構造解析プログラムへのオブジェクト指向の適用

佐賀大学大学院　学生員 ○ 今村　敬
佐賀大学　正会員　荒牧　軍治
佐賀大学　正会員　古賀　浩二

1.はじめに

オブジェクト指向はソフトウェア管理危機に対する解決法として提案されたプログラミング手法のひとつであり、大型ソフトウェアの管理コストが時間とともに増大する問題を解決するために適用されている．この手法はプログラムを必要とする様々な分野で適用され，成功をおさめているが、構造解析プログラムの分野では、他の領域に比べオブジェクト指向を適用する事例が少ない．その理由としては次のような事が考えられる．(1)構造解析等の数値計算プログラムはその歴史が長くソフトウェア資源が多く蓄積されている，(2)問題の性質上、高速計算が求められ，プログラムの最適化が必要な事，などである．どのようなプログラミング手法を用いても同じ解析結果を求めることができるが、プログラミング手法が違えば，問題に対する考え方は大きく違ってくる．オブジェクト指向手法はこれまでの手続き型の手法とは大きく異なるため、プログラム全体の基本設計から変更しなければならなくなる。これがオブジェクト指向の適用事例の少ないひとつの理由と考えられる．本論文では，構造解析プログラムに対して，オブジェクト指向を適用する時の考え方などを示し，従来からの手続き型の手法に比べ、この手法が構造計算プログラムにも有効である事を示す．

2．構造解析とオブジェクト指向プログラミング(OOP)

2.1オブジェクト指向について

従来型手法で構造解析を行う場合，その処理フローを構成するのに，どのような処理（サブルーチン）が必要で，それらをどのように組み合わせるかを検討した．これに対しOOPでは，解析の中にどのようなオブジェクトが存在するか検討し，それらがどのように動作するのか，によって目的とする処理を実現する．このようにプログラミングの対象になっている解析に対して，それぞれの捉え方には大きな違いがある．

オブジェクトは，オブジェクト自身を特徴付ける情報と，オブジェクト自身が関係する処理をひとつにまとめた（カプセル化した）ものである．カプセル化により，オブジェクトの詳細をプログラム全体から隠ぺいする事が可能となった．また，OOPではオブジェクトの抽象化が可能である．私達はある物事をとらえようとする時，全体を大きく把握し，分類する．そして分類されたそれぞれを細分化してゆく．人間の思考でははじめに抽象的に，次第に具体的な表現へと移っていく階層構造をなすトップダウンアプローチである．オブジェクト指向では問題をこのようにとらえ，思考に近い表現をプログラムとする事ができる．それを可能にするのは，継承，オーバーローディング，ポリモーフィズムなどといったオブジェクト指向に特徴的な技術である．

2.2 解析プログラムへの適用

構造解析プログラムが適用されるのはおおよそ次のような場合がある．静的解析（線形，材料非線形，幾何学的非線形），動的解析（線形，材料非線形，幾何学的非線形）などである．これらがプログラムとなった時，その解析フローは用いるアルゴリズム等により違いは生じる．例として構造物の静的解析を考える．計算の流れはおおよそ図1のようになる．解析の基本となる式は次のようになる．

$$[K]\{x\}=\{F\}\ ,\ [K]=\sum^{n}K^{e}\ ,\ [K^{e}]=\int_{V^{e}}B^{t}DB\ dV^{e}$$

これらはそれぞれ全体剛性方程式，全体剛性行列，要素剛性行列である．要素剛性行列をみれば，この表現は要素の種類によらず同じ表現である．また，全体方程式も線形，非線形などの違いによらず，同じものである．構造解析プログラムでは，解析上必要な数式は共通しており，これを軸に階層構造を形成する事が可能である．有限要素は様々なものが提案されているが，それらは要素剛性行列K^{e}という表現を軸に抽象化したオブジェクトから派生したものと考えられ，階層構造をなす（図2）．構造解析プログラムへのオブジェクト指向の適用は数学的表現をもとに，どのようなオブジェクトが含まれているのか，検討する事になる．構造解析の中にはさきに挙げたクラスのほかにも，オブジェクトとして扱う事ができるものとして，節点，材料にかかわる情報などが考えられる．

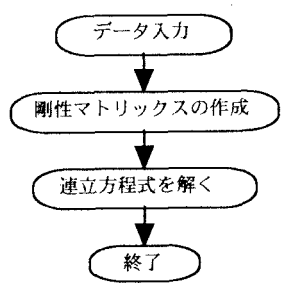

図1．解析フロー

2.3 従来型プログラムとの比較

・行列演算　適切な演算子の設定により，プログラム上で数式そのままを表記でき，直感的表現が可能となる(図3)．解析プログラムでは行列演算の最適化は重要であるが，その作業を解析プログラムとは独立に行う事ができる．データ構造の変更も容易で，行列の要素を数値だけで無く，関数として扱う事も可能である．従来型手法では，プログラムの他の部分に影響を与える事なくデータ構造の変更を行う事は困難である．

・要素　要素をオブジェクトとし，解析部分からは具体的な要素にアクセスする事なく，要素オブジェクトの上位階層の抽象要素へのアクセスする形式をとる．従来型手法では解析部分から，具体的な要素へのアクセスが必要であった．このため，解析部分，特に全体マトリックス作成の部分では，要素に関する詳細な情報，形状，節点自由度などという情報があらかじめ分かっている必要があった．そのため，機能追加として新たな要素をプログラムに追加する際，プログラム全体の見直しが必要となっていた．オブジェクト指向を適用した場合，要素に関する作業は，要素自身がその内部の情報によって行うことができる(図4)．また，要素に非線形性を考慮する場合，従来型手法では解析部分で要素の監視を行う必要があったが，要素自身の振る舞いにその責任を持たせる事ができ，プログラムが複雑な構造となる事をさける事ができる．

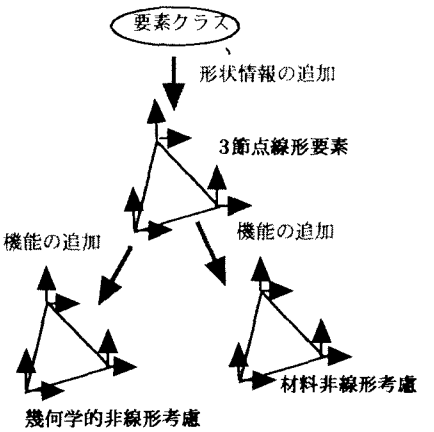

図2.要素の階層

式　$x = K^{-1} f$　・・・（1）

```
//式（1）の計算プログラム
matrix K;
vector x,f;
-----------------
x=K^(-1)*f;
-----------------
```

図3 プログラムの例

解析実行部　解析実行部は適用する計算法に応じてオブジェクト化し，階層化が可能である．個々のアルゴリズムにかかわる部分を除いて，各オブジェクトは継承を適用して，ほとんどをプログラムを共有する事ができる．これにより，プログラムの開発は，計算手法にのみ集中する事ができ，効率的な開発が可能となった

2.4 オブジェクト指向適用時の注意

OOPの適用により，プログラム全体の記述が簡単なものができた．また，プログラム各部の再利用が容易で，効率的なプログラムの作成環境を提供できる事がわかった．しかしながら，その作業開始の段階では従来型手法にくらべ，プログラム作成の進み具合は遅いものとなった．これは，オブジェクト指向を適用する場合には，対象となる問題について慎重な検討が必要であり，この段階での設計がうまくいかない場合，従来型手法にくらべ，プログラムに関する作業が混乱したものとなる可能性がある．

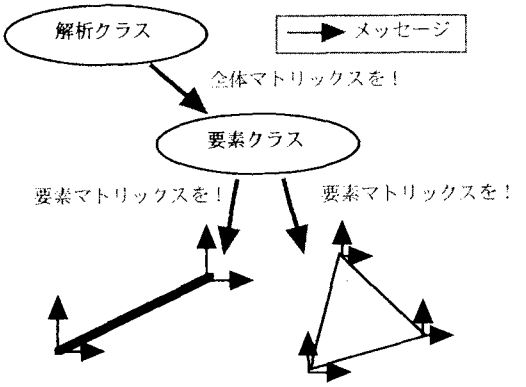

図4.　メッセージの例

※開発したオブジェクト指向構造解析プログラムを利用して、地震時動的解析について、本発表会において同じ研究室の学生が発表している．

参考文献

1)B.ストラウストラップ: プログラミング言語C++　第3版，アジソン・ウェスレイ・パブリッシャーズ・ジャパン

2)S.マイヤーズ : Effective C++　改訂2版，アスキー

3)J.ラコス : 大規模C++ソフトウェアデザイン，ピアソン・エデュケーション

4)矢川元基，関東康祐 : オブジェクト指向計算力学入門，培風館