

「目で見る計画」の手法—GERTの実用化

(1) そのねらいと図式表現法

石堂 一成

1. はじめに

GERT (ガート, Graphical Evaluation and Review Technique) は, PERT の拡張と考えられる面と, 確率的システムのシグナル・フロー・グラフの拡張とみられる面とを併せもった計画手法であり, 筆者らも1973年頃からその実用化に取り組んできた。しかし, 提唱者 Pritsker [4], [5] らによる図式表現法は複雑であり, そのままでは実用化に難点もあったので, 筆者らは図式表現法の整理・拡張を行ない, さらに独自に汎用シミュレータも開発し, EASYGERT (別名 MHI-GERT) と称して実用化した [1], [2]。その後も改良・拡張を続けて現在に至り, 大は大規模プロジェクトの計画から小は日常の設計計画業務に至るまで幅広く活用して成果をあげている [3]。

本稿は2回に分ける。今回は, EASYGERT の狙い, 図式表現法を説明し, 汎用シミュレータ, 利用形態および実用化の状況を次回に述べる。これが少しでも読者の方々のお役に立つことを願っている。

2. EASYGERT の狙い

2.1 実用的な計画手法

どんなに立派な計画手法でも, 実用的でなければ現場では役に立たない。人間が活動の主体であるような計画の手法が実用的であるためには, 次の3つの条件が絶対に必要である。

- (1) 目で見えてわかること (一目瞭然)
- (2) 現実を正確に反映し得ること (科学性)
- (3) 計算のプロセスが単純明快であること (説得性)

これらの条件を満たせば,

- (4) 合理的な意思決定が容易となり
- (5) どんな人でも納得して計画を実行に移すことができる

ようになる。つまり, 現場で役に立つ。

思惑どおりに事が運ぶとは限らないのが世の中の現実というものである。必勝を期して入学試験を受けても, 必ず何割かの者が不合格となるのは, 毎春見られる光景である。そこで自殺するのは, 誰もが陥りやすい甘い思惑の破綻の1つの結果である。私たちの望むものは, 転んでもタダでは起きないしたたかな計画であり, それを生みだす逞しい計画手法である。そして, それこそ, EASYGERT の主眼にはかならない。

2.2 従来の手法との相違

どのような計画においても, それが主体的な計画であるかぎり, なんらかの達成目標をもっている。目標を達成するための計画を, 規模の大小にかかわらず, 本稿ではプロジェクトと呼ぶことにする。

プロジェクトを成功させるためには, ふつういくつかのチェックポイントを設け, 作業の進捗状況を確認しながらプロジェクトを進めていく必要がある。そのとき, 現実のプロジェクトでは, たとえば新部品が性能試験で不合格というような事態が発生し得る。その結果, 手もどり・繰返しとなる場合もあるだろうし, 既存の別部品で間に合わせる場合もあるかもしれない。

また, 現実のプロジェクトの立案段階では, 必ずいくつかの代替案が存在する。個々の作業項目に対する代替案もあるだろうし, それらの組合せとしての全体レベルの代替案もあり得る。

このような

- (1) 作業の進捗結果に依存する対応策の変化
- (2) 手もどり・繰返しの存在
- (3) 意思決定により選択の可能な複数の代替案

などの現実の問題を, 従来の手法で, 正確に (実用的な計画手法の必要条件(2)), しかも目で見えてわかるように (同じく条件(1)), 表現することができたであろうか。残念ながら, それは困難であった。

GERTは, これらの現実の問題のうちの(1)および(2)の解決を旨として開発されたものであった [4], [5]。

いしどう かずしげ 三菱重工業㈱

表 1 従来のGERTとの相違点

ノードの種類	従来のGERTのノードを4種類に整理し、新しい種類を1種類(意思決定ノード)追加している。
習熟性	手もどりによる作業の繰返しにともなう習熟性(所要時間・費用・人員等の変化)を扱う。
対象要因	所要時間・費用・人員(または機械台数)について総合的な分析ができる。
シミュレータ	入力が容易で、現実の作業の処理に即応したシミュレーションを行なっており、出力もわかりやすい。
代替案の比較	個々の作業項目レベルの代替案およびそれらの組合せとしての全体レベルの代替案の比較が容易である。

これは、たしかに大きな進歩であった。筆者らは詳細な調査を行ない、現実のプロジェクトへの適用化をはかった。しかし、従来のGERTでは、(2)を考慮して繰返ししの図式表現を許しているにもかかわらず、それにもなう習熟性の概念が欠落していた。代替案の取扱いもGERT以前の手法と変らないばかりか、現実のプロジェクトに適用しようとするといくつかの欠陥があった。これらを1つ1つ克服していくと、結果として従来のGERTとは表1のような相違点をもつ1つの実用的な道具

ができあがった。筆者らは、これを、使いやすいGER Tという意味でEASYGERTと呼ぶことにした。

2.3 定性的利用段階と定量的利用段階

EASYGERT の利用には、定性的段階と定量的段階とがある。各段階での利用手順の詳細は後述することとして、ここでは各段階での利用の狙いを図1に示す。なお、定性的利用だけであれば、必ずしも電算機を使用しなくてもかまわないが、できれば使用するほうが便利である。

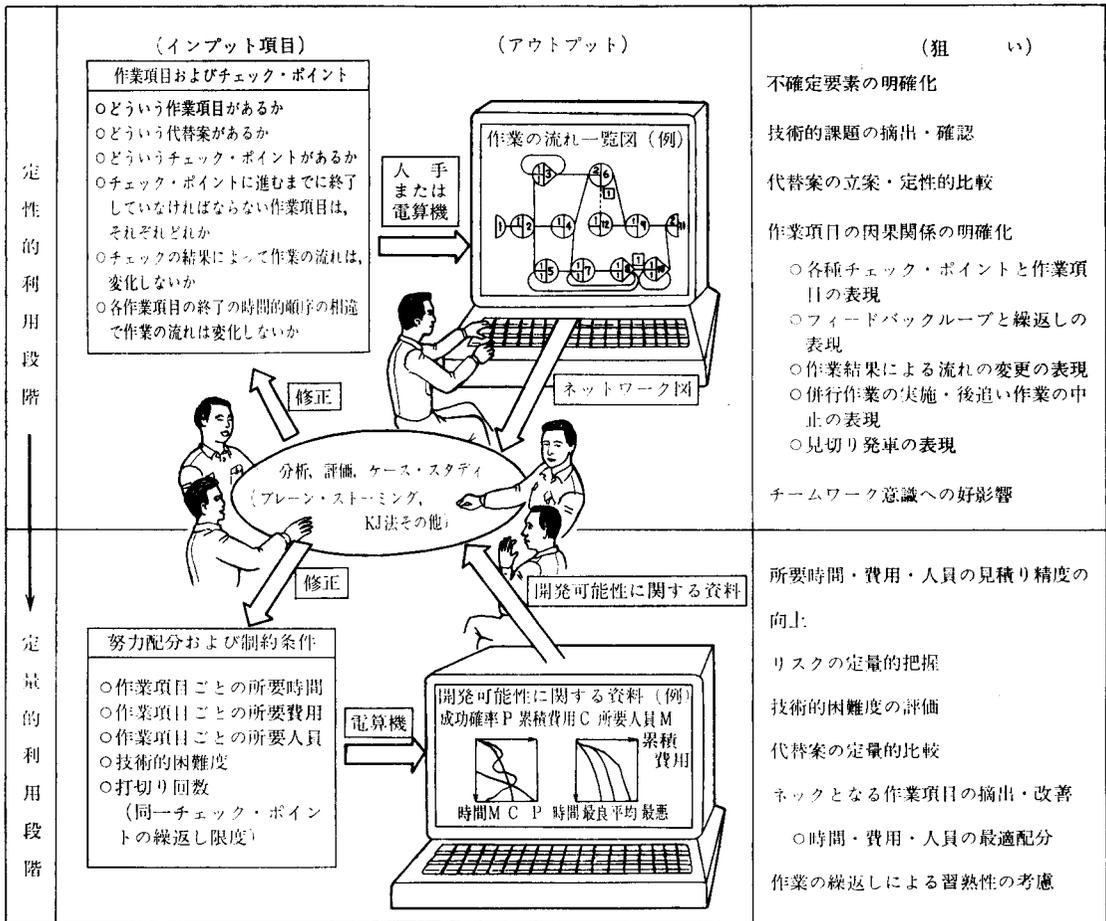


図 1 EASYGERT の利用の狙い

表 2 ノードの種類

種類	図式表現法	意味
終了ノード		最終到達ポイント。
開始ノード		スタート・ポイント。
確定ノード		このチェックポイントを通過すると、その直後の作業項目のすべてが開始される。
確率ノード		このチェックポイントを通過すると、その直後の作業項目のうち1つだけが開始される。どれになるか、あらかじめ知ることができない。
意思決定ノード		このチェックポイントでは、その直後の作業項目のうちどの1つを選ぶかが人間の意思によって決定される。どれにするか、あらかじめ決められることができる。

3. EASYGERT の図式表現法

まず、EASYGERT の図式表現法を簡単に説明する。

3.1 ノード

プロジェクトを構成する作業項目およびチェックポイントをそれぞれアクティビティおよびノードと呼ぶ。

(1) ノードの種類

ノードには、表 2 に示すような 5 種類がある。

(2) ノードの実現 (リリース数, 再リリース数)

チェックポイントでのチェックが完了し、後続作業を開始できる状態になることをノードの実現と呼ぶ。ノードの実現のためには、そのノード直前の作業項目のうちの完了したものの個数がリリース数と等しくなることが必要である。これを利用して、図 2 のような AND や OR の関係を表現できる。リリース数はノード内の左上部分に記入する。

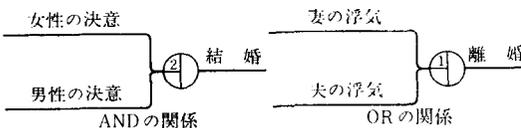


図 2 リリース数の例

リリース数はそのノードの 1 回目の実現に関するものであるが、再リリース数はそのノードの 2 回目以降の実現に関して同様の役割を果たすものである。その際、そのノードの前回の実現以降に完了した項目数と比較する。再リリース数はノードの左下部分に記入する(図 3)。



図 3 再リリース数の例

(3) 確率的分岐

確率ノードを使って確率的分岐を表現する。図 4 の例

のように、後続のアクティビティのうちどれか 1 つが必ず起こる。

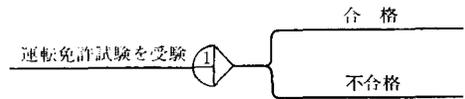


図 4 確率的分岐

確率的分岐に関して現実には同一作業項目の繰返しが起こることがある。図式表現法そのものとしては、確率的分岐とは無関係に繰返しが存在してもかまわない(図 5)。

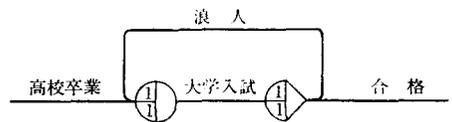


図 5 繰返し

(4) 意思決定による分岐

意思決定ノードを使って、意思決定による分岐を表現する(図 6)。

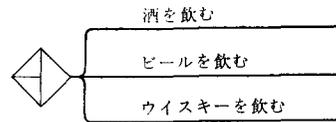


図 6 意思決定による分岐

(5) ノード番号

各ノードを識別するためにノード番号を自然数で与える。ノード番号は、図 8 のようにノードの右半分(終了ノードでは右横)に記入する。定性的利用段階では必ずしも記入しなくてもよい。

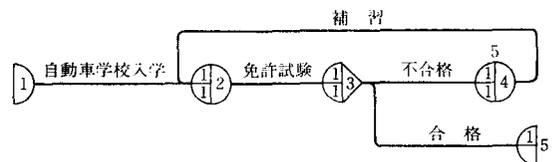


図 7 プロジェクトの打ち切り (5 回試験に落ちるとあきらめる例)

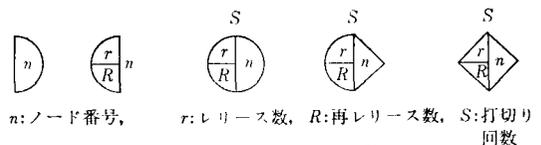


図 8 ノードの図式表現法

(6) 打ち回数

チェックポイントの繰返しに限度があって限度に達するとプロジェクト全体を中止するという場合を、打ち

り回数によって表現する。打ち切り回数は、図8のようにノードの真上に記入する。

3.2 アクティビティ

(1) アクティビティの属性

アクティビティには、必要に応じて属性を与えることができる。属性には、アクティビティ名称、生起確率、所要時間、所要費用、所要人員または所要機械台数、識別番号、モディフィケーション番号(後述)などがあり、図9のように表記する。なお、P***、T***、C***、M***の順序は自由である。

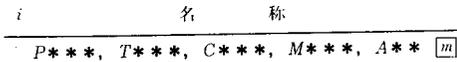


図9 アクティビティの図式表現法

**：実数値

***：習熟ファンクション番号(後述)または実数値

P***：生起確率、T***：所要時間、C***：費用

M***：所要人員または機械台数、A**：余裕時間

i：アクティビティ識別番号

m：モディフィケーション番号(後述)

(2) 生起確率

確率ノードの直後に続くアクティビティにのみ与えるもので図10のように表わす。この図の例は合格の確率が70%の場合である。

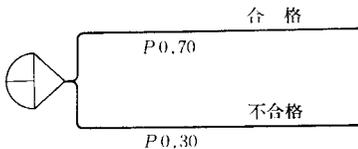


図10 生起確率の表記例

(3) 所要時間

シミュレーションを実施する場合は、所要時間は0または正の整数で与える。単位時間の取り方は自由である(図11)。

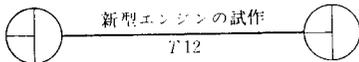


図11 所要時間の表記例(12単位時間がかかる場合)

(4) 所要費用

所要費用は、その作業項目に必要な額を与える。単位費用の取り方は自由であり、負の値を与えてもよい(図12)。

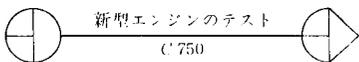


図12 所要費用の表記例(750単位費用がかかる場合)

(5) 所要人員または所要機械台数

人員または機械のどちらとして解釈するかはプロジェ

クトごとに統一しておく。これは、延べ数ではなく、その作業項目を遂行するのに実際に必要な人員数または機械台数を与える(図13)。

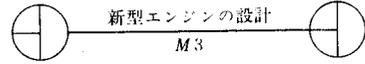


図13 所要人員の表記例(3単位人員がかかる場合)

(6) アクティビティ識別番号

識別番号は、アクティビティの前のノードと後のノードを共有する複数のアクティビティが存在する場合、識別のために与える。ネットワーク中の全アクティビティを識別するための番号として使用してもよい(図14)。

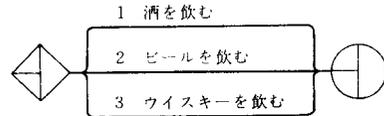


図14 アクティビティ識別番号の表記例

(7) ダミー・アクティビティ

作業項目間の順序などの複雑な因果関係を明確に表現するために実質的に作業のないものをダミー・アクティビティとして記入してもよい。

(8) 結線記号

ネットワーク図を手書きする場合、アクティビティの混線を防ぐために、図15のように結線記号⇨を用いてもよい。結線記号⇨の中には、A~Zの文字を書き込んで対応づける。

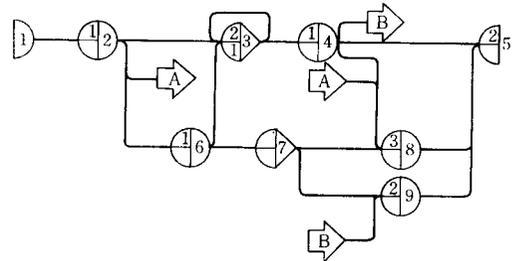


図15 結線記号の使用例

3.3 モディフィケーション(ノードの置き換え)

作業の進捗結果によってプロジェクトの進め方を変更するような計画をモディフィケーションによって図示する。具体的には、指定した作業項目の終了時点で、指定したノードの置き換え(モディフィケーション)を行なうというもので、□の中に番号を入れ、該当部分に書く。置き換えられるノード間は、もとのノードの下部と新しいノードの上部を点線で結ぶ。その際、前述の結線記号を使ってもよい。新しいノードをさらに置き換えるようにモディフィケーションを何重にも指定してもよい。図

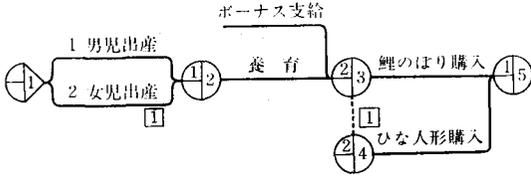


図16 モディフィケーションの例

16の例は、男児であれば鯉のぼりを購入するが、女児の場合はひな人形を購入するという計画を示す。

3.4 習熟性の表現

同一の作業項目の繰返し実施によって、たとえば図17のようにその作業項目の生起確率、所要時間・費用・人員等が変化する場合がある。これらは習熟ファンクションを設定することによって表現する。具体的には、図18のようにアクティビティごとに習熟ファンクション番号（たとえばF1, F2, …）を記入し、別途1回目から順次その値を記入した一覧表を作成する。

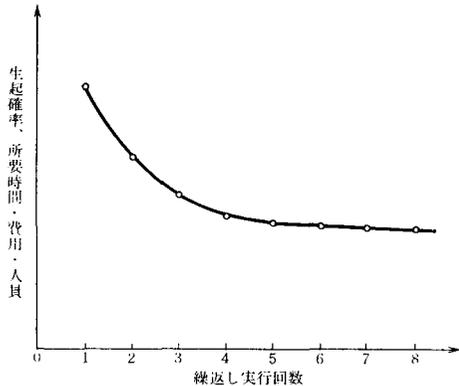


図17 習熟曲線

アクティビティ名称

PF1, TF2, CF3, MF4

習熟ファンクション番号	回数	回数													
		1	2	3	4	5	6	7	8	9	10				
F 1	1	0.3	0.4	0.5	0.6	0.7	→								
F 2	5	3	→												
F 3	100	60	→												
F 4	4	3	→												

(矢印は以下同様の意味)

図18 習熟ファンクションの例

3.5 代替案の比較 (意思決定のパターン)

1つのネットワーク中に複数の意思決定ノードが存在する場合、それらの意思決定ノードでのアクティビティの選択が関連性をもつことがあるが、その選択の組合せをパターンと呼ぶ。図19の例ではパターンとして4つが

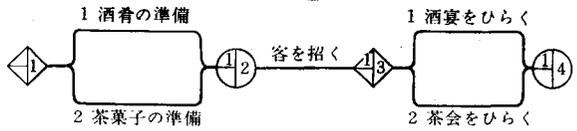


図19 意思決定のパターンの例

パターン	意思決定ノード1での選択	意思決定ノード3での選択
1	1	1
2	1	2
3	2	1
4	2	2

考えられるが、意思決定の整合性から有力なのはパターン1とパターン4であるから、この2つを中心に比較する。EASYGERTの汎用シミュレータを使用すれば、複数のパターンを容易に比較できる。

3.6 代替可能要素の併行開発

実際のプロジェクトでは、費用より納期を重視して代替可能な要素を併行開発し、一方がO.K.となれば他方の開発を中止してただちにO.K.となったほうの製作を開始するというような場合がある。これは図20のように表わす。

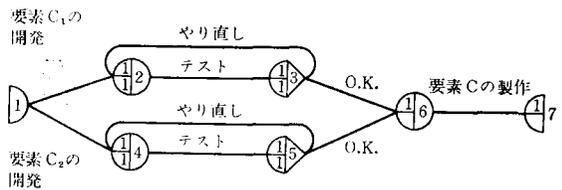


図20 代替可能な要素の併行開発の例

3.7 後追い作業の中止

図20の併行開発の例のように一方がO.K.となって他方の開発が中止される場合がある。このように最終到達ポイント(終了ノード)に到達するために無意味となった作業を後追い作業と呼ぶ。EASYGERTのシミュレーションでは、後追い作業はただちに中止され、費用・人員などの計算もそれに対応して処理される。

3.8 期限設定による作業の流れのコントロール

モディフィケーションの利用により、たとえば図21の

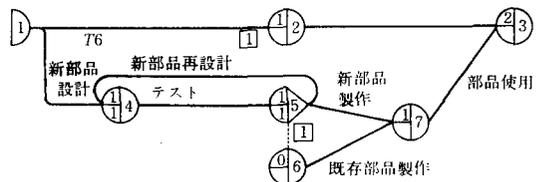


図21 期限設定による作業コントロールの例

ように6カ月経過時点で新部品の開発に成功していなければ、その開発を中止し、既存部品を製作し使用するというような作業の流れを表現できる。

3.9 ネットワークの分割・合成

確定ノードを左半分と右半分に分割してそれぞれ開始ノード、終了ノードとすることによってネットワークを分割することができる。分割・合成は EASYGERT のシミュレータのデータ・ベース上でも容易に実行できる(図22)。 (次回につづく)

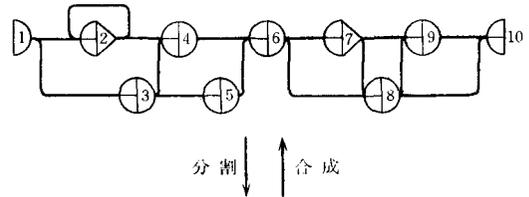


図22 ネットワークの分割・合成の例

参 考 文 献

- [1] 石堂一成他：不確定性をもつプロジェクトの計画手法 MHI-GERT. 三菱重工技報, 13, 6(1976), 1023-1031
- [2] 石堂一成, 他：ネットワークによる計画評価手法 GHI-GERT. 日本OR学会1977年度秋季研究発表会アブストラクト集, 1977, 20-23
- [3] 石堂一成：「目で見える計画」の手法—GERTの実用化. 日本OR学会1982年度秋季研究発表会アブストラクト集, 1982, 26-27
- [4] Pritsker, A. A. B. and W. W. Happ : GERT Part I Fundamentals. *J. of Industrial Eng.*, 17, 5(1966), 267-74
- [5] Pritsker, A. A. B. and G. E. Whitehouse : GERT Part II Applications. *J. of Industrial Eng.*, 17, 6(1966), 293-301