

オブジェクト指向による複合材料の力学的解析

三木光範

大阪府立大学工学部航空工学科

繊維強化型複合材料積層平板の剛性と強度の解析をオブジェクト指向に基づき行う方法を示す。これは、現在開発を進めている複合材料エキスパートシステムの一部をなす。構成素材の種類、繊維の配向角、及び積層順序などによって力学的異方性が著しく異なる積層板の力学的性能を簡単に評価するための積層シミュレーターをSmalltalkにより構築した。主要な成果は、解析に必要なクラスとその階層構造を明かにしたこと、エキスパートシステムとして一般的なメッセージを検討したこと、そして物理量を全てオブジェクト化したこと、などである。ここでの方法論は複合材料構造設計のための大規模な知識ベースシステムの構築に有用である。

Object-Oriented Approach to Mechanical Analysis of Composite Materials

Mitsunori Miki

Aeronautical Engineering, University of Osaka Prefecture
4-804, Mozu-umemachi, Sakai, 591, Japan

A trial development of an expert system for the mechanical analysis of composite materials is performed using an object-oriented approach. A knowledgebase is constructed for fibrous laminated composites by making appropriate classes and their hierarchical structure. The concepts for making classes and messages are described. Methods for dealing with synonymic messages and for dealing with dimensions such as those of stress, strain, modulus, and compliance are proposed. An example for calculating the stiffness, the compliance, and the strength under an in-plane load is presented and the proposed expert system is found to be useful for the analysis and design of composite materials.

1. 緒言

複合材料の最大の特徴は、軽量、高剛性、高強度、そして、設計可能性¹⁾である。極度の異方性を持つ複合材料は等方性材料である鉄鋼やアルミニウム合金と異なり、材料設計がまずいと逆に大変性能の悪い材料となる。このため、材料設計は重要となるが、既存の有限要素解析システムで複合材料構造物の設計を行う場合には複合材料の複雑さがある程度単純化する必要がある、このことは、ある意味で複合材料の可能性を制限していることになる。その上、設計変数の数があまりにも多く、また局所的最適解も多く、現在のシステムでは対応できない面もある。また、新しい材料であることから、素材、製造方法、材料特性、試験方法、力学的現象、および解析手法などに関して新しい進展が絶えずあり、既存のシステムではこうした状況に柔軟に対処できない。

こうした状況を解決するには、知識の増加に対してシステムを柔軟に拡張でき、しかも複合材料の専門家でもなくても容易に複雑な解析やシミュレーションのできる複合材料の力学的解析および設計のためのエキスパートシステムが必要になる。ここでは筆者等が開発した複合材料エキスパートシステムについて述べる。

2. 複合材料エキスパートシステムの基本構想

2.1 本システムが目指すもの

複合材料構造の設計には多くの満たすべき条件が存在し、設計者はそれらのバランスを考え、使用する素材や積層構成、各プライの繊維配向角などを変えながら満足解が得られる間で試行錯誤を繰り返す。本システムはそういった設計の試行錯誤過程を支援するものである。しかし、設計のためだけの専用エキスパートシステムではなく、設計者、技術者、研究者、教育者など複合材料に関係する人々がデスクサイドで専門的な検討、計算、調査などを手軽に行えるシステムとする。そのため、複合材料のための電子ハンドブックの役目も果たす必要がある。

ここで本システムの目指すものを挙げると、

(1) 複合材料のためのエキスパートシステム

従来、種々のエキスパートシステムが開発されてきたが複合材料を対象にしたものは皆無である。

(2) 固有知識に関する汎用のデスクトップエキスパートシステムとすること

特定の作業に使用する専用のエキスパートシステムではなく、設計者、技術者、研究者、教育者など複合材料の関係する人々がデスクサイドで専門的な検討、計算、調査を手軽に行えるシステムとする。

(3) 知識表現にオブジェクト指向を用いてオープンアーキテクチャにすること

汎用のエキスパートシステムは知識構造が開放型でなければならない。これによって、知識の拡充や個別的知識の増補が可能となる。開放型とするためには知識表現が一樣で、しかも処理効率の高いものでなければならず、オブジェクト指向は最適である。

(4) デスクトップエキスパートシステムとして複合材料のための電子ハンドブックの役目も果たすこと

オブジェクト指向型知識表現によって様々な要求に柔軟に対処できるシステムとなる。

(5) 物理的次元の処理を行い、数値に意味を持たせる

従来、計算機で扱う数値はたとえそれが物理的次元を持っていても、一般には計算機の中では数値としての属性しかなかった。ここではデータをオブジェクトとして扱うことにより物理的次元をもつ数値はすべてその属性をもつことになる。

(6) 材料データベースとの有機的な結合を取り入れる

材料特性値そのものをオブジェクトとして表し、それを用いた計算などは数値を媒介せずに行う。

(7) 個別的知識増補のためのガイドラインを構築する

知識構造を解放型とするとともに、その構造の詳細な解説書を作成し、新しく増補される知識の構築手順を示す。こうすることにより増補された知識の共有が可能になる。

このシステムは、素材、製造方法、材料特性、試験方法、力学的挙動、解析手法などに関して知識を有しており、例えば、炭素繊維強化エポキシ材料の種類を知る場合には、

CFRP showTypes.

とインプットする。また、

FW explain.

とインプットするとFW成形法についての説明が得られる。ユーザーはシステムに蓄えられた知識を呼び出

す最小限の言葉を学ばよく、しかもその言葉は自然言語に近いので、容易に使用することができる。

2.2 知識表現と使用言語

本システムのような試行錯誤型のシステムに最も適する知識表現として挙げられるのはオブジェクト指向型表現である。オブジェクト指向とは対象領域に現れる概念や物理的実体をオブジェクトというもので表現し、それらオブジェクトの処理をメッセージと呼ばれる簡単なプロトコルによって制御するものである。

オブジェクト指向言語はFORTRANやBASICのようなこれまでの手続き型言語と異なり、図1のように、(弾性定数という)オブジェクトに(逆数を求めよという)メッセージを送ることにより(コンプライアンスを求めるといふ)処理を行うというパラダイムを持っている。オブジェクト指向言語はそのモジュール性と階層的構造によってソフトウェアの創造、設計、および保守に著しい生産性の向上をもたらすほか、人間の思考方法に近いので、機械構造の解析や設計といった分野でも極めて有効である²⁾。

オブジェクトとは図2に示されるように、内部状態とメッセージを受けたときに行うべきこと、返答すべきことについての知識、演算をもつ抽象データ型のモジュールである。内部状態として持つ値もオブジェクトであり、それらのオブジェクトが図3のようにネットワーク状の参照関係を形成している。共通の性質を持つオブジェクトはまとめてクラスとして定義される。そしてクラスから生成された具体的な個々のオブジェクトのことをインスタンスとよぶ。

ここで重要なのはクラスは階層構造を成しており、階層構造の中の下位のクラスはその上位のクラスの性質を継承することである。下位のクラスには上位のクラスとは異なるそのクラス特有の性質だけを記述すればよい。この継承機構によって複雑なクラスの挙動もコンパクトに記述できる³⁾。これを差分プログラミングと呼ぶ。

オブジェクト指向型知識システムではクラスを定義することにより構造化された形で知識が組込まれる。具体的な問題のモデルは、組込まれたクラスのインスタンスとして生成されたオブジェクトの集合で表現される。推論は推論制御部というようなものによって行

われるのではなく、各オブジェクトがメッセージパッシングによって自己の手続きを起動させることにより自律分散的に行われる。図4は設計対象のある設計項目を変化させたときの他の項目への波及を模式的に示したものである。

本システムではシステム構築用言語にオブジェクト指向言語として代表的なSmalltalk-80を用いる。この言語はMacintosh, PC-9800, NEWS, Sun, Appollo, HP9000, Tektronics, Xerox などの計算機で用いることができる。

3. 複合材料の力学的解析へのオブジェクト指向的アプローチ

3.1 面内荷重下での複合材料の剛性および強度解析

ここではオブジェクト指向による解析の例として面内荷重下での積層型繊維強化複合材料の面内剛性、コンプライアンスおよび強度を解析することを考える。

この手順は以下の通りである。

- (1)積層材料の決定: すなわち、材料定数とプライ厚さを与える。材料定数は工学的弾性定数4個、強度定数5個である。
- (2)積層構成の決定: 各層の繊維配向角を決定する。
- (3)工学的弾性定数4個から繊維軸系の弾性定数を計算する。強度定数5個から破損則における強度パラメータを計算する。
- (4)繊維軸系の弾性定数(3×3行列)を各層の繊維配向角に従って回転座標変換する。これらを合計して全体の面内剛性(3×3行列)を求める。
- (5)面内剛性の逆行列として面内のコンプライアンスを求める。
- (6)与えられた面内荷重とコンプライアンスから面内ひずみ(3要素)を求める。
- (7)適当な破損則、たとえば最大ひずみ則で各層の破損状態を検討し、そののち全体の初期破損を検討する。

こうして検討した結果、破損が生じるようであれば積層構成を変更して、再度同様の検討を行う。

3.2 クラスおよびクラス階層構造

解析や設計をオブジェクト指向のパラダイムの下に行うには、対象を適切なクラスに分類し、その階層構造を与え、そして、各クラスおよびそのインスタンス

に固有の手続き(メソッド)を設定しなければならない。

オブジェクト指向解析に対して必要なクラスおよびクラス階層構造を考える指針は、人間が認識する現実世界をなるべくそのまま表現することである。そうすれば、(1)エキスパートシステムの開発者およびユーザーにとって自然であり、しかも、(2)知識の増加に柔軟に対応できるものと考えられる。

ここでは複合材料積層平板の面内荷重下での挙動を解析することを例にとって説明する。すなわち、図5のように、繊維強化複合材料のラミナがその繊維配向角を変えて積層され、その平板がある面内の荷重下でどのように変形するかを考える。このとき、オブジェクトとして、素材、ラミナ、積層平板、積層構成、弾性定数、面内剛性、面内コンプライアンス、面内荷重、面内ひずみ、面内応力、層応力、厚さ、そして、座標などが考えられる。素材のサブクラスとしては複合材料、そのサブクラスとして炭素繊維複合材料、そのサブクラスとしてたとえばT300/5208がある。また、面内弾性定数、面内剛性、面内コンプライアンスなどは 3×3 のマトリックスであり、これをそれぞれのスーパークラスとすることができる。また、面内荷重、面内ひずみ、面内応力は3次元ベクトルであり、これをそれぞれのスーパークラスとすることができる。こうして設定されたクラスおよびクラス階層構造を図6に示す。重要なことは、下位のクラスは上位のクラスの性質を継承する(インヘリタンス)ことである¹⁰⁾。この特記すべき性質により、複雑なクラスの挙動を容易に記述することができる。

3.3 インスタンス変数

クラスが設定されると次にそのインスタンスの内部状態を表すインスタンス変数を設定する。このとき次の三つの指針を考えた。

- (1) クラスの挙動がユニークに決定されること。
- (2) 変数はそれぞれ独立であること。
- (3) 複雑になり過ぎるときはクラスを分割する。

図6に示したクラスのうち、 3×3 のマトリックスおよび、3次元ベクトルについてはインスタンス変数は単純に各要素となる。一方、素材のインスタンス変数は面内弾性定数、面内強度定数、密度、価格、およ

び材料名とする。これらはすべてそれぞれに関連するクラスのインスタンスであり、たとえば面内弾性定数は直交異方性工学的弾性定数のインスタンスであり、それはまた、縦方向ヤング率、横方向ヤング率、せん断弾性定数、ポアソン比の4つのインスタンス変数を持っている。

一考を要するのは複合材料平板と積層構成のクラスである。積層平板が単一のラミナで構成されている場合には、複合材料平板に、ラミナ、積層構成、そして、形状というインスタンス変数を設定し、ラミナのインスタンスには素材と厚さを、積層構成のインスタンスには各層の配向角と層数などを設定すればよい。しかしながら、ハイブリッド積層やサンドイッチ積層を考えるときには、積層構成の中に素材および層の厚さを追加しなくてはならない。ここではまず、単一のラミナからなる積層板に話を限定し、続報で知識の追加がどのように柔軟に行えるかという考察を行う。図3は重要なクラスについてインスタンス変数のネットワーク構造を示したもので、オブジェクトの内部状態を表す変数がさらに別のオブジェクトから成り、複雑な現実世界が極めてシンプルに表現されていることが分かる。

3.4 メソッドの設計

クラス構造およびインスタンス変数が設定されると次にデータ処理の制御を行うメソッドを設計する。

オブジェクト指向がこれまでの手続き型アプローチと本質的に異なる点は、特定の操作がそのプログラム中で普遍的な意味をもたない点にある。オブジェクト指向ではメッセージに対する応答はすべてそれを受けたオブジェクト自身が決めることであって、普遍的な意味は考えなくてもよい。この概念によって、人間にとって自然なメッセージを設定することができ、また、同一のメッセージでクラスによって異なる処理、あるいは、異なるメッセージで同一の処理を実現できる。また、物理的次元をシステムに持ち込むことができ、ユーザーは計算機の中に実態を想定することが容易になる。これらについて説明する。

(1) 人間にとって自然なメッセージの実現　これはエキスパートシステムのユーザーインターフェイスにおいて極めて重要な意味を持っている。設計の指針

は、ユーザーのレベルを設定し、そのユーザーには説明無しで分かるようにすることである。たとえば、マトリックスの各要素の値をゼロにするには、自然なメッセージとして、

aMatrix clear (1.1)

aMatrix initialize (1.2)

aMatrix setZero (1.3)

aMatrix putZero (1.4)

aMatrix new (1.5)

などが考えられる。どれを選ぶかは開発者の判断によるが、もし、どれも捨てがたいのであれば、たとえば clear というメッセージのメソッドとして各要素をゼロにする具体的な手続きを設定し、他のメッセージはすべてレシーバオブジェクトに clear を送信するようにメソッドを設計する。こうして、システムは同類語辞書を持つことになり、自然な応答に答えるようになる。これをポリモアフォロジと呼ぶ。

一方、" + " という記号は和を求めるものとして一般的であるから、ベクトルやマトリックスに対してもこれが使えるようにした。すなわち、

aVector + aVector (2.1)

aMatrix + aMatrix (2.2)

については各要素の和を求める演算が行えるようにした。これは実際の手続きとしては異なっているが、概念として同じものを抽象的に取り扱う際に重要な考え方である。これをオーバーローディングと呼ぶ。

(2) 物理的次元の保有 一般の手続き型言語では各変数が物理的次元を保有することは困難である。たとえば、図7(a)ではひずみが応力とコンプライアンスによって計算されているが、各変数は単なる数を表し、物理的次元は人間の頭の中にあるに過ぎない。しかし、オブジェクト指向言語では適切にクラスおよび対応するメッセージを設定することにより、図7(b)では正しくひずみが求められ、そのひずみを応力で除すことにより弾性定数が正しく求められる。一方、(c)では右辺で定数倍された応力が求められ、それを anInplaneStrain という変数に代入してもやはり応力であり、それを応力で除すときにエラーとなる。なぜなら、応力というクラスは「/応力」というメッセージが理解できないからである。すなわち、応力というクラ

スのインスタンスは「*コンプライアンス」というメッセージにはひずみのインスタンスを返し、「*数値」というメッセージには応力のインスタンスを返すように設計することが可能となる。

物理的次元が保有されることにより、開発者やユーザーの不注意による誤りが防止され、しかも、シミュレーションにおいては実体の様相がよく伝えられる。

このような観点で設計したメッセージは以下のカテゴリに分類される。それぞれの例を図8に示す。

(1) インスタンス生成 各クラスのインスタンスを生成する。図8(a)ではT300/5208, ラミナ, そして積層構成のインスタンスが生成されている。

(2) アクセス 各インスタンス変数とのアクセスを行う。インスタンス変数はこのメッセージでのみアクセス可能である。図8(b)ではラミナの材料が、材料の弾性定数が、そして積層構成の5層目の配向角がそれぞれ応答される。

(3) テスト オブジェクトのテスト。図8(c)は積層構成が対称かどうか、および積層構成がハイブリッドかどうかのテストをおこなう、応答は真か偽である。

(4) エラー通知 エラーメッセージをだす。図8(d)では積層板は対称ではないとのエラーメッセージが表示される。

(5) 計算 インスタンス変数から簡単に計算される変数を求める。

図8(e)では積層構成の片側積層数が計算され、また、矩形板の面積が計算される。

(6) 算術 算術的操作を行う。ただし、数学的および工学的な操作を含む。このメッセージはその表現が人間にとって親しみやすいため有用である。図8(f)では片側積層に関する面内剛性を2倍して全体の面内剛性を求め、面内荷重を厚さで除して平均面内応力を求めている。

(7) 数学的関数 数学的および工学的関数を求める。図8(g)では inverse というメッセージが弾性定数に送られるとコンプライアンスを返す。もし、コンプライアンスに送られると弾性定数を返す。

(8) 解析 複雑な計算を行う。図8(h)では複合材料平板の面内剛性が求められ、繊維方向を60度回転

させたときの弾性定数を求めたものである。

(9) 表示 オブジェクトの内容を表示する。図8(i)では面内剛性の6個の独立した要素が画面に表示され、また、面内ひずみの3個の要素が画面に表示される。

送られたメッセージがどのように理解されるかの例として、図9に積層板の面内剛性を求めるためのメソッドを示す。

4. 実行結果

こうして必要なクラスとメソッドが作成されると、これで複合材料積層板のための解析エキスパートシステムが完成する。たとえば、ある積層板の面内剛性、コンプライアンス、与えられた面内荷重下でのひずみをそして破損状態を求める場合には図10のようなプログラムを組むだけでよい。

ここでは、素材としてT300/5208を用い、全積層数8層の対称積層板を考え、中央面からの配向角は 30° 、 -30° 、 30° 、 -30° であるとする。平板の形状は矩形で座標(0,0)と(0.3,0.1)(単位:m)を対角線の端点とする。CompositePlateのインスタンスに madeOf:材料 StackingSequence:積層構成 geometry:形状 というメッセージを送ることにより具体的に複合材料平板が生成する。あとはこの平板の面内剛性の表示、面内コンプライアンスの表示を行っている。最後に、平板のx方向荷重45kN、y方向荷重17kN、せん断荷重-15kNのもとでの面内ひずみが求められている。

このようなプログラムは作成が容易で、しかも他の人にとっても可読性が高く、プログラムの保守も極めて簡単である。重要なことは、一旦、基本クラスとメソッドがコンピュータのシステムに組み込まれると、あとはコンピュータのスイッチを入れるだけでこうした知識を自在に利用することができることである。図11はこうした状況を示している。図12は図10のプログラムの実行結果である。なお、使用した計算機はMacintosh IIである。

5. ユーザーインタフェースについて

5.1 開発目標

本システムは詳しいマニュアルがなくても容易に操

作できる必要がある。そこでユーザーインタフェースに要求される第一の条件はユーザーフレンドリーなことである。その要求を実現する一つの方法として視覚的ユーザーインタフェースが挙げられる。システムの操作はアイコンやメニューをマウスで操作することによって行う。例えば、複合材料積層板の積層をシステム上でシミュレーションする場合を考える。

図13に示すようにメニューから積層作業を選択しプリプレグ置場を示すウィンドウをオープンする。そこにはプリプレグの山を表すアイコンが表示されている。それぞれ0度、+45度、-45度、90度材である。マウスでそれらの山から一枚ずつ適当に取り出し、作業台上で積層してゆく。こうしてできた積層板に大域変数名を付ける。以後、この変数名を用いてワークスペース上で積層板の剛性を求めたりできる。解析結果は必要に応じてグラフや表によって表示される。

4.2 開発環境

本システムの開発言語であるSmalltalk-80にはグラフィック機能やウィンドウ、メニュー、カーソルといったユーザーインタフェースを強力に支援するクラスがあらかじめプログラミング環境として用意されている。よって、それらの機能をシステムの目的に合わせてカスタマイズすることで上記のユーザーインタフェース機能を比較的容易に満足させることができる。

4.3 開発手順

複合材料に関する技術者、研究者などを対象にユーザーインタフェースに対する要求を調査する。次にそれらを整理し、システムの専門家と共同してユーザーインタフェースの開発を進める。開発は、プロトタイプを構築してそれをユーザーに評価してもらい、ユーザーの希望を取り入れプロトタイプを改良する、というサイクルによって行われる。この場合、ユーザーの要求とシステムの複雑さとのバランスは重要な課題である。

5. ユーザーによる知識の拡充について

5.1 Smalltalk における知識拡充

Smalltalk はそのクラス階層構造と継承機構により知識の拡充が容易である。新しい知識を表現するクラスをシステムに組込む場合、まずそのクラスの上位概

念に相当するクラスを見つけ、そのサブクラスとしてクラスを追加する。継承機構によって新たなクラスに対する記述量が少なくてすみコンパクトな知識表現が行える。

オブジェクト指向ではシステム内の処理はオブジェクトがメッセージを送信し合うことによって行われる。その際、メッセージを送る側のオブジェクトは送るべきメッセージさえ知っていればよく、送り先のオブジェクトの内部表現には依存しない。そのため、メッセージセレクトが合っていれば、既存のシステムに変更を加えることなく新たなクラスを追加できる。

5.2 問題点

Smalltalk は知識の増補が容易に行える。しかし、ユーザーが各々の方法で個別的知識の増補を行うと、それらの知識を各ユーザーが個人的に利用する場合には問題はないが、他のユーザーと知識を共有しようとする場合、知識の整合性がとれず、個別的知識をユーザー間で互いに共有することができない。

ユーザーが増補した個別的知識を共有できるようにするためには知識増補のためのガイドラインを構築する必要がある。クラス構造や用語について統一し、そのガイドラインに従って知識の増補を行うことで知識を共有でき、知識ベースの有効利用につながる。

本稿ではオブジェクト指向による開放型知識構造を有する複合材料エキスパートシステムの基本構想と具体的な強度解析方法の構築について述べた。本研究に関しては現在、産・官・学からなる研究会（ECOM研究会）が日本材料学会複合材料部門委員会内に組織されており、プロトタイプ構築のための作業が進められている。なお、積層作業用のウィンドウの開発に当っては富士ゼロックス情報システム(株)の青木 淳氏に大変お世話になった。

文献

- (1) 三木, 最適材料設計, 材料, 38-425, (1989), 105-111.
- (2) The Smalltalk-80 Programming System Reference Guide and Release Notes, (1988), ParkPlaceSystems, Inc.
- (3) Goldberg A. and Robson D., Smalltalk-80: The

Language, Xerox Palo Alto Research Center (1987)

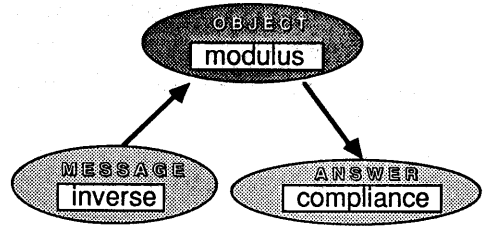


図1 メッセージ送信

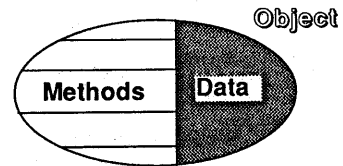


図2 オブジェクト

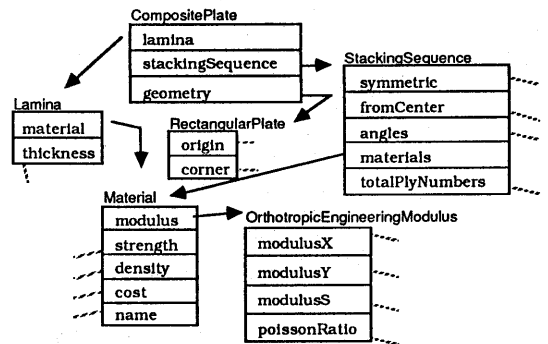


図3 インスタンス変数間の関係

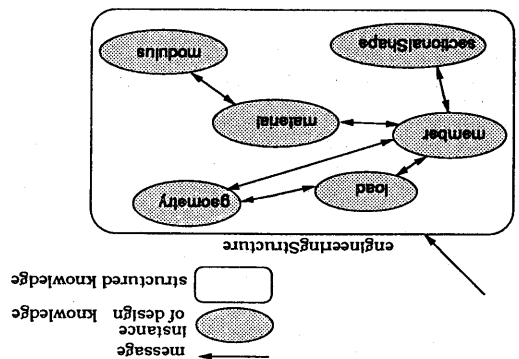


図4 メッセージのネットワーク

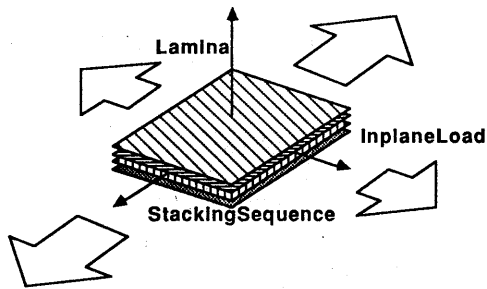


図5 解析対象のモデル化

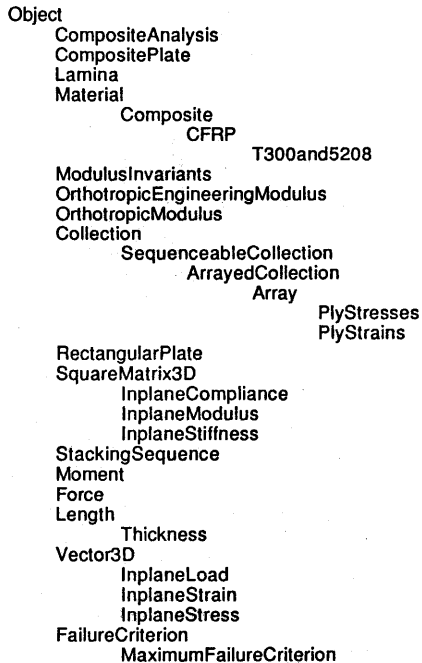


図6 作成したクラスおよびクラス階層構造

```

compliance = 5.3e-10
stress = 4.7e7
strain = compliance * stress (a)

anInplaneCompliance ← InplaneCompliance new.
anInplaneCompliance a11:1.5e-9 a12:3.2e-8 a13:0
a22:2.4e-8 a23:0 a33:3.7e-8.
anInplaneStress ← InplaneStress new.
anInplaneStress e1:4.7e7 e2:2.1e6 e3:-6.8e7.
anInplaneStrain ← anInplaneCompliance * anInplaneStress
anInplaneModulus ← anInplaneStress / anInplaneStrain (b)

anInplaneStrain ← anInplaneStress * 5.3e-10
anInplaneModulus ← anInplaneStress / anInplaneStrain (c)
  
```

図7 次元の保有

1. Instance creation
 - T300and5208 new (a.1)
 - Lamina material:argument1 thickness:argument2 (a.2)
 - StackingSequence newOnTotalPlyNumbers:argument (a.3)
2. Accessing
 - aLamina material (b.1)
 - aMaterial modulus (b.2)
 - aStackingSequence angleOfPly:5 (b.3)
3. Testing
 - aStackingSequence isSymmetric (c.1)
 - aStackingSequence isHybrid (c.2)
4. Error notification
 - aStackingSequence errorNotSymmetric (d.1)
5. Calculation
 - aStackingSequence halfPlyNumbers (e.1)
 - aRectangularPlate area (e.2)
6. Arithmetic
 - totalStiffness ← halfInplaneStiffness * 2 (f.1)
 - averageStress ← anInplaneLoad / thickness (f.2)
7. Mathematical function
 - aCompliance ← anInplaneModulus inverse (g)
8. Analyzing
 - aStiffness ← aCompositePlate inplaneStiffness (h.1)
 - anOffAxisModulus ← anOnAxisModulus rotate:60 (h.2)
9. Displaying
 - aStiffness show (i.1)
 - anInplaneStrain show (i.2)

図8 メッセージのカテゴリ

InplaneStiffnessOnSymmetric

"Calculate the inplane stiffness of a symmetric laminate"

```

| onAxisModulus offAxisModulus eachAngle
plateThickness temp |
stackingSequence ← self stackingSequence.
(stackingSequence isSymmetric) ifFalse:[
stackingSequence errorNotSymmetric].
onAxisModulus ← self lamina onAxisModulus.
offAxisModulus ← InplaneModulus new clear.
1 to:stackingSequence halfPlyNumbers
do:[:index |
eachAngle ← stackingSequence angleAt:index.
offAxisModulus ← offAxisModulus +
(onAxisModulus rotate:eachAngle)].
offAxisModulus ← offAxisModulus / stackingSequence
halfPlyNumbers.
plateThickness ← self lamina thickness *
stackingSequence totalPlyNumbers.
temp ← InplaneStiffness new.
temp ← (offAxisModulus * plateThickness)
↑ temp
  
```

図9 メソッドの一例


```

| laminaCF layUp shape composite inplaneLoad
inplaneStrain plyStrains strainCriterion |
laminaCF + Lamina material:T300and5208 new
thickness:1.25e-3.
layUp + StackingSequence newOnTotalPlyNumbers:8.
layUp symmetric:true;
fromCenter:true;
ply:1 angle:30;
ply:2 angle:-30;
ply:3 angle:30;
ply:4 angle:-30.
shape + RectangularPlate origin:(0@0)
corner:(0.3@0.1).
composite + CompositePlate new.
composite madeOf:laminaCF;
stackingSequence:layUp;
geometry:shape.
composite inplaneStiffness show.
composite inplaneCompliance show.
inplaneLoad + InplaneLoad new
longitudinal:4.5e4;
transverse:1.7e4;
shear:-1.5e4.
inplaneStrain + composite
inplaneStrainOnLoad:inplaneLoad.
inplaneStrain show.
plyStrains + composite plyStrainsOnInplaneStrain:inplaneStrain.
plyStrains show.
strainCriterion + MaximumStrainCriterion new
eps1L:-8e-3 eps1U:8e-3 eps2L:-20e-3 eps2U:3.8e-3 eps6:-9e-3.
strainCriterion failureChecksOn:plyStrains

```

図10 積層板の面内剛性，コンプライアンス，ひずみ
および破損を解析するユーザープログラム

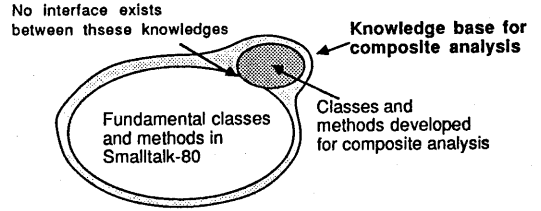


図11 知識の獲得

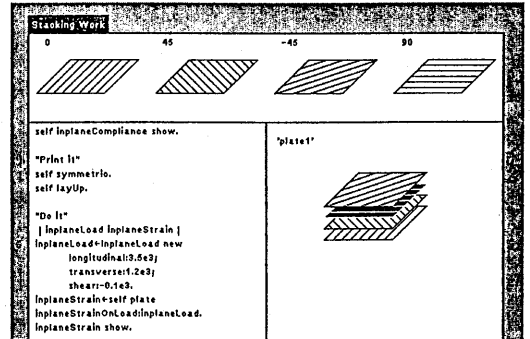


図13 積層作業のグラフィックスユーザーインターフェース

System Transcript	System Browser
<pre> Strain2=0.00455433 Strain6=-0.00268357 Ply-2 Elements of inplane strain Strain1=0.00492496 Strain2=0.00101815 Strain6=-0.00139965 Ply-3 Elements of inplane strain Strain1=0.00138879 Strain2=0.00455433 Strain6=-0.00268357 Ply-4 Elements of inplane strain Strain1=0.00492496 Strain2=0.00101815 Strain6=-0.00139965 failure check on maximum strain ply 1---Failure on transverse tension ply 2---not failure ply 3---Failure on transverse tension ply 4---not failure first ply failure=true failure checks over </pre>	<pre> Interface-Clocks System-Support System-Changes System-Compiler Files-Streams Files-Abstract Files-Macintosh Interface-Macintosh Composite-Analysis System-Printing CompositeAnalysis CompositePlate FailureCriterion InplaneCompliance InplaneLoad InplaneModulus InplaneStiffness InplaneStrain InplaneStress instance class laminaCF layUp shape composite inpla plyStrains strainCriterion laminaCF + Lamina materia:T300and5208 n thickness:1.25e-3. layUp + StackingSequence newOnTotalPlyT layUp symmetric:true; fromCenter:true; ply:1 angle:30; ply:2 angle:-30; ply:3 angle:30; ply:4 angle:-30. shape + RectangularPlate origin:(0@0) corner:(0.3@0.1). composite + CompositePlate new. </pre>

図12 実行結果