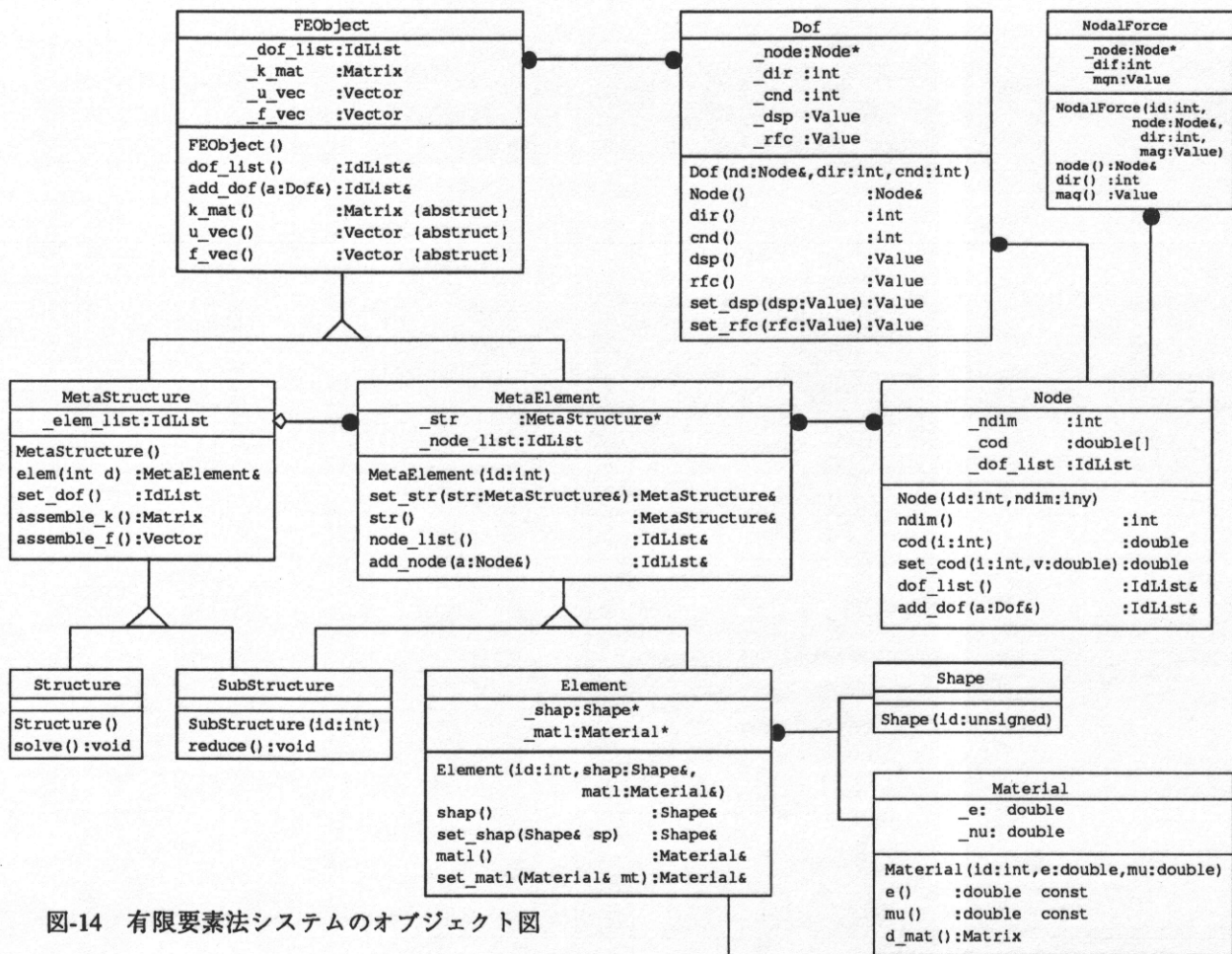


図-14 有限要素法システムのオブジェクト図

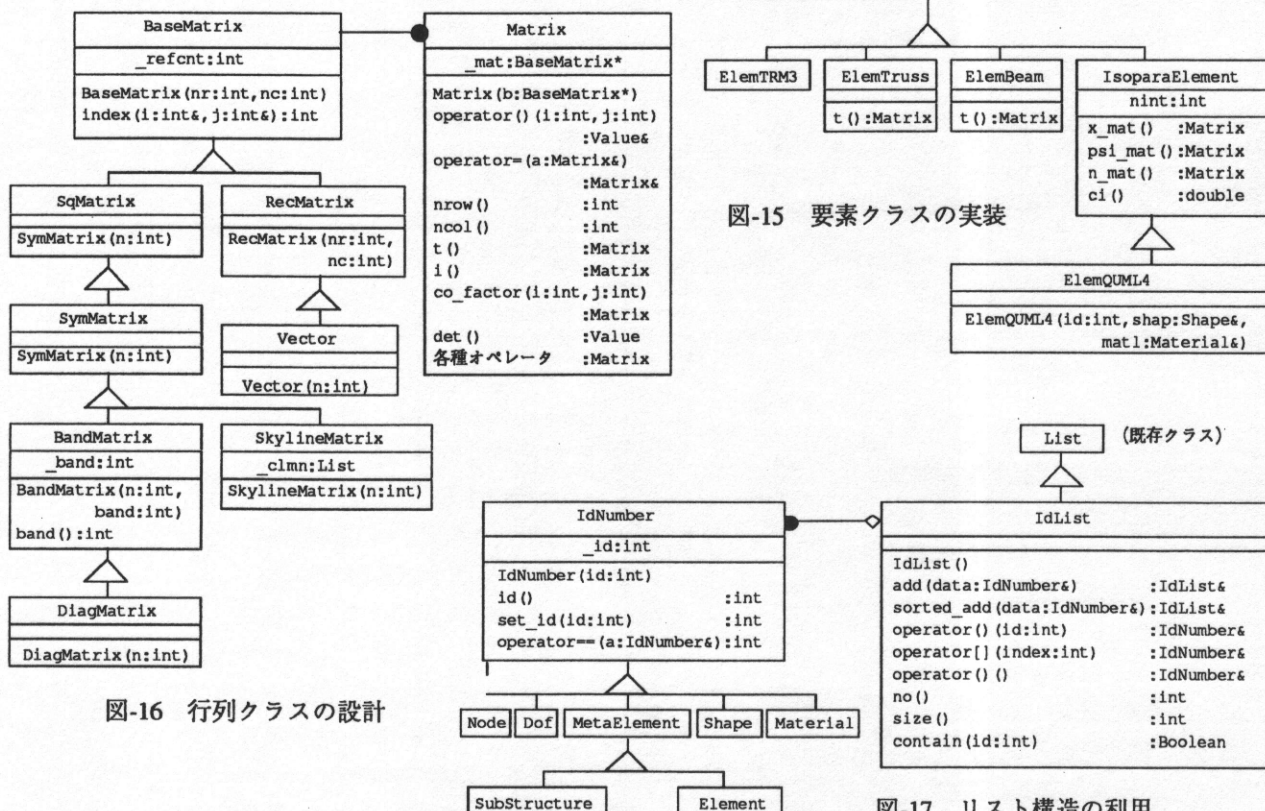


図-15 要素クラスの実装

図-16 行列クラスの設計

図-17 リスト構造の利用

## 謝辞

本論をまとめるに当たり、清水建設大崎研究室の勝倉裕博士、海老原学氏、広瀬啓一氏、名古屋大学の多賀直恒教授に貴重な助言を頂きました。記して謝意を表します。

## 参考文献

- 1) Dahl, O-J, Myrhaug, B. and Nygaard, K. : (Simula 67) Common Base Language, Pub. N. S-22, Norsk Regnesentral (Norwegian Computing Center), 1970
- 2) Goldberg, A. and Robson, D. : Smalltalk-80: The Language and its Implementation, Addison-Wesley, 1983
- 3) Meyer, B. : Object-Oriented Software Construction, Prentice-Hall, 1988
- 4) Booch G. : Object-Oriented Design, Benjamin / Cummings, 1991
- 5) Peter Coad, Edward Yourdon. : Object-Oriented Analysis Second Edition, Prentice Hall, 1993
- 6) Rumbaugh, J. et al. : Object-Oriented Modeling and Design, Prentice Hall, 1991, 訳: 羽生田栄一: オブジェクト指向方法論OMTモデル化と設計, トッパン, 1992
- 7) 矢川元基, 赤星保浩 : 不完全コレスキー分解の高速化: ベクトル計算機向き手法の開発, シミュレーション, 第8巻, 第1号, pp.49-55
- 8) 矢川元基 : 計算力学における並列化手法, 第39回応用力学連合講演会論文集, pp.1-8, 1989
- 9) Farhat, C., and Roux, F-X. : A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm, Int. J. Num. Methods in Engrg, Vol.32, pp.1205-1227, 1991
- 10) 吉岡顕, 矢川元基, 吉村忍, 曾根田直樹 : 大規模・超高速計算力学のためのネットワーク・コンピューティング手法の開発, 日本建築学会論文集A編, 第57巻, 第541号, pp.22-30, 1991
- 11) 三好俊郎, 高野直樹 : 構造解析におけるトラッキング/ステアリングシステムの開発, 第16回構造工学における数値解析法シンポジウム論文集, pp.607-610, 1992
- 12) 例えば, MacNeal, R. H. : The NASTRAN Theoretical Manual, the Macneal-Schwendler Corporation, 1972
- 13) 鷲津久一郎他 : 有限要素法ハンドブックI, II, 培風館
- 14) 石田栄介, 新美勝之, 福和伸夫, 中井正一, 多賀直恒 : オブジェクト指向による構造解析に関する研究, 建築学会学術講演梗概集, pp.1553-1554, 1992
- 15) 石田栄介, 新美勝之, 福和伸夫, 中井正一, 多賀直恒 : 有限要素構造解析のオブジェクト指向分析に関する研究, 東海支部研究報告集, 第31号, pp.165-172, 1993
- 16) Nakai, S. et al. : A Knowledge-Based Structural Analysis Based on Object-Oriented Approach, Microcomputers in Civil Engineering, No.7, pp.15-28, 1992
- 17) Forde, B.W.R., Foschi, R.B., and Steimer, S.F. : Object-Oriented finite element Analysis, Comput. & Structures, Vol.34, pp.355-374, 1990
- 18) 三木光範, 杉山吉彦, 内田雄治 : オブジェクト指向によるトラス構造解析, 日本機械学会論文集A編, 第57巻, 541号, 1991
- 19) Zimmermann, T., Dubois-Pelerin, Y., and Bomme, P. : Object-oriented finite element programming: I. Governing principles, Comput. Methods Appl. Mech. Engrg., Vol.98, pp.291-303, 1992
- 20) Dubois-Pelerin, Y., Zimmermann, T., and Bomme, P.: Object-oriented finite element programming: II. A prototype program in Smalltalk, Comput. Methods Appl. Mech. Engrg., Vol.98, pp.361-397, 1992
- 21) Dubois-Pelerin, Y., Zimmermann, T. : Object-oriented finite element programming: III. An efficient implementation in C++, Comput. Methods Appl. Mech. Engrg., Vol.108, pp.165-183, 1993
- 22) 三村泰成, 赤星保浩, 関東康祐, 青佐俊彦 : オブジェクト指向型有限要素解析システムの開発, 第16回構造工学における数値解析法シンポジウム論文集, pp.601-606, 1992
- 23) Baugh Jr., J. W., and Rehak, D.R. : Data Abstraction in Engineering Software Development, J. Computing in Civil Engineering, Vol.6, No.3, pp.282-301, 1992
- 24) Abdalla, J. A., and Yoon, C. J. : Object-Oriented Finite Element and Graphics Data-Translation Facility, J. Computing in Civil Engineering, Vol.6, No.3, pp.302-322, 1992
- 25) 大坪英臣, 久保田晃弘, 川村恭己 : EWS環境におけるFEM解析モデル生成のためのヒューマンインターフェース, 第16回構造工学における数値解析法シンポジウム論文集, pp.625-630, 1992
- 26) 中井正一, 勝倉裕 : オブジェクト指向と構造解析の融合, 第6回計算力学講演会, pp.51-52
- 27) 関東康祐, 青佐俊彦, 赤星保浩, 三村康成 : 非線形構造解析におけるオブジェクト指向アプローチ, 第6回計算力学講演会, pp.45-46
- 28) 大坪英臣, 鈴木克幸, 川村恭己 : オブジェクト指向アプローチによる大規模有限要素モデルの作成, 第6回計算力学講演会, pp.37-38
- 29) 石田栄介, 新美勝之, 福和伸夫, 中井正一, 多賀直恒 : 部分法を活用した有限要素動的構造解析のオブジェクト指向プログラミング, 第17回構造工学における数値解析法シンポジウム論文集, pp.471-476, 1993
- 30) 新美勝之, 石田栄介, 福和伸夫, 中井正一, 勝倉裕 : オブジェクトモデリング技術による有限要素法構造解析の分析と設計, 第17回構造工学における数値解析法シンポジウム論文集, pp.477-482, 1993
- 31) 長松昭男 : モード解析, 培風館
- 32) Stroustrup, B. : The C++ Programming Language Second Edition, Addison Wiley, 1991
- 33) R.B.Davies : Documentation for newmat07, an experimental matrix package in C++, 1993
- 34) Gorlen, K. E., Orlow, S. M., and Plexico, P. S. : Data Abstraction and Object-Oriented Programming in C++, John Wiley & Sons, 1990
- 35) Linton, M. A. et al. : InterViews Reference Manual Version 3.0.1, 1991