

対話形式プログラムの開発方法に関する一考察

株式会社 大林組 正会員 ○浜 嶋 鉦一郎
中 田 雅 治
正会員 藤 井 義 明

1. ま え が き

最近、土木分野においても対話形式による設計方法の必要性が認識され、各種入出力装置を用い、迅速な入力を可能にしたり、より明解に計算結果を表示する対話形式プログラムの開発および利用が盛んに進められている。しかしながら、こうした対話形式プログラムも、プログラム開発者の経験や技術レベルにより、プログラムの質や操作方法に相違が生じ、利用者側の混乱を招く場合もあり、必ずしも、開発効果が十分発揮されていないようである。

当社では、数年前から対話形式プログラムの本格的な開発を始めた。今後も多くのプログラムを開発する予定だが、プログラムの増加に伴ない、上記の問題の発生を防ぐことが重要な課題となって来た。一方、対話形式プログラムの開発工数は、バッチ形式より相当大きくなるので、プログラム製造の生産性向上を図らねばならない。

本文では、対話形式プログラム開発の問題点を具体的に述べた後、図-1で示されるプログラム開発の流れの中で、問題解決のための一つの方法として標準化したサポート・プログラムを利用した対話形式プログラムの開発について述べる。

サポート・プログラムは、多くのプログラムを実用化する中で、対話方法、表示方法、操作方法等、対話プログラムに必要な機能のよりよい形式を模索した結果を、プログラム開発の支援プログラムとして体系化したものである。

2. 対話形式プログラム開発における問題点

設計者がコンピュータを道具として使いこなすためには、対話形式プログラムが希求されることは、各分野で認識されている。しかし、この種のプログラム開発には、新たな問題点が発生している。以下に、当社における対話形式プログラムの開発およびその利用の現状から今後の問題点を具体的に述べる。

(1) 各対話形式プログラムにより表示方法および

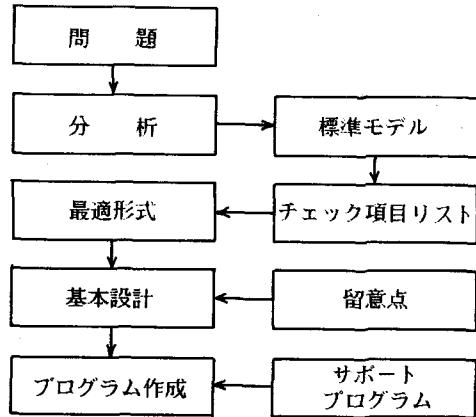


図-1 プログラム開発の流れ

操作方法が異なる。これは、プログラム内容（入力データの量および内容、計算処理内容、出力内容）が異なること、また開発者個人の経験、知識、個性が異なること、さらに利用対象者の知識レベルの想定が異なることによるものである。そのため、プログラム毎に操作方法を覚える必要があり、それから誤操作を生む要因ともなり、生産性の向上に限界を来たしている。

(2) 対話形式の最大の利点は、計算条件の変更により、様々な計算処理の試行錯誤を行えることにある。バッチ処理プログラムをTSSで実行可能となるように変換したものも、対話プログラムと言えるが、処理効率が悪い場合が多い。この種のものは簡単に作成できるので比較的多いが、もっと効率の高いプログラムを作成することが必要である。

(3) データ保存が重要な処理となり、プログラム開発時の留意点となる。さらに、使用者が安全にデータを保有できる支援システムを考慮しないと操作上の失敗時に大きな手戻りを起こし、さらにコンピュータへの不信感を抱かせることにもなるので、安全性の高いプログラムを開発しなければならない。

3. 対話形式の標準モデル

プログラム開発に先立って、対話形式プログラムにはどんなタイプがあるかを知ることは、システム構築に大いに参考となる。以下に、入力主体型、計算主体型、バッチ型を説明している。これ以外に、入力と計算の両方を主体とするものもあるが、簡単な処理については都合がよいが、作業量の多いものは、むしろ分けた方が効率的であると考え省略している。

(1) 入力主体型プログラム

他の関連プログラムで用いるデータの作成を目的とするもの、あるいは、座標読取を主体とした簡単な計算処理で目的を達成するものである。データ生成の生産性を高めるためにはマニュアルが不要となるよう考慮する必要がある。入力データは図形表示によりチェックできるため信頼性が高く、表示画面のハードコピーを成果品として使用するなどの方法で生産性が向上できる。

バッチプログラムが存在するものは、データ形式を同一とし、両用を可能とすることが多い。

(2) 計算主体型プログラム

外部に保有した入力データを用いて、試行錯誤を主体とした計算を行うものである。対話処理のための命令は、座標読取装置、キーボード、図形表示画面からのいずれからも可能で、命令されたことを計算実行し、その結果を数値表示および図形表示するもので、結果に対する判断が容易となり効率よく処理できる。

成果品は、画面のハードコピー以外に、バッチプログラムと同様な計算結果をファイルに出力し、プロッター図も、作図用ファイルに出力可能である。また、外部データの内容の更新も可能である。

(3) バッチ型プログラム

既存のプログラムをTSSで実行可能となるように若干の変換をしたものであり、基本的に対話機能を持っていない。TSSではバッチ処理より優先的に処理されるので、入出力は支援コマンド(後述)を用いて管理し、計算だけ行うものである。入力データの変更、計算処理、結果のチェックが別々のプログラムとなるため、計算主体型と比較して操作手順が多くなること、さらに、不要な計算も同時に行われることとなり、無駄な面が大きい。しかし、既

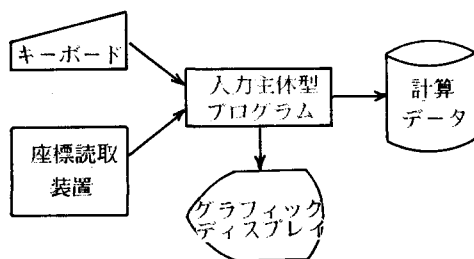


図-2 入力主体型プログラム

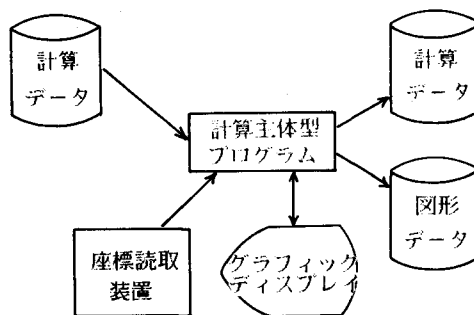


図-3 計算主体型プログラム

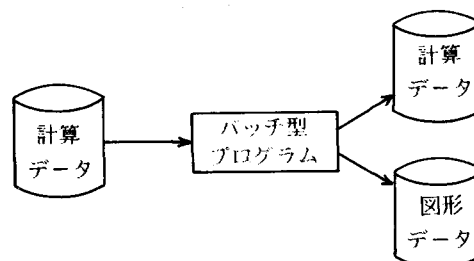


図-4 バッチ型プログラム

にチェックされたデータを用いて自動計算する場合は効率的である。

4. 対話プログラム支援コマンド

TSSのもとでは、一般的に計算結果を直接プリント出力やプロットテープへの出力が不可能である。そこで、一時的に計算結果を外部に出力する必要がある。もとより、ファイルからの出力変換は、バッチ処理であるが、バッチジョブ実行用のJCLに必要なパラメータを対話形式で入力し、JCLの作成と実行を自動的に行うコマンドを用意している。

- (1) プリント出力コマンド—PRINT
- (2) 図形データ表示コマンド—PLOTGA
- (3) プロッター図作成コマンド—PLOTTAPE

5. 基本設計における内容分析と留意点

基本設計段階において、開発プログラムに適した対話形式を設定することは、重要な作業である。対話形式は、入力データ、計算処理、結果の出力の各内容をチェックすることにより、選定する必要がある。表-1のチェック項目リストでは、チェック項目と各標準モデルの適合性を示している。これだけでは完全ではないが、客観的に対話形式が決定されることの意義は大きい。

つぎに、対話形式プログラムのシステム設計における留意点を述べる。

(1) 実使用での対話作業に必要な時間を想定して、実行中断の有無、頻度を把握する、そして、迅速に中断できるような機能構造とする。

(2) データ作成に要する時間が非常に長いものはマシントラブルによるデータ消失等を考慮して、データを適時外部へ保存できる機能を用意すること。

(3) 操作ミス、実行ミスは必ず発生するという前提で、簡単な操作により前段階へもどることが可能なシステムとすること。特に、データ消去や置換処理では、操作の確認処置を行い、大きな手戻りとなるぬようにすることが重要である。

(4) より良いシステムとするには、図形表示機能を重点的に利用し設計することである。不要な機能は問題外であるが、計算結果の図化は可能な限り取り入れるべきである。

6. 標準化したサポート・プログラムによる開発

(1) 利用方法

図-8にこのサポート・プログラムのシステム構成を示す。これには対話プログラムとして必要な機能がほとんど網羅されており、システム設計の基本となっている。新規プログラムの開発は、このサポート・プログラムをベースにし個々のプログラムを作成する。リスト-1に示すように、ソースリストはカナによるコメントが挿入されて明解である。

このサポート・プログラムの中で、標準化した機能は、以下のとおりである。

- ① 全体システムの階層構造
- ② 選択メニュー画面での操作方法
- ③ デジタイザーへの図面セット
- ④ 図形表示画面の図形拡大機能



図-5 プリント出力コマンド



図-6 図形データ表示コマンド



図-7 プロッター図作成コマンド

	入力 主体型	計算 主体型	バッチ 型
1. 入力データ			
(1) 保存データファイルがない	○	△	×
(2) 保存データファイルがある			
・ 変更する必要がない	○	○	○
・ 変更が必要がある	○	×	×
- 多い	○	○	×
- 少ない	○	○	×
(3) 対話入力データがない	×	×	○
(4) 対話入力データがある			
・ 固定条件データ	----	多い	○
・	----	少ない	○
・ 変更条件データ	----	多い	○
・	----	少ない	○
・ 座標データが多い			○
・ 入力データを図形表示する			○
2. 計算処理			
(1) 多様な機能をもつ	×	○	○
(2) 自動計算機能をもつ			
- 全部	×	×	○
- 一部	×	○	×
(3) 試行錯誤による計画を行う	×	○	×
(4) 計算応答時間	----		
長い	×	△	○
短い		○	○
(5) 結果の確認	----		
数値表示		○	○
図形表示		○	○
(6) データを新規作成する		○	○
3. 計算結果の出力			
(1) プリント・ファイルの作成	○	○	○
(2) プロッター・ファイルの作成	○	○	○
(3) 図形画面のハードコピー	○	○	×
(4) 文字画面のハードコピー	○	○	×
(5) 対話入力データの保存	○	○	×

表-1 チェック項目リスト

⑤ 入力データ画面レイアウトおよび入力操作方法

- ⑥ 記憶データの数値表示機能
⑦ デジタイザでの座標読取方法
⑧ 図形表示画面での座標読取方法

(2) 利用効果

- ① 開発工数の大幅減少
② プログラムの対話形式の統一、操作方法の統一
③ プログラムの質の向上
④ プログラミング技術の早期習得
⑤ メインテナンスの容易性確保

(3) 既存プログラムの機能一覧

標準化したサポート・プログラムに含まれている機能がこれまでに開発されたプログラムの中でどれだけ用いられているかを表-2でチェックした。

ほとんどのプログラムが大半の機能を取り入れており、この面から、サポート・プログラムを使用することにより、プログラム開発時の省力化が期待できる。

7. あとがき

対話形式プログラム開発の生産性を向上させるために、対話部分を自動的にコーディングするような手段も開発されているが、著者らは、プログラムの質も含めたプログラム開発の生産性向上を目指し、過去の実績をもとに作業分析を行い、その結果、標準化したサポート・プログラムを整備し実用化を試みた。これにより、システム設計からプログラム製造に至るまでの全体工期を縮小させる効果は大きいと確信している。

```

C      **** パターンセツト O PPGH(ワフ-ニ Q-T) スル
C      CALL GSLPG('GAAPAP','1')
C
C      **** ウィンドウ タイプ (グラフィックディスプレイエカクハイオシヨクハイトスル)
C      CALL GASWIN(WINDOW,1)
C
C      **** データゼロ クリフ
C      CALL ZEROCL
C
1000
C      **** ショクメニューノヒヨウシトヒンダクハナツコ
C      CALL MENU00
C
C      CALL OPTION(LOPT)
C
C      GO TO (10,20,30,40,50,60,70,80,90,100),LOPT
C
C
C      GO TO 1000
C
10
C      **** データインタツク
C      CALL INPUT
C
C      GO TO 1000
C
20
C      **** スケッチノセツト
C      CALL ZUMENS
C
C      GO TO 1000
C
30
C      **** グラフィックディスプレイニヨクワスシ
C      CALL MENU10
C
C      GO TO 1000

```

リスト-1 ソースリストの一部

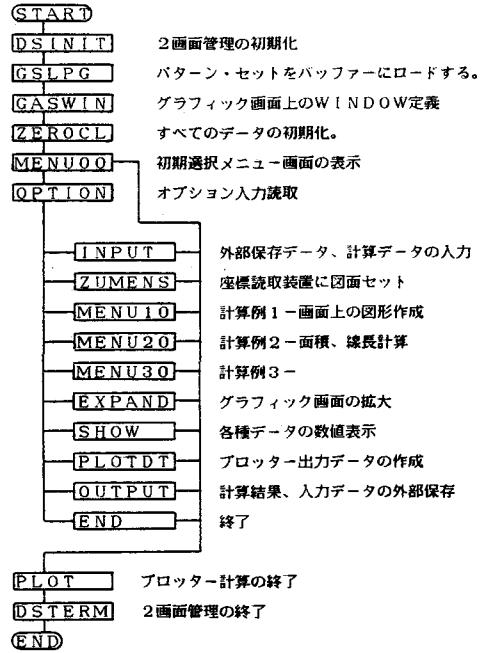


図-8 サポート・プログラムのシステム構成

対話機能	プログラム名									
	MESHDT2	MESHCCC	LANDPLAN	ROADPLAN	PROCHECK	MANHOLE	MENSEKI	SEKISAN		
1. 初期選択メニュー画面表示	○	○	○	○	○	○	○	○		
2. データの初期化	○	○	○	○	○	○	○	○		
3. 外部保存データ入力	○	○	○	○	○	○	○	○		
4. 座標読取装置に図面セツト	○		○	○			○	○		
5. 対話入力処理	○	○					○	○		
6. 計算条件の試行錯誤			○	○	○	○				
7. 計算結果の数値表示	○	◎	○	○	○	○	○	○		
8. 計算結果の図形表示	○	◎	◎	○	○	○	◎	◎		
9. 画面上的図形拡大		○	○	○	○	○		○		
10. 計算データの確認表示		○	○	○	○	○				
11. プロッター出力データ作成		◎	◎				◎			
12. 計算結果の外部出力	◎		◎				◎			
13. 計算データの外部出力		◎	○	○	◎	◎		○		
標準モデルのタイプ	人	計	計	計	計	計	人	人		

表-2 既存プログラム機能一覧表

参考文献

- 1) 浜嶋鉦一郎：対話形式による宅地造成計画システム、第20回IBMユーザーシンポジウム論文集、PP.79~106