

オブジェクト指向による FEM 解析支援システムの構築法[†]

永 沢 茂*1・佐久田博司*2・宮田保教*2

ABSTRACT The paper deals with a method of designing a data generator for an FEM analysis system. The proposed analysis support system has been developed in Smalltalk/V, an object oriented language, on a personal computer. The system has a special multieditor *Hier Browser*, hierarchical online manual, and a window for the format information of each *Keyword*. In the developing process, the following features are found to be clear: 1) Input data of general purpose FEM systems consist of standardized data format for each function. By introducing *Keyword-object* into these functions, it makes possible to hide or mask the format information and simplifies the construction of design support system. 2) Data structures of *Keywords* are classified four patterns (partitioned, normal, variable and branch). 3) Unit sets of editing windows for *Keywords* have two list panes and one text pane. 4) By establishing common messages between *Keyword* and *Hier Browser*, it simplifies these class methods and clear essential facilities. 5) By implementing parallel retrieval with list-panes for classes and instances, high performance has been achieved. 6) Objective structure of the design rules is evaluated as a virtual action operator. **Keywords:** object-orientation, man-machine interface, general purpose FEM analysis, pre-processor design, data masking, virtual action operator.

1. ま え が き

数値解析法として広く用いられる FEM (有限要素法)¹⁾には、個々の問題に応じていちいちプログラムを開発する必要がないように、多目的の機能を盛り込んだ汎用プログラム (MARC, NASTRAN, ABAQUS, SAP など) がよく知られている。

しかしこれら汎用プログラムは、多機能のために一般にマニュアル情報が膨大であり、しかも機能の組み合わせによって複雑な操作を要するために、高度な経験的知識を必要とする。このため、マニュアル類への依存性を軽減し、解析分野に応じた適正な入力オプションや書式文法を与える知識支援システムが必要であり、実際の支援システム^{2)~4)}や知識システムの構築方法⁵⁾などが研究されている。

ところが、2)~4)は MYCIN 型、5)は論理型で、何れも IF/THEN 形式のシステムであるため、原理

的に知識の規則性能に依存して、対話のスタート点から分岐方法が一意に進められる。このため、要求者がその時点で最も欲する質疑応答にならず、特に部分的な編集をしたい場合や、作業者が優先順位をある程度明確に持つ場合、あるいは意味不明の問答を後回しにできない場合、好ましいとは言えない。必要なルールを組み入れることでこれらの問題に対処することは可能であると考えられるが、複雑なルールが多くなり、システム構成上非常に不利になる。

この問題点は、汎用 FEM の知識の空間構造が、解析分野の機能毎に規格化されているという特質を持ち、IF/THEN 形式の表現とは構造が異なることに起因すると考えられる。そこで著者らは、ある程度の知識を有する経験者を対象とし、規格化された知識空間内で効果的な支援システムを実現するために、階層状知識表現とその実現手段を提案する。すなわちここでは、汎用 FEM の入力データは機能が規格化されているため、データ構造が第一義で、データ同志の相互関係を処理する手順は従属的に発生するものとする。そして、規格化されない経験的規則の機能については、IF/THEN 形式の知識として併用するが、規格化部分を隠蔽した形態とする。

この考え方には、一般的な IF/THEN 形式に比べて、経験知識と文法知識とを明確に分類し、IF/

Object-Oriented Construction Method for an FEM Analysis Support System (Objective Structure for Data Format and Hierarchical Knowledge Expression). By Shigeru Nagasawa, Hiroshi Sakuta and Yasunori Miyata (Information Processing Center, Nagaoka University of Technology).

*1長岡技術科学大学情報処理センター

*2長岡技術科学大学機械系

†1991年4月26日受付, 1991年7月1日再受付

THEN 型の規則を簡素化できるため、能率、コストの面ではるかに有利という特徴がある。

この考え方を具体化するには、データをオブジェクトモデルにあてはめ、部品化された入力データの特徴をオブジェクト指向の考え方^{6),7)}で整理する方法が適している。そこで試作システムでは、Smalltalk 言語^{8),9)}を用いた。この上で前記階層構造を実現するには、マルチウインドウとマウス・メニューを用いるのが自然であり、マニュアルの書式文法をプログラム化しやすい。

以下では、まず MARC と SAP を題材に汎用 FEM の入力データの構造を考察し、その表現に階層状知識が適していることを示す。次に、経験的設計知識と入力データとの関係構造を明らかにし、支援システムの基本設計を述べる。続いて、階層状知識の編集環境特性を考察した後、最後に MARC のために Smalltalk で開発したシステムについて、その概要と、実際の問題に適用した場合の特徴を示す。

2. 入力データの形式

汎用 FEM の入力データは、一般に、境界条件、物性値などの機能別にラベル（オプション名）を付けており、必要な数値データがそれに続いて配置される。

表 1、表 2 に MARC, SAP の入力データに関する主な機能項目とその順序構成を示す。MARC は非線形解析によく使われ、SAP は線形解析に使われる。機能の分類方法は、このように汎用プログラム毎に異なるが、各 FEM コードの個々の項目は、類似の構造と役割を共通に有することが一般的である。なお、SAP などのオプション名を明示的に含まない場合であっても、機能別に構造体データの集合とみなすこと

表 1 MARC 入力データの構成

機能分類	主な項目と内容
パラメータカード	ジョブ識別用の表題情報, 要素型の選定, 解析モデルの選定 (熱, 弾塑性変形, 振動など), 連成問題の選定, 配列空間の予約など
モデル定義カード	要素・節点空間の指定, 要素と節点の定義, 物性値の定義, 境界拘束条件, 初期状態値の設定, 収束計算の許容誤差の設定, 出力情報の選定など
履歴定義カード	増分量の定義, 境界拘束条件の変更, 出力情報の変更など

表 2 SAP 入力データの構成

機能分類	主な項目と内容
先頭カード	ジョブ識別用の表題情報
統制カード	要素・節点空間の指定, 配列空間の予約, 解析モデルの選定 (静, 動解析), 診断モード / 実行モードの指定など
節点データ	座標系 (直交, 円筒系) の指定, 節点の定義, 拘束条件, 温度場の指定など
要素データ	物性値の定義, 体積力の指定, 要素型の指定, 要素の定義
荷重/質量データ	節点毎の集中荷重, 質量 (振動用) の定義
荷重係数	組み合わせ荷重の定義
動的解析	振動解析の条件指定

は可能であるから、該当する解析機能があるならば、仮想ラベル（オプション名）を付与することは、一般に可能である。

MARC と SAP をもとに、入力データの書式構造を一般化すれば、図 1 のように分類できる。ここで、 a_1, \dots, c_1, \dots は、スカラー量としての数値データであり、 $LineVector_1, \dots, SubLineVector_1, \dots$ は、行ベクトルとしての数値データ列を表わす。Partitioned-LABEL₁ などは、表 1, 2 の機能分類に該当するラベルである。

3. 入力データのモジュール構造

入力データにおいて、ラベル名とそれに続く数値デ

```

LABEL1 Characteristic Keyword with normal attribute data
      a1,a2,a3,...,aN
LABEL2 Characteristic Keyword without attribute data
      :
LABEL3 Characteristic Keyword with branch attribute data
      b1,b2,...,bM,(bM+1,...,bM+K)
LABEL4
      :
LABELP1 Characteristic Keyword with
      c1,c2,c3,...,cL
      variable attribute data
      LineVector1
      :
      LineVectorJ1
      SubLineVector1
      :
      SubLineVectorJ2
      SubsubLineVector1
      :
      SubsubLineVectorJ3
      :
Partitioned-LABEL1 Partitioned Keyword without
LABELP2 attribute data
      :
    
```

図 1 入力データの書式構造

ータは、1つのデータ構造体と解釈できる。この構造体は、図1に示される形式の数値データを内部状態変数(属性データ)として有するが、さらに、入力データの編集に必要な手続きを、それ自身に固有の機能として持たせることによって、機能部品としてのモジュールと解釈できる。以下では、これをキーワードと称する。入力データ全体もまた、キーワードを構成要素とする順序付集合としてのモジュールと解釈できる。ここでは、属性データの構造について考察を行なう。

キーワードは、その属性データの構造によって図2のように分類できる。これらは、属性データを持つものとそうでないものとに大別されるが、後者は次の2つに分けられる。

- 入力データを階層化するための識別キーワード
- 計算モデルを選択する特性キーワード

3.1 識別キーワード

識別キーワードは、キーワード単体でのデータ構造として見ると、属性データを持たない特性キーワードと同じであるが、それ自身のモジュール属性に役割を有するのではなく、入力データ全体の階層化すなわちカテゴライズを目的とした識別子である。MARCの場合、表1における機能分類の群を固有の識別キーワードによって区切っている。SAPは、動的解析を中心に各種オプションを末尾におき、データ順序の前半部でオプションの項目数が変化しないよう配慮されており、しかも、線形問題であるために計算条件の履歴を必要としない。従って明示的な識別子は必要ない。

識別キーワードによる階層化は、各階層に適合する特性キーワードを判別するためだけではなく、異なる階層において、同じ名称の特性キーワードでありながら役割が変化する場合に必要である。非線形解析における一般化荷重や拘束条件の初期設定と計算過程における条件変更の設定は、この例に該当する。このような位置付けから、識別キーワードを入力データ全体の任意構成要素として配置する本手法のほか、入力データ全体を表すモジュールに、予め階層化構造を属性として持たせることも、1つの妥当な方法である。

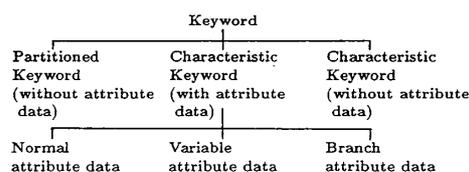


図2 入力データセットのキーワード分類

3.2 特性キーワード

前者の属性データを有する特性キーワードは、その属性データの構造によって次の3種類に分けられる。

- 標準型 (normal)

属性データの個数と順序ならびにその役割が固定。

- 可変型 (variable)

属性データのある部分の個数が任意。

- 分岐型 (branch)

多目的の汎用プログラムに特有の性質。

一般的には、この3種類の合成型として表現できる。

MARCの Title や、SAPの Heading Card は、標準型に該当する。

節点や要素の定義は、可変型の代表的な例である。

MARCは、節点定義に Coordinates を用いており、配列空間の情報と座標値からなる。SAPは、節点定義に Master Control Card による配列空間の指定と Nodal Point Data による座標値の定義を組み合わせ使用する。

多目的の汎用 FEM では、温度や振動、歪などの解析分野に応じて、同じ名称のキーワードでありながら、属性データの構造と意味が異なる。MARCの Control は、収束誤差を規定するが、流体モデル、固体モデルなどの条件に応じて属性データの内容が異なる。この性質は、多形態性 (Polymorphism) として整理できる。また、材料物性値を記述する MARC の Property は、可変型と分岐型の合成型として表現される。

3.3 可変型のための補助モジュール

キーワードの属性データは、内部状態を表す変数と解釈できるが、属性データの中に可変(不定)構造を有する場合があるため、全ての数値データと1対1に対応する状態変数を定義することは困難である。可変構造とは、図1で示す LineVector など、型が固定された行ベクトルの数であると認識されるため、1行毎を基本要素としなければならない。このような可変構造は、解析モデルの要素分割や境界条件の適用範囲を設計する場合などに発生する。

行ベクトルの生成には、2つの方法が考えられる。

1. 行ベクトルを独立した補助モジュールとし、キーワードの1つの属性に、不定型順序集合変数 (OrderedCollection) を定義し、これに行ベクトルのインスタンス (生成された実体) を必要な数だけ追加する。
2. キーワードの属性として、行ベクトルの成分変数と合わせて不定型順序集合変数を定義し、必要

な数だけ行ベクトル成分の数値データを生成して追加する。

ここで、2.の構造は、閉じたモジュールとして評価される。しかし、内部におけるモジュールの意味特性は、難解なものになる。ある行ベクトルの1つの成分数値を指し示すには、状態変数を多価関数として評価する。これは、データ構造の同相性を破壊していると解釈できる。これに対して1.は、キーワードと異なる行ベクトルのインスタンスを追加したものと評価され、すべての数値データが1価性を有する。反面、キーワードの定義において、入力データオプション名の直接的写像ではない、便宜的な補助モジュールを考えなければならない。支援システムを開発する過程において、1価性を有するモジュール間でメッセージ通信を行なうことは、プログラムの流れを簡単にし、難解なアルゴリズムを避けるという意味で評価される。この考察に基づいて、本支援システムの開発は1.の手法を採用している。

MARCでは、オプション名を表すキーワードを含めて4層まで階層化された行ベクトルを考えればよい。SAPでは、2層まで考えれば十分である。この階層構造をn層($n > 0$)まで表現でき、しかも視覚的に効果のある知識表現法として、マルチウインドウ型エディタ(ブラウザ)による作業環境が考えられる。これについては、後述する。

以上の考察から、汎用FEMの入力データを構成するキーワードのデータ構造が明確になった。

4. 経験的知識のモジュール構造

前節で考察した入力データの編集に関する経験的知識は、一般にIF/THEN型規則で記述され、キーワードのモジュールと別に構築される。以下では、これらの規格化されない知識を支援するモジュールを「設計規則」と言う。

3),4)によれば、知識支援の実現において、解析分野の自由度を選択肢により狭めていき、選択条件を満足する事例を引用する方式が採られている。この過程において、選択肢は、設計規則の属性データであり、選択条件項目は、キーワードの組み合わせ実現などの手続きであると認識される。これは、選択肢を状態変数とし、選択条件をメソッド(手続機能の呼称名)とするモジュールにはかならない。赤木ら¹⁰⁾は、船舶設計モデルにおいて、設計パラメータをオブジェクト、設計処理をメソッド、パラメータ間の処理連鎖をメッセージ通信としてとらえており、本考察における選択

肢と選択条件へのモジュール評価は、この考え方と一致する。

選択肢(状態変数)と選択条件項目(メソッド)の関係は、キーワードの属性データ(状態変数)と内部処理(メソッド)のそれとは異なり、選択条件項目の実現が主であって、選択肢は、重要ではあるが補助的な、しかも仮想的なデータである。選択肢の状態量が指し示すものは、実現のための経路特性すなわち動的な過程である。入力データに対して動的な過程を規定し、メッセージ通信によって実行した結果として、新たな入力データが生成される。このことから、設計規則モジュールの果たす役割が、作用演算子に相当するものと認識できる。

5. 支援システムの概要

5.1 基本設計

支援システムは、主として入力データ(AsyData)とそれを構成する部品としてのキーワード群(PartsWord)、それらを組み立てて編集するためのブラウザ(HierBrowser)、ならびにブラウザを介して経験的知識を支援する設計規則(DesignExpert)に分割される。

入力データ、キーワード、設計規則は、何れも機能モジュールであるが、これらをオブジェクトとして実現すると、実体と概念の間に無理がなく、しかもブラウザを含めて統括的にシステムを構築できる。

PartsWordは、キーワードの抽象クラスとして位置付けられる。その意義は、全てのキーワードの重複部分を共通化することである。AsyDataは、キーワード・インスタンスを順序付けて配置するための入れ物である。

AsyDataの編集行為は、ある程度の経験者であれば、ブラウザを介した階層状知識表現の環境下で、キーワードに隠蔽された文法情報を参照することによって、従来の製本マニュアルを参照する場合と比べてはるかに効率的に行われる。しかし、規格化されえない経験的知識を支援することによって、より高度な編集を行うことができる。この実現のために、ユーザは、AsyDataを直接編集する代わりに、DesignExpertへのメッセージ通信により、知識の支援を得る。図3は、これらの関係を模式的に表している。

AsyDataは、入れ物としての機能のほか、部品の持つ情報から総合的に生成される、新たな確定的情報について処理すべきである。1つの構築例として、要素と節点の情報からメッシュ分割状態図を表示する機

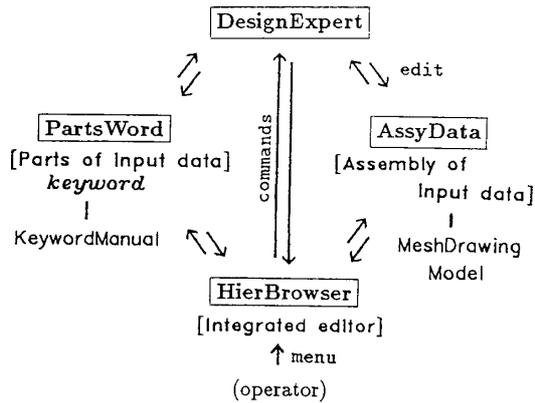


図3 支援システムの役割分担

能を設定している。ただし、作図の手続きについては、要素図形のモジュール (MeshModel) を用意し、これによって処理する。

5.2 設計規則の実現形態

図4は、設計規則に関するクラスの階層形態を示す。解析分野の大分類に従って分岐されたクラスにより、知識の細分化がはかれる。また抽象クラス DesignExpert を用意し、設計基本事項の共用化をはかる。各分野の設計規則オブジェクトは、共通メッセージ basic, boundary, material などによって選択条件の特性を分類できる。また、分野毎に特有のメッセージを設けることもできる。これらに対応したメソッドは、個々の分野に依存し、選択肢の組み合わせ条件によって挙動が変わる。その内容は、IF/THEN 型で実現されるが、規格化された情報を顕在的に含む必要がないため、簡単化される特質がある。

このようなメソッドによってカテゴリ化された IF/THEN 型の設計規則は、2階層のフレーム型表現と解釈することも可能であり、福田ら⁵⁾の指摘するフレ

Object

DesignExpert

BearingStability

Buckling

⋮

ElasticPlasticity

HeatStressCouple

HeatTransfer

⋮

図4 クラス DesignExpert の階層構造

ームベースの利点を生かしながら、キーワードの隠蔽情報と異なる次元で併用できるため、効果的である。

5.3 ブラウザの基本機能とメッセージ構文

ブラウザ (HierBrowser) は、階層的なデータ表現と編集のためのユーザ・インターフェースとして実現され、次の主な機能を提供するよう設計される。

- キーワード・インスタンスの追加、削除、複製。
- キーワード名、属性データ変数名の参照と選択。
- 属性データの編集。
- 入力データファイルの出力 (送信)。
- 外部参照ファイルの読み込み。
- キーワードおよび属性データに関するオンライン・マニュアルの提供。
- キーワード・クラスの参照。
- 設計選択の実行。

これらは、ブラウザの中に専用に登録されたウィンドウ・メニューを介して、所定のメッセージを相手のオブジェクトに送ることによって実行される。各オブジェクト間の通信を考えるに際し、その役割と目的によって共通のメッセージを使うよう規定する。メッセージ (メソッド名) を統一することによって次のような意義がある。

- 個々のキーワードの挙動は異なるが、それらの機能と手続きを概観できるため、新規開発部品の基本仕様が明確になる。
- 駆動側のブラウザに備えるべき基本機能が明確になり、キーワードに対する手続きが目的別に単一化される。

ユーザの編集行為は、ブラウザのウィンドウ・メニューを選択することによって実行できる。このような間接的な編集法は、入力ミスの削減に有効である。主な共通メッセージを表3に示す。

キーワード (PartsWord)、入力データ (AssyData)、ブラウザ (HierBrowser) 間の共通メッセージを分析することによって、これらのオブジェクトとしての枠組みが明らかになる。キーワードとブラウザの模式的なオブジェクト構造を図5に示す。

表3 主な共通メッセージ

メッセージ名	意味・役割
help,minihelp	マニュアル・ウィンドウを開く
display	データコードの出力
inserFile	ファイルの読み込み
add/remove	インスタンスの追加/削除
createLine	対応する行ベクトルの生成

属性変数の定義
 インスタンス変数: " string "

メソッドの定義
 clear
 属性変数 " string " の初期化.
 display
 入力データファイルへの出力 (実行用データの生成).
 help
 ウィンドウマニュアルの検索と表示.
 set : aComment
 属性変数 " string " への値の読み込み.

キーワード TITLE のオブジェクト形態

属性変数の定義 各サブウィンドウ自身, サブウィンドウの選択状態を示すフラグ, 編集対象 (MarcData)

メソッドの定義
 追加, 削除, 複製
 キーワードインスタンスの並べ換え.
 参照, 選択, 値の編集
 マウスクリックによるキーワード名, 属性データ変数名の参照と選択. 属性データの編集.
 出力 (送信), 読み込み
 入力データファイルの出力. 参照ファイルの読み込み.
 マニュアル呼び出し
 キーワード及び属性データに関するマニュアルの提供.
 :

ブラウザ HierBrowser のオブジェクト形態

図5 オブジェクトモデルの形態

6. ブラウザの構成設計

6.1 基本要素の構造

オブジェクト情報のウィンドウ表示と編集作業についての形態は、キーワードのデータ構造から決定されなければならない。キーワードは、表示・選択すべき情報として、3種類あると認識される。

- 1) キーワード自身の名前
- 2) 属性データ (状態変数) の名前
- 3) 属性データ (状態変数) の値

1), 2) は、固有の識別名称であり、選択の対象となる。3) の値は、表示と変更を行なう対象であるが、一般にスカラー量ならびにベクトル量を扱う。3) がスカラー量の場合、1つのキーワード・インスタンスを状態表示して操作するには、2つの選択用ウィンドウ (リストペイン) と1つの編集用ウィンドウ (テキスト

ペイン) が必要である。この3枚の組合せは、属性データの値がベクトル量であるオブジェクトを表現する1基本単位になっている。3) がベクトル量で、その成分がスカラー量である場合、キーワードを表現するのに必要なウィンドウは、2基本単位 (6枚) であり、3) のベクトルの成分がベクトルでその成分がスカラーならば、3基本単位を必要とする。図6は、3基本単位までの属性データの対応関係を示す。このような、基本単位ウィンドウによるキーワードの表現は、キーワードの名前と属性データの変数名を表示するので、その値が何を意味するかというマニュアル情報を視覚的に提供している。この環境形態は、入力値の型規制としても有効である。

6.2 マルチ・ウィンドウの構造

MARCのキーワード Contact¹¹⁾は、4基本単位 (12枚) のウィンドウを必要とする。これは、4階層の行ベクトルによる可変データ構造のキーワードであることによる。このような多層のウィンドウを同時に扱うことは、型規制とマニュアル情報を提供する優位性に反して、操作性は必ずしもよいと言えない。そこで、頻繁に使われる2基本単位までを常設ウィンドウとし、それ以下の階層については、動的にウィンドウを起動する方式を採用する。この動的な部分については、ウィンドウの特性から2つの実現方法がある。1つは、常設ウィンドウと同位置に重ね書きあるいは切替えて使用する方法であり、もう1つは、マウスの指示により、任意位置にウィンドウを開く方法である。前者は、段階的・選択的に各階層のベクトル量を表現するので、直前の選択条件にあった選択肢と対応する階層ウィンドウを取扱うことになり、型と操作範囲の規制に有効と考えられる。これに対して後者は、次の短所を持つ。A) 現実的なスクリーンサイズの制限から、上位階層ウィンドウが下位階層ウィンドウに隠さ

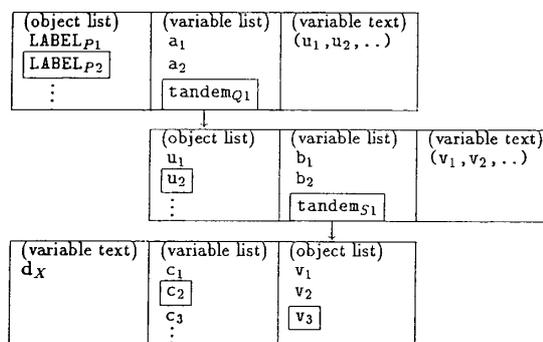


図6 多層型キーワードの検索機構

れる部分が多い。B)開く位置をマウスで指示するので、頻りにウィンドウを切替える場合には手順が多く、適当ではない。これらの考察から、キーワードを表現するマルチ・ウィンドウとして、前者の方式を採用した。

一般に、 n 階層のベクトルを表示するために、 n 基本単位 ($3 \times n$ 枚) のウィンドウを提供すればよい。このようなキーワードの階層表現は、変数名と値を同時に扱えるため、従来の平坦な表現法と比較して、過去の例題を評価してユーザにわかりやすく提示でき、流用できるという特徴を有する。

6.3 ウィンドウ・マニュアル

キーワードの情報は、階層表現によって効率的に提示されるが、意味不明の項目については、特に初心者にとって、明示的な文法と対話するしかなく、このことが編集作業のどの時点で発生するかは極めて様々である。そこで、キーワードや属性データの名前を指定し、直ちにその場でマニュアル情報を引出すことができるようにする。すなわち、ブラウザによって階層化され、指定した範囲での関連マニュアルをウィンドウ表示する方式である。このような「階層化マニュアル」は、必要なとき必要な範囲に絞って提供できるだけでなく、知識の蓄積をも細分化させるため、平坦なマニュアルの構築方法と比べて利用段階における作業性を高める。

マニュアル情報は、各キーワードの属性データと解釈することも可能であるが、むしろ、保守性や拡張性を考慮すると、システムから独立したマニュアル情報の集積を行って外部記憶装置との通信規約をキーワードの手続き属性として設けることによって、機能の分散化をはかることができる。

ウィンドウ・マニュアルの起動は、マウス指定により任意の位置で行なう方式を採った。理由は、その性格上、頻りに開閉操作を行なうものではなく、内容によって、ウィンドウ・サイズを調整したほうが快適な場合があるからである。

6.4 ブラウザの動作環境

MARCのキーワードを扱った支援システムについて述べる。入力データのインスタンス AssyData に対してブラウザを呼出すと、図7のようなウィンドウが開かれる。ウィンドウ(1)には、登録されたキーワード・インスタンスの名前が表示される。マウスを使ってキーワードを指定すると、その属性データの変数名をウィンドウ(2)に表示する。(3)に、(2)で選択した変数の内容(値)が表示される。(4)~(6)は、可変構造

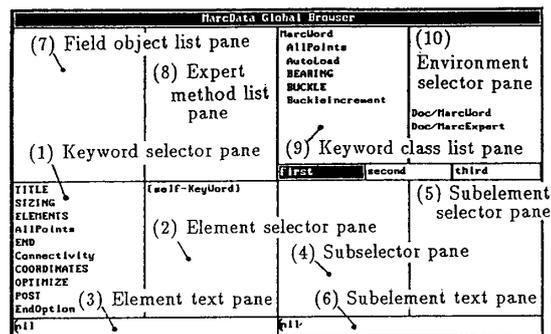


図7 ブラウザ HierBrowser の各枠の名称

型オブジェクトの行ベクトルのためのウィンドウで、ラベル・ウィンドウ first, second, third をマウス指定することによって各階層のウィンドウを交互に切替えることができる。ウィンドウ(9)は、既に登録されているキーワード・クラスの全リストを表す。クラスをマウスで指定し、ウィンドウ(1)において追加メニューを実行することで、(1)にキーワード・インスタンスを生成することができる(図8)。このような「併用検索」は、平坦な単一のエディタによる編集とマニュアルを参照する従来の環境と比べて、選択作業性を高めている。ウィンドウ(7)、(8)は、設計選択に関連して使用される。

図9は、ウィンドウ(9)の Connectivity に対して、メッセージ help を実行し、ディスクファイルのマニュアル情報を表示している。その内容はテキスト形式で、ユーザが自由に追加・修正可能である。

本システムは、作成途中における参照機能を有しており、ウィンドウ(1)におけるメニュー fileShow, meshView によって入力データコードのイメージならびにメッシュ分割図を表示できる。

図10は、弾塑性問題における境界条件のメソッドを

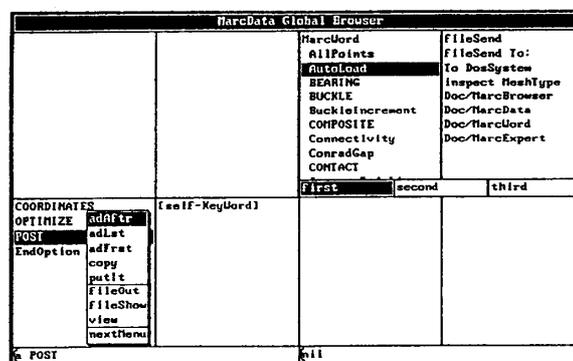


図8 クラスとインスタンスの併用検索

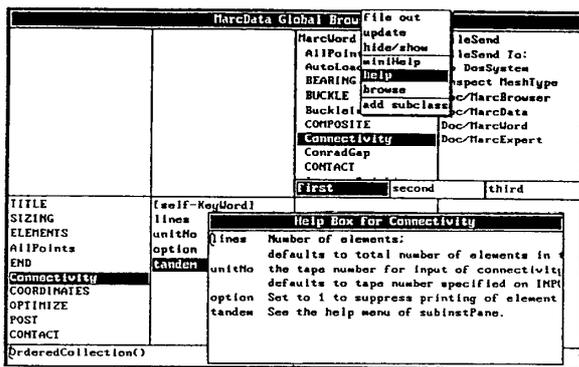


図9 キーワード Connectivity のウィンドウマニュアル

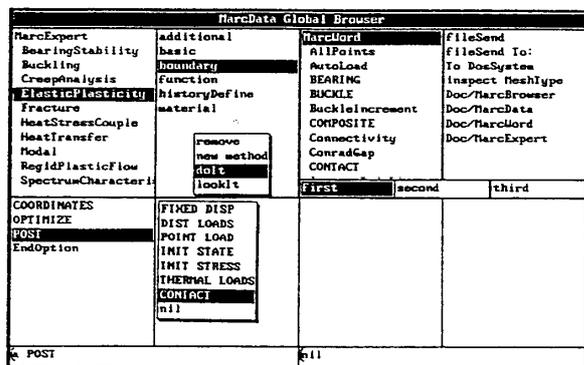


図10 設計規則オブジェクトの生成と作用形態

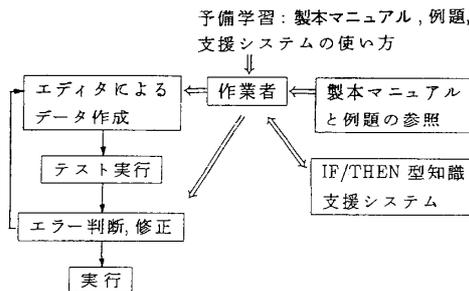
実行し、その結果として関連するキーワードの選択メニューを表示している状態を示す。

7. 本支援手法の効果

汎用 FEM 入力データは、規格化された機能とデータ構造を有するため、継承性や多形態性に基づいて整理することで、拡張性や再利用性において効率の良いプログラム化が実現できる。特に、FEM のラベル付きオプション（キーワード）を基本的なオブジェクトとして扱う本手法は、支援システムの設計におけるモジュール単位として妥当であるのみならず、利用段階の問題である膨大な文法情報をラベル名に隠蔽させ、作業者の負担を軽減させる。また、キーワードに含まれない文法以外の設計知識を明確にできるため、設計規則としてのとりまとめは、従来の IF/THEN 型規則と比べて複雑さが軽減される。

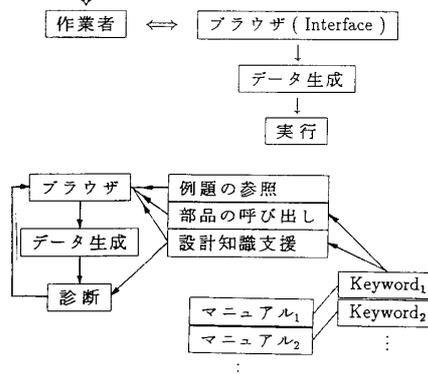
図11は、階層状知識表現のために導入されたブラウザを FEM 入力データの編集に適用する場合と、従来の編集作業との違いを示す。

従来手法(1)では、予備知識として膨大なマニユア



(a) 従来手法による FEM コード設計。

予備学習: Keyword の浅い知識, 例題, 支援システムの使い方



(b) 本手法による FEM コード設計。

図11 編集作業における手順の比較

ルの詳細情報を持つ必要があるとともに、あいまいな文法を常に確認しながら、入力データを編集しなければならないため、十分な設計を行うために大きな時間的負担を要する。これに対して、本支援手法(b)では、キーワード毎にマニュアル情報を管理し、隠蔽しているため、ブラウザのメニューに用意された基本的な共通メッセージを理解し、使うだけで効果的に処理できる。

また、例題を利用する場合、ブラウザを使った支援手法(b)では、キーワード毎に階層的なデータの変数名や値を容易に照合できるため、(a)のように製本マニュアルと対比させてデータを解読する手法と比べて、その利点は明らかである。

8. あとがき

汎用 FEM の解析支援方法に関して、入力データのモジュール構造とその環境特性について考察した結果、以下のことが明らかになった。

1. MARC と SAP を題材にして入力オプションのデータ書式を分析し、その構造的特性を一般化した。

2. 入力オプションを1つのモジュール(キーワード)として取り扱うことで、支援システムの開発はキーワード毎に独立して行える。
3. 階層化マニュアルの実現により、解説情報の効率的な参照が容易である。
4. 内部データ構造を隠蔽した部品としてのキーワードを扱うことによって、また、キーワードが有する固有の手続き機能を共通のメッセージ通信によって操作することによって、個々のキーワードの深い知識に直接関与することなく、支援システムの開発ならびに利用を進めることが可能である。
5. 可変型属性データを持つキーワードは、補助モジュールを導入することによって、内部構造を同相に維持したプログラミングが可能であり、標準(固定)型属性データを持つものとは比べても設計性を損わない。
6. ブラウザにおいて、キーワードのクラスとインスタンスのリストペインを双方に検索する形態は、選択の作業性を高める。
7. 可変型属性データに対応できる多階層のデータ表現法を示した。
8. 設計規則は、疑似的な作用演算子としてのモジュール化である。
9. 階層表現は、参考例題コードの変数の意味・名称と値とを同時に提示できるため、評価法として効果的である。
10. 汎用 FEM のソルバーが内部で備えるべきデータ構造をキーワードによって整理できる。

9. 謝 辞

本研究を進めるにあたり、有益な助言をいただきました九州工業大学小林史典助教授に厚く謝意を表します。

参 考 文 献

- 1) C. A. Brebbia: Finite Element Systems (A Handbook), Computational Mechanics Centre (1982)
- 2) J. S. Benet, R. S. Engelmire: SACON: A Knowledge-Based Consultant for Structural Analysis, MDA-903-77-C-0322, Stanford Univ. (1978)
- 3) 四方, 森丘, 小林, 梶原: 構造解析汎用パッケージのコンサルテーションシステムの開発 (MARC 版の概要), 情報処理学会第36回全国大会講演論文集, 20-1 (1988)
- 4) 春木, 小寺, 青江: 数値解析コンサルティングエキスパートシステム, 電子情報通信学会全国大会講演論文集, 1-138 (1987)
- 5) 福田, 中桐, 大久保: 有限要素解析支援エキスパートシステムの開発, 日本機械学会論文集 (A 編), 56-521, 168 (1990)
- 6) B. J. Cox, 前川監訳: オブジェクト指向のプログラミング, トッパン (1988)
- 7) B. Meyer, 二木監訳: オブジェクト指向入門, アスキー出版 (1990)
- 8) A. Goldberg, D. Robson, 相磯監訳: Smalltalk-80 (言語詳解), オーム社 (1987)
- 9) A. Goldberg, 相磯監訳: Smalltalk-80 (対話型プログラミング環境), オーム社 (1986)
- 10) 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), 50-500, 1017 (1987)
- 11) MARC マニュアル, C3.4-24, 日本 MARC (1988)