日本建築学会技術報告集　第3号, 256-261, 1996年12月
AIJ J. Technol. Des. AIJ. No. 3, 256-261, Dec., 1996

コンピュータによる建築構造設計支援のあり方に関する研究
（その1）FACT法の提案

STUDY REGARDING THE REQUIREMENTS FOR BUILDING STRUCTURAL DESIGN SUPPORTED BY THE COMPUTER : (Part 1) Proposal of the Fact Method

稲田達夫 ——*1

*1

Tatsuo INADA

Building structural design support programs must flexibly meet various demands that the designers should have. However, most of the current building structural support programs are formed in conventional procedural manner, therefore most of them are inflexible. For many years, the author has studied the requirements of the computer programs that truly support building structural design. As a result, it is recommended to introduce the new type of program development method ; the FACT method, originated and named by the author, which is fundamentally different from conventional procedural programs.

キーワード：
建築構造設計支援プログラム，プログラム開発手法

Keywords :
Building structural design suported by computer, Program development method

1　はじめに

建築構造設計という活動は，地震力や風力といった，複雑な自然現象を外力条件とし，また建築特有の単品的生産を基本とすることから，構造設計者の様々な場面における判断と思考の積み重ねが要求される活動であり，正に典型的な人間の知的活動と呼ぶにふさわしい総合的な営みとして捉えられる。そして，近年のコンピュータ利用技術の急激な発展に伴い，それをコンピュータにより支援しようとする試みもまた本格化しており，ますます重要度を増しつつあるのが現状である。

しかし，建築構造設計支援プログラムは本来設計者が持つ様々な要求に対し柔軟に対応し得るものでなければならないが，既存の建築構造設計支援プログラムの多くは，そのような観点を欠いており，その意味では固定的で融通性に欠けるものがほとんどである。それで，筆者は建築構造設計という活動を真に支援するコンピュータプログラムとはいかにあるべきかという事柄について，考察を進めてきたものであるが，その結果としてＦＡＣＴ法と呼ぶ従来のプログラム方式とは根本的に異なる新しいプログラム方式を考案したので，それを建築構造設計支援プログラムに適用することを提案するのが，本論文の主旨である。

2　構造設計の特殊性

建築構造設計という活動を真に支援するコンピュータプログラムはいかにあるべきかを議論するために，まず建築構造設計の本来のあり方を考察し，その特殊性を明らかにする。

2.1 関係条件の充足性

対象とする事象の成立する為の条件が，あらかじめ完全に与えられており自明であるような問題を，「関係完全充足問題」と呼ぶ。コンピュータプログラムが対象とする問題が，関係完全充足問題であれば，諸条件の充足性の証明は一義的に行う事が可能であり，プログラムは単純プロセスとなる。

一方，対象とする事象の成立する為の条件が，完全には与えられない問題を，「関係不完全充足問題」と呼ぶ。このような問題を，コンピュータプログラムで扱おうとする場合には，何らかの形で人間の判断に基づいて擬似的な関係充足性を作り出す必要があり，問題は複雑となる

建築構造設計を支援するコンピュータプログラムの特殊性を考察するためには，まずその関係充足性を明らかにすることが重要と考えられる。

*1 三菱地所(株)丸ビル改築設計室　副室長・博士（工学）
（〒100 千代田区丸の内2-2-1　岸本ビル2F）

*1 MITSUBISHI ESTATE Co., LTD.

2.2 構造設計の関係充足性

建築構造設計の目的は，建築主が要求する経済的条件と耐荷性能，使用性能を満足する建築を，建築家のイメージした建築本来の機能，特徴を生かしながら，かつ建築基準法等の定める社会的要請を満足しつつ，実現することである。それを具体的に実現するためには，まず適切な荷重条件を設定し，建築を構成する各種構造部材の保有する諸性能を把握し，さらに荷重によって励起される各部材の損傷の状態を計算等を駆使して予測し，その妥当性を建築構造工学の成果を基礎として，検証することが重要と考えられる。

これらを検証するための建築構造工学の成果としては，通常，専門書，論文，あるいは，建築学会等の定める規基準，指針，その他建築基準法等があり，それらを総合することにより建築構造設計を行うための工学体系が形成される。建築構造設計分野はこれらの設計体系の整備が比較的進んでいる分野であり，それに従えば，建築構造設計は比較的容易に行われると考えられる。従って，建築構造設計は，関係完全充足問題として捉えられがちであるが，必ずしもそうではない。実際には，構造設計者は設計を進めるに当たり，様々な場面で未知の問題に遭遇し，その都度様々な判断を強いられることになる。その判断の仕方により，建築構造設計は異なる結論を生み出すこともしばしば有り得る。その意味で，建築構造設計は関係不完全充足問題として，捉えるべきである。

建築構造設計が，関係不完全充足問題として捉えられる所以としては，例えば以下のような事項が挙げられる。

a) 地震外力に関する研究は活発に行われているが，その成果が設計で取り扱うために十分に成熟した技術となっているかについては，疑問が残る。例えば建設地における最大級の地震をどのように想定するかという建築構造設計上の最も基本的な問題についても，未だ共通認識はできていないのが実状である。

b) 要求性能を満足する為に必要な設計用諸元（例えば，部材の復元力特性，破壊のメカニズム等）が，必ずしも全てが解明されているわけではない。特に，大地震を想定した場合の高軸力下における柱の挙動や，鋼材の破壊性状等，未解明の問題が多く，想定外力下における建物の状態の把握は，構造設計者の判断が要求される重要な問題である。

c) 上記設計用諸元に基づいて行われる様々な解析方法の妥当性についても，全てが明らかになっている分けではない。例えば，上部建物の減衰特性等のような極めて基本的な問題についても，把握されていないのが実状である。

従って，以上述べた様な「関係不完全充足問題」としての特徴を持つ建築構造設計を進めるに為には，まず構造設計者は未だ解明されていない多くの問題が存在することを常に認識し，その不完全な関係を充足する為に必要な情報の収集に努力し，その評価能力を養い，それに基づく創意工夫と総合的判断を行うことが重要である。

3 建築構造設計支援プログラムの要件

第2章で建築構造設計は，関係不完全充足問題であると述べた。従って，建築構造設計支援プログラムの役割としては，設計の様々な場面で要求される構造設計者の判断を，如何に支援しうるかが，重要な要件となる。それでここでは，本来の建築構造設計支援プログラムの要件とはいかなるものであるかを考察するために，その要件と関わる重要な概念を列記してみると，以下の通りとなる。

3.1「構造」的把握に対する支援

一般に「構造」という言葉の意味は，「事象を要素の集合として捉えた場合の，要素間の関係の集合」と定義することができる。また，「事象を要素の集合体として捉え，その内部法則を調べることにより，全体の挙動を把握しようとする科学的方法」のことを「構造」的アプローチと呼ぶことができる。

「構造」的アプローチは，建築構造設計のような，関係不完全充足性に特徴付けられる人間の典型的な知的活動を進めるためには，極めて有効な方法と見ることができる。事実，建築構造設計は，モデル化（つまり「要素」への展開）と法則性（つまり「関係」）の追求を基礎として進められるものであり，その際，構造設計者にとってその「構造」（つまり「要素」間の「関係」）の把握は，建築構造設計上の最も重要な課題と言える。

従って，建築構造設計支援プログラムは，単に建物の数値的な目標の到達の正否を判定するのみならず，その「構造」の把握のために有効なツールでなければならない。その為には，「構造」的アプローチによく適合した表現形式を，プログラムの表記法に採用していることが望ましいと考えられる。

3.2 拡張性，再利用性

構造設計者は本来，独自の判断やモデル化，アイデア等を設計に反映するために，独自のプログラムを作成することを望むものである。そのためには，他人の作ったプログラムの一部を容易に改変したり，過去に作られた数多くのプログラムを編集して次のプログラムに容易に発展させることが可能であることが，極めて重要となる。このような構造設計者の本質的な要求に応え得るプログラム方式を実現するために必要な機能として，プログラムの「拡張性，再利用性」が挙げられる。

拡張性とは，プログラムの機能を追加したり，プログラムの一部の機能を改良したりすることを可能とする能力のことである。再利用性とは，過去に作成されたプログラムを部品として自由に編集することを可能とする能力のことである。そのためには，プログラムは常に分かり易く維持管理が行われていることが重要である。建築構造設計支援プログラムを構造設計者にとって真に有効なものとするためには，プログラムの自由な拡張性，再利用性は極めて重要な要件と考えられる。

3.3 正確さ

3.2で，プログラムが分かり易く維持管理されていることが重要であることを述べたが，それを実現するためのもう一つの要件として，プログラムの「正確さ」の問題が上げられる。

正確さとは，プログラムが仕様書通りに確実に機能を発揮する能力のことであって，プログラムの正確さを実現する為には，

①プログラムの仕様書の作成において，利用者と作成者が共通に理解できる，簡明な表記法を採用していること。

②採用された仕様書の表記法に対して，それと同一の「構造」を持つプログラムの表記法が，確立していること。

の2点が重要と考えられる。

3.4 柔軟な操作性

建築構造設計支援プログラムは一般に，扱うデータ量が膨大であ

り，例えば三菱地所の保有する建築構造一貫計算プログラム「ＡＳＴＳ」においては，１００を超える入力データ形式を持ち，１５０を超える出力形式を持ち，１８００を超える内部データ項目を持つ。そして，それらのデータを柔軟に操作し得ることが建築構造設計支援プログラムにとっての重要なテーマとなる。

通常プログラムにおいては，データとプロセスは「構造」的に分離されていることが望ましく，そのためにデータベースマネジメントシステム等を利用することが考えられるが，建築構造設計プロセスのように，並列的なデータフローを伴い，また複雑な解析プロセスが介在するプログラムにあっては，そのような方法ではデータとプロセスの依存関係を断ち切ることは難しい。従って，従来のプログラミング方式では，データ操作はプロセスの起動手順に依存することになるが，データ量が膨大な場合そのことは利用者の負担を増大させる。もし，データとプロセスの依存関係を分離することが困難であるならば，逆にプロセスの起動手順がデータの操作に依存して決定されるようにプログラムを組むことが可能であれば，この問題は解決する。

従って，建築構造設計支援プログラムの第4の要件は，プロセスの起動手順にデータ操作が依存しない，逆にプロセスの起動手順がデータ操作に依存して決定されるような，「柔軟な操作性」を確立することである。

3.5 Ｆｏｒｔｒａｎ言語との親和性

技術計算プログラムの動作環境（コンピュータハード及びＯＳ等）の技術革新は目ざましく，極めて流動的であるのが実状である。一方技術計算プログラムの寿命はそれに比べはるかに長く，従って標準化に対する配慮は是非とも必要となる。新しいプログラムを作成した場合のバグの発生は煩わしい問題であり，既存のプログラムが利用可能であるならば，例え簡単な問題であるとしても極力再利用を心掛けるべきである。

従って，過去における数多くのプログラム資産を最大限に利用するためには，「Ｆｏｒｔｒａｎ言語との親和性」は，建築構造設計支援プログラムにとっての欠くことのできない重要な要件の1つである。

4 問題解決のためのアプローチ

以上述べた5つの要件を満足する方法を模索するために，まず最初に過去においてコンピュータ科学分野で確立された様々な方法論を，建築構造設計支援プログラムの開発に援用することを考えてみよう。これら5つの要件に関わる重要な概念として，以下の3つが上げられる。

- 構造化手法
- オブジェクト指向
- 人工知能

まず，建築構造設計支援プログラムの開発運用に「構造化手法」を応用することを考えてみよう。構造化手法は，「構造」的アプローチを基礎としたプログラム開発方法論であって，その最大の利点は，プログラム本来の「構造」を表現するための明解な記述方式を確立したことである。そしてこのことは，以上で述べた5つの要件のうちの特に「構造的把握に対する支援」「拡張性，再利用性」の2つの要件に対し，大きな効果をもたらすことが予想される。しかし一方構造化手法は，構造的アプローチに基づく仕様書の表記法と，構造的アプローチとは異なる表記法で成り立つ手続き的プログラム方式の間で，一貫性を確保する事が困難であり，このことは特に「正確さ」の点で大きな問題を引き起こすことが予想される。

次にオブジェクト指向を建築構造設計支援プログラムの開発運用に応用することを考えてみる。オブジェクト指向的アプローチは，上記5つの要件の内「拡張性，再利用性」及び，「柔軟な操作性」の観点からは極めて有効な方法と考えられるが，その利点を有効に生かすためには最終的にオブジェクト指向言語で記述されることが望ましく，「Ｆｏｒｔｒａｎ言語との親和性」の面で問題が生じる。

また，人工知能的アプローチは，本来建築構造設計支援プログラムのような古典的な計算プロセスのプログラミングを支援するために考案されたものではなく，人間の思考を代行するためのものであり，本質的に異なるアプローチである。その結果生まれたプログラム方式の中に，科学技術計算プログラムにとって有効な概念が存在するとしても，根本的な問題の解決は期待できない。

5 ＦＡＣＴ法の提案

ここでは，第3章で明らかにした建築構造設計支援プログラムの5つの要件を満足するための，新しいプログラム方式である「ＦＡＣＴ法」の提案を行う。

以下に「ＦＡＣＴ法の基本的思想」及びそれを実現するための「表記法」「動作原理」の観点から，「ＦＡＣＴ法」を考案した基本的な考え方を明らかにする。

5.1 わかりやすさの実現

1人で進めてきた仕事の内容について，他人の意見やアイデアを求めることにより，思いもよらぬ発想の展開に遭遇することは，建築構造設計を進める上で，よく経験することである。このような情報の共有ないしは交換によって，知的創造性を増幅させることは，構造設計者にとって極めて有意義なことである。

建築構造設計支援プログラムは，単に建築構造計算のための道具として存在するのではなくて，上記のような情報の交換ないしは共有による構造設計者の知的創造性を増幅するための「共有情報」としての役割をも担うものであると筆者は考える。建築構造設計においては，構造設計者の独自の創意工夫や判断に基づいて作成される建築構造計算プロセスもまた知識の一部であると考えられる。建築構造計算プロセスを実体化したものは建築構造計算用のサブルーチンプログラムである。従って，建築構造設計支援プログラムにあっては，それを構成するサブルーチンプログラムもまた，知識の一部として捉えることができる。

従って，建築構造設計支援プログラムを構成するサブプログラム群を「共有情報」として有効に活用する基盤が確立されれば，構造設計者にとって極めて有意義なものとなるはずである。しかし，そのためには，過去において作成された多くのプログラムが，わかりやすく維持管理されていることが不可欠となる。それを実現することが，本論文でＦＡＣＴ法の導入を提案する本来の主旨である。

プログラムのわかりやすさを実現するためには何が必要であるか，まず最初にそれを明らかにする。

a) 表記法の確立

一般に，プログラムの方式を，その動作上の特徴から分類すると，「動的なプログラム方式」と「静的なプログラム方式」の2つのプログラム方式に分類することができる。

動的なプログラム方式とは，プログラムの実行時に人間の介在により動作が影響を受けるプログラム方式であり，会話型のプログラムがこれにあたる。一方，静的なプログラム方式とは，プログラムの実行時に人間の介在の影響を受けないプログラム方式であって，一括処理型のプログラムがこれにあたる。

また一般に，プログラムの方式を，その表記上の特徴から分類すると，「命令型のプログラム方式」と「宣言型のプログラム方式」の2つのプログラム方式に分類することができる。

命令型のプログラム方式とは，プログラムの構文の順序付けが重要な意味を持つプログラム方式である。一方宣言型のプログラム方式とは，構文の順序付けは意味を持たず，データや関数等のプログラム構成要素間の関係付けが重要な意味を持つプログラム方式である。一般に，動作が動的なプログラムに対しては表記法としては命令型のプログラム方式が適しており，静的なプログラムに対しては宣言型のプログラム方式が適していることが言われている。

そして，プログラムのわかりやすさを実現するためには，プログラムの動作上の特徴に適したプログラムの表記方式を採用することが，まず重要と考えられる。

その為に，「構造化手法」を基本とし，それに「オブジェクト指向」と「人工知能」の利点を加味する事により，この問題の解決を図ることを試みる。その為に筆者は，建築構造設計支援プログラムを，「データの操作」「データと関数の関係」「関数」の3つの要素に分離して取り扱うことを提案する。

「データの操作」とは主としてデータの入出力操作のことであって，この部分のみプログラムの動作時に人の介在を許すことにする。従ってプログラム方式としては動的であり，表記法としては命令型のプログラム方式を採用することが望ましいと考えられる。

「データと関数の関係」とは，プログラムのデータ構造のことであって，あるデータを作成するために必要な関数とその基になるデータの関係を図式的に表現したものである。この関係により，建築構造設計に内在するの本来の「構造」が表現されることになる。この部分は，動作時に人の介在による影響を受けることはなく，静的なプログラム方式となる。従って，本来は宣言型のプログラム方式を採用することが，プログラムのわかりやすさの観点からも望ましいと考えられる。

「関数」とは，データとデータの間に介在する処理をプログラム化したものであり，実体はＦｏｒｔｒａｎ言語で記述されたサブルーチンプログラムである。このことにより，Ｆｏｒｔｒａｎ言語との親和性は確保される。

従って，ＦＡＣＴ法で組まれたプログラムの表記法は以下のようなものとなる。

データの操作は「ＦＡＣＴメインプログラム」と呼ばれるＦｏｒｔｒａｎ言語で記述されたメインプログラムで表現する。ＦＡＣＴメインプログラムにはデータの操作のみを記述し，関数の起動の手順は一切記述する必要はない。これは，人間が行うデータ操作とプログラムの処理手順を完全に分離するためのものである。この表現形式は，「オブジェクト指向」の持つ利便性と同様の効果を狙ったものであるが，このことは膨大なデータを扱うプログラムの利用者の負担を大幅に軽減するはずである。すなわち，建築構造設計支援プログラムの要件の観点から見れば，特に「柔軟な操作性の確保」の点で有効と考えられるものである。

データと関数の関係は，「構造化手法」の代表的な図式表現であるデータフローダイヤグラムと類似の表記法を採用する。この図式表現のことをＦＡＣＴ法では「データ関数関連図」と呼ぶ。この表現形式を採用することにより，建築構造設計プロセスに内在する「構造」とプログラムのデータ構造が同一の形式でわかりやすく表現可能となり，建築構造設計支援プログラムの第1の要件である「構造的把握に対する支援」の問題を満足することができる。さらには構造設計支援プログラムの重要な要件である「正確さ」，「拡張性，再利用性」の観点からも有効と考えられる。

さらに，データ関数関連図とb)で後述する「矛盾解消機構」を組み合わせることにより，構造化手法におけるドキュメントとプログラムの不一致の問題を解決し，従って建築構造設計支援プログラムの重要な要件である「正確さ」の確保の問題も解決可能となる。なぜならば，「矛盾解消機構」の導入により，仕様書の表現する構造に従って，プログラムは自動的に実行されることとなり，仕様書からプログラムへの変換が不要となるからである。

関数の実体はＦｏｒｔｒａｎ言語で記述したサブルーチンプログラムである「関数プログラム」で表現する。関数プログラムで取り扱うデータは，データ関数関連図で表現されたデータを除いて，全て関数プログラム内のみでやりとりされる。つまり，個々の関数は，データと処理の関係で定義された入力データと出力データの関係を表現するのみであり，関数間は相互に独立の関係となる。

最後に建築構造設計支援プログラムの重要な要件である，「Ｆｏｒｔｒａｎ言語との親和性」の問題について見れば，ＦＡＣＴメインプログラム及び関数プログラムをすべてＦｏｒｔｒａｎで記述することにより，「標準及び互換性」の問題は解決されることになる。

以上ＦＡＣＴメインプログラムとデータ関数関連図，関数プログラムを作成することにより，ＦＡＣＴ法によるプログラムの記述は完了する。

b) 動作原理の導入

次に，プログラムのわかりやすさを実現するために必要なことは，a)で確立した表記法に基づいてプログラムの実行を行うための，動作原理を確立することである。そのためにＦＡＣＴ法では，「矛盾解消機構」と呼ぶ，独特の動作原理を導入する。矛盾解消機構は，a)で示した「データ関数関連図」と組み合わせて用いられるものであるが，人工知能分野における応用技術である，宣言型のプログラム方式と自己発見的動作原理の組み合わせの概念を応用したものであり，ＦＡＣＴ法の最大の特徴を成すものである。

即ち，ＦＡＣＴ法で作成されたプログラムは，ＦＡＣＴメインプログラムを起動することにより動作する。ＦＡＣＴメインプログラムは，データの操作のみが記述されており関数の起動については一切記述されていない。従って，ＦＡＣＴメインプログラムで指示された人間の行うデータ操作手順に基づいて，最適な関数の起動手順を自動的に決定する動作原理を導入することが必要となる。この動作原理のことを「矛盾解消機構」と呼ぶ。矛盾解消機構はデータ関

数関連図に基づいて動作する。この独特の動作原理を導入することにより，ユーザーのデータ操作に基づいて必要なプロセスが自動的に選択され起動されることになり，3.4で述べたデータの操作がプロセスに依存しない，逆に言えばプロセスの起動手順がデータ操作に依存して決定される柔軟なプログラム方式が可能となる。即ち，矛盾解消機構の導入により始めて，a)で述べたＦＡＣＴ法の記述形式は意味を持つことになり，建築構造設計支援プログラムの全ての要件を現実のものとして満足することが可能となるのである。

以上まとめると，ＦＡＣＴ法とは，「構造化手法」「オブジェクト指向」「人工知能」の技術的成果の融合を基本とし，「データ関数関連図」に代表される簡明な表記法を採用し，「矛盾解消機構」と呼ばれる独特の動作原理に基づいて動作する，プログラム開発技法である。

5.2 ＦＡＣＴ法によるプログラムの記述

ＦＡＣＴ法におけるプログラムの記述は，「データ関数関連図」「関数プログラム」「ＦＡＣＴメインプログラム」の3つを作成することにより行われる。

a)データ関数関連図

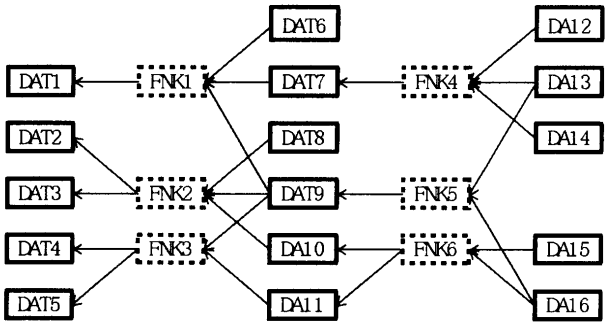


図1　データ関数関連図

データ関数関連図はＦＡＣＴ法で組まれたプログラムのデータと関数の関係を図式表現したものである。（図1 参照）

図1 で実線で囲まれているのがデータである。点線で囲まれているのが関数である。関数の右側に矢印で連結されているデータを入力側データと呼ぶ。関数の左側に矢印で連結されているデータを出力側データと呼ぶ。

「データ関数関連図」の作成に当たっては，以下の規則を遵守することが必要である。即ち，

規則1）関数とその入力側のデータとの関係（図1 で関数と右側のデータとの関係）は多対多の関係を許す。一方，関数と出力側のデータとの関係（図1 で関数と左側のデータの関係）は1 対多の関係に限る。つまり，あるデータの内容を決定する関数は1 つである。

規則2）データの流れの方向は，図1 で右から左への1 方向に限る。即ち，左側のデータが右側の関数の入力となる様な，ループの関係は許されない。

b)関数プログラム

関数プログラムとは，関数の実体であるＦＯＲＴＲＡＮサブルーチンプログラムのことである。関数プログラムは，ＦＡＣＴ法で組まれたプログラムにおける，個々の関数の機能を規定する。関数プログラムの典型的な構文形式を図2 に示す。

```
      SUBROUTINE func01(iname,ni,oname,no)
      COMMON //aaa(siz1),bbb(siz2),ccc(siz3),
     ¥          ddd(siz4),eee(siz5)
      CHARACTER iname(ni)(*),oname(no)(*)
C                                  ＊＊データの読み込み＊
      CALL DBGET(u2,iname(1),aaa,siz1,iflag)
      CALL DBGET(u2,iname(2),bbb,siz2,iflag)
C                                  ＊＊計算の実行＊＊
        ・関数の実際の処理手続きを記述する
C                                  ＊＊データの書き込み＊
      CALL DBWRIT(u2,oname(1),ddd,siz4,iflag)
      CALL DBWRIT(u2,oname(2),eee,siz5,iflag)
      RETURN
      END
```

図2　関数プログラムの構文形式

c)ＦＡＣＴメインプログラム

ＦＡＣＴメインプログラムとは，ＦＡＣＴ法におけるメインプログラムのことである。ＦＡＣＴメインプログラムが起動される事により，ＦＡＣＴ法で組まれたプログラムは実行に移される。

ＦＡＣＴメインプログラムにおいては，通常のプログラムにおける「処理の手続き」，つまり「関数の起動の手順」は一切記述する必要はない。記述する必要があるのは，「対象とするデータの取扱い」，つまり外部データの内部データへの読み込み，及び内部データの外部データへの書き出しのみである。

ＦＡＣＴメインプログラムの典型的な構文形式を図3 に示す。

```
      PROGRAM fctmpr
      COMMON /comb/ aaa(siz1),bbb(siz2),ccc(siz3),
     ¥              ddd(siz4),eee(siz5),fff(siz6)
C                                  ＊初期設定＊
      CALL RFASET(u1,iflag)
      CALL VFOPEN(u2,fname,imod,iflag)
      CALL AIFINT(ainame)
C                                  ＊データの読み込み＊
        ・外部より初期データの読み込み
          例)   READ(5,*) aaa,bbb,ccc
C                                  ＊データのデータ
C                                        ファイルへの転送＊
      CALL DBWRIT(u2,dname1,aaa,siz1,iflag)
      CALL DBWRIT(u2,dname2,bbb,siz2,iflag)
      CALL DBWRIT(u2,dname3,ccc,siz3,iflag)
C                                  ＊結果の変数上への転送＊
      CALL DBREAD(u2,dname4,ddd,siz4,iflag)
      CALL DBREAD(u2,dname5,eee,siz5,iflag)
      CALL DBREAD(u2,dname4,fff,siz6,iflag)
C                                  ＊結果の書き出し＊
        ・計算結果を外部に書き出す
          例)   WRITE(6,*) ddd,eee,fff
C                                  ＊クローズ処理＊
      CALL VFCLOS(u2,iflag)
      CALL RFACLS(iflag)
      STOP
      END
```

図3　ＦＡＣＴメインプログラムの構文形式

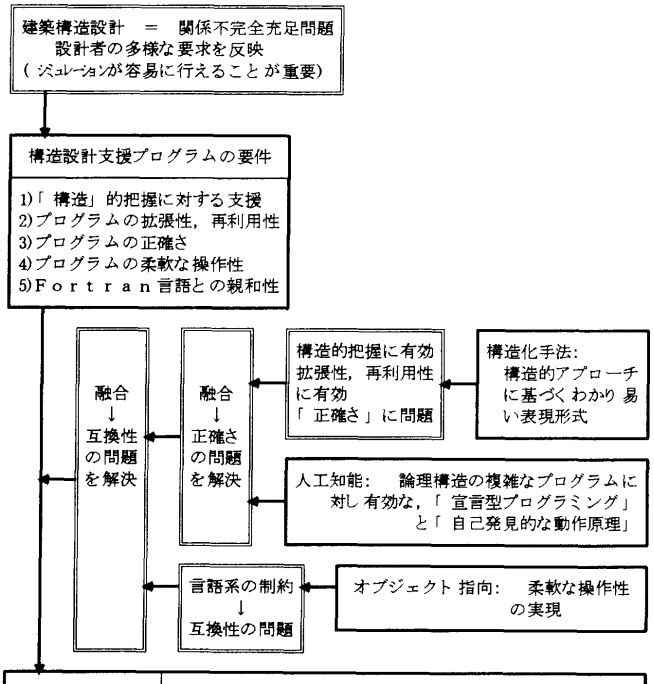
5-3)ＦＡＣＴ法の優位点

一般にプログラムの仕様を決める際には，本質的な面に焦点をあてて，より詳細で偶然的な問題については，できる限り後まで無視することが望ましとする考え方がある。即ち，「抽象化」とよばれ

る概念である。一方，特定のプログラムが管理するデータに対しては，そのプログラム内でしか変更を許さないとする考え方がある。即ち，「データ独立」の概念がこれにあたる。

ＦＡＣＴ法では，まず特定のデータを決定する関数は，一つの関数プログラムに限定しており，徹底した「データ独立」が成されている。また，プログラムの「構造」を決定するデータとデータの関係は，抽象化の概念に基づいて，データ関数関連図により，静的で整然とした秩序が確立されている。プログラムは，入出力操作のみを記述したＦＡＣＴメインプログラムの要求により動作するが，その際，矛盾解消機構の働きにより，一義的に，利用者に負担を一切かけることなく，その動作は確定する。

この様に，プログラムの「分節化」の徹底を図ることにより，所期の目標である，典型的な知的活動としての建築構造設計を真に支援するコンピュータプログラムが実現したことが，ＦＡＣＴ法を他の開発手法と比較した場合の優位点と見ることができる。



ＦＡＣＴ法の提案	備　　考
ＦＡＣＴメインプログラム	「データの操作（つまりデータの入出力処理）」のみ記述した，命令型のプログラム。主として柔軟な操作性を実現するためのものである。
データ関数関連図	プログラムが扱うデータと関数の関係を図式的に表現したもの。プログラムの構造をわかり易く表現するために有効と考えられる。宣言型のプログラム方式を採用している。
関数プログラム	Ｆｏｒｔｒａｎ言語で具体的な処理の内容を記述した，サブルーチンプログラム
矛盾解消機構	データの操作がプロセスに依存しない，逆に言えばプロセスの起動手順がデータの操作に依存して自動的に決定される，独特の動作原理。

図4　ＦＡＣＴ法の提案

6　まとめ

本論文は，筆者が行った一連の研究を学位論文としてまとめたものの一部を，紹介することを目的としたものである。即ち，人間の典型的な知的活動としての建築構造設計を支援するコンピュータプログラムはいかにあるべきかという事柄について考察を行い，まず「創造的活動」としての建築構造設計は，関係不完全充足問題としての特殊性を内在することを明らかにした。そして，そのような複雑な問題を支援するプログラムの要件として，

- 「構造」的把握に対する支援
- 拡張性，再利用性
- 正確さ
- 柔軟な操作性
- ｆｏｒｔｒａｎ言語との親和性

の5点が重要であることを述べ，さらにこのような要件を満足するためには，既存のプログラム開発方法論では困難であることを示し，それら従来のプログラム開発方法論を統合することにより，筆者が考案した「ＦＡＣＴ法」を導入することを提案した。（図4参照）

今後，「ＦＡＣＴ法の詳細」，特にその記述方式と動作原理である矛盾解消機構の働きを中心に報告し，さらに「適用事例」として，建築構造一貫計算プログラムに適用した場合と，振動解析プログラムの作成に適用した場合について，順次報告を予定している。

謝辞

本論文の執筆にあたり，丁寧なご指導を頂いた，東京大学教授秋山宏先生に心より感謝致します。

研究を進める上で，丁寧なご指導を頂いた，三菱地所（株）理事山田周平氏に心より感謝致します。

本論文の執筆にあたり，有意義な助言を頂いた，東京都立大学教授山崎真司先生に心より感謝致します。

参考文献

1) 稲田，永田，入江：構造一貫計算システム（ＡＳＴＳ）へのオブジェクト指向プログラミングツール（ＦＡＣＴ）の組み込み，1989.3，日本建築学会第１１回情報システム利用技術シンポジウム梗概集
2) 稲田，永田，入江：振動解析プログラム自動コーディングツール（ＦＡＣＴ－ＤＡ）の開発，1990.3，日本建築学会第１２回情報システム利用技術シンポジウム梗概集
3) 稲田，永田，溜，原：構造計算システム（ＡＳＴＳ）の改良について，「（その１）開発支援ツール「ＦＡＣＴ法」の適用」，1994.9，日本建築学会大会学術講演梗概集
4) 稲田：コンピュータによる建築構造設計支援のあり方に関する研究「ＦＡＣＴ法の提案」，1996.3，東京大学学位論文
5) P. Wegner著，尾内理紀夫訳，はやわかりオブジェクト指向，1992年，共立出版
6) 矢田光治監修，ＡＩ総覧，1987年，（株）フジ・テクノシステム
7) Yourdon, Constantine著，原田実，久保未沙訳，ソフトウェアの構造化設計法，1986年，日本コンピュータ協会
8) Tom DeMarco著，高梨智弘，黒田純一郎監訳，構造化分析とシステム仕様，1986年，日経マグロウヒル社
9) JAMES MARTIN, INFOMATION ENGINEERING BOOK I～III，1989年，Prentice-Hall International Editions
10) 淵一博監修，溝口文雄，古川康一，Ｊ・Ｌ．Ｌａｓｓｅｚ編，制約論理プログラミング，1989年，共立出版

[1996年6月27日原稿受理　1996年9月10日採用決定]