

早稲田大学 理工学部 正員 宮原 玄

本文はプログラム・システム開発技法の立場から、過去に行った地盤・基礎・構造系の地震動に対する相互作用解析プログラム・システム開発も反省、考察し、問題点を整理した上で、IBMが推奨しているIPT (Improved Programming Technologies) の、旧システムを包括した、新システム開発への適用について述べたものである。

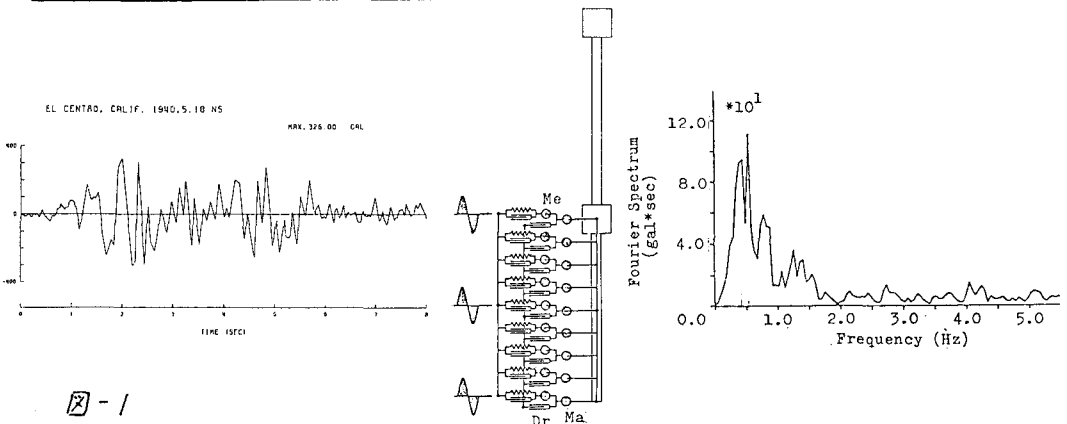
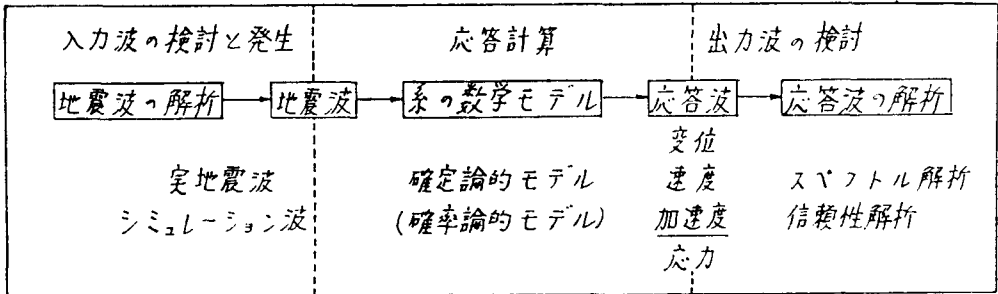


図-1

プログラムに関する問題点：これまでに、図-1に示した手順に従って、地盤・基礎・構造系の相互作用解析プログラムの開発を行ってきた。一般に、プログラムの開発には、多数の人員の努力、長い期間、更に多額の費用が投入される。しかるに、開発されたまわりのプログラムは、その都度設定した目的に合致したものであるにせよ、その目的から少し離れた目的に適用しようと試みる場合に支障をきたし、プログラムを新規に開発し直さねばならぬことが多い。プログラム内の一ヶ所の小さな変更の影響が、そのプログラム内の他の箇所波及して行くことはしばしばある。新たな目的のために、元のプログラムの手直しを行う場合でも、それが作動することを最重要条件として開発されるため、他の目的をも想定した拡張性を持った保守の行い易いプログラムでなく、手直しには困難が伴う。これまでに開発してきたプログラムは、これらの点で、「ソフトウェア」の体裁をとっているものの、新しい目的に柔軟に対応できない「ハードウェア」であった。

また、プログラムの開発が個人の好みに任されることが多く、占有記憶容量が少く、実行速度の速いプログラムが良いとされてきたため、開発者にしか理解できないプログラムになってしまうことも多い。電算機の発達に伴って、記憶装置が廉価になり、演算速度が速くなると、ますます、プログラムの拡張および保守が開発者によってもなされるとは限らないから、誰にでも容易に理解できるプログラムの開発に精力を注ぐ必要がある。この点もプログラムの付属文書の不完全さとともに反省しなければならない。

データに関する問題は：本システムで扱うデータは入力波に関するデータ、系の物理的性質に関するデータおよびプログラムで計算される各種の応答からなる出力波に関するデータに大別される。入力波に関するデータのうち、シミュレーション波は一つの特性のもとで多数の波を発生させる必要があり、必要な度に発生させることは無駄であり、時間の刻み幅も小さく設定されるので、莫大な記憶容量と長期間の保存を要する。また地震波を扱う場合には、それらの記録のフォーマットを統一し、入力波用のモジュールを整理する必要がある。系の物理的性質に関するデータは問題毎に異なるので長期間の保存を要しない。ただし、各種の構成要素毎に異なるフォーマットを用いることは複雑であり間違いの原因になるので統一する必要がある。出力波に関するデータを得るには多くの時間を要し、その量は系の自由度の数、着目する応答の数、現象の継続時間などによって決まるが一般には極めて多い。しかし、それらのすべてのデータをプログラムによって出力させることは適当でない。従って、その問題の解析が完了するまでプログラム内に留め、必要に応じてそれらを取り出し、付加的な解析を実行した後に取出せる様に配慮する必要がある。

問題集の整理：以上を要約すれば、労力的、時間的、費用的な無駄を省くために、品質（信頼性、保守性、拡張性）の良いプログラム・システムを開発し、それらを蓄積すると共に、そのプログラムから利用しやすいデータを準備することに留意すれば良いことになる。しかし、この種の問題は技術計算のみならず管理業務に関するプログラム・システムでも深刻な問題となっており、ハードウェアの発達に比較して立遅れているソフトウェアの開発技法がソフトウェアの危機を招来していると指摘されている。

問題集の解決策：この危機を乗り切るために推奨されているIPTを新プログラム・システムの開発に適用している。管理業務用プログラム・システム開発のために考案されたIPTは技術計算に向いていない点もある。それらの点はその考え方を採用した。それらを整理すると次のようになる。

(1) プログラム・システムに期待する機能を明確化し、処理方式の調査、研究、選択を行う。

(2) 充分な時間をかけて設計を行う。

i) 細分化、単純化、機能の均一化をしたモジュールによって、プログラムを階層構造化する。

ii) 各モジュール間のインターフェースを明確に定義する。

iii) トップ・ダウン方式を採用する。

iv) ストラクチャード・コーディングを採用する。すなわち、連続、反復、判断の機能によって、モジュールを標準化するとともに、スイッチの機能によって、モジュールに機能増設の機能を持たせる。

(3) トップ・ダウン プログラミングを採用する。すなわち、各モジュールのコーディングとテストをトップ・ダウン方式で行う。これによれば、テスト・ドライバが不用となり、各モジュール間のインターフェースの検証が各段階で完了していくために、最末端のモジュールのチェックが終了した時にプログラムシステムが完成することになり、開発の生産性を向上させることができる。

(4) IPTに含まれているHIPO、プログラム記述言語の考え方を採用して、システム設計の段階で各モジュールに関して、機能、処理手法、変数名の説明などを文書化する。

ここに示した技法は息の長い仕事であるが、プログラム・システム開発に長期的効果をもたらすと考えている。

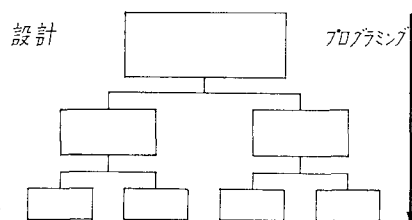


図-2 トップ・ダウン方式

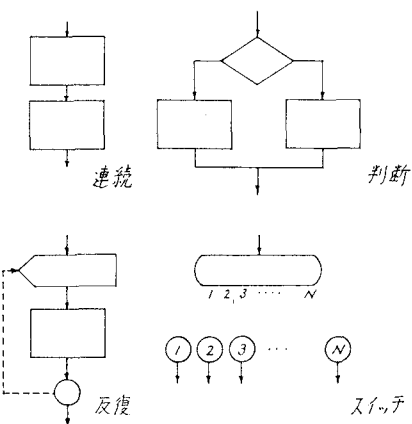


図-3 ストラクチャード・コーディング