

数値情報環境の理論とその応用

畠 山 正 行*

Theory of Numerical Information Environment and Its Applications

by

Masayuki HATAKEYAMA
Tokyo Metropolitan College of Aeronautical Engineering

ABSTRACT

We propose a concept of "Numerical Information Environment (NIE, hereafter)" and its actual method of system constitution. The NIE has been studied and developed to store, analyze and simulate a large scaled numerical data and information of a target (natural phenomena or an artificial structure) interactively and iteratively. First, a global concept of "Numerical Information", and "Numerical Information World" are defined. To establish and operate the NIE, a concept of "Software Bus (I, II, III)" and its actual programming method have been developed on the basis of the World. The Software Bus system always plays the role of a distributing center of all the numerical data and information, and is also a general method to transmit them among the modules in the NIE and between the processes in the UNIX.

By using the software bus system, any kinds of direct numerical simulation or analyses programs can be mounted on the software "Bus" system. One more feature is that any modules in the NIE can directly be used in the user's program system without any modifications of source programs. Therefore, the system make-up is rapidly performed and operation of the NIE is flexible. To get a "view" of this database oriented system, a "Bus View" system has been developed with the aid of a graphic system. We especially describe an application system named "Dynamic DB System". We can conclude that the NIE is suitable for drawing various interesting and valuable knowledge or view of the user's "target".

1. はじめに

コンピュータの処理(計算)結果, 或は実験データなどを例えばデータベース化などして, 多種・多様・大量の数値データをして蓄積しておく, 或は数

値実験(シミュレーション)そのものの制御や観測を行いたい, そしてこれらの数値データを自在に取り出して解析する, といったソフトウェア・システムは, コンピュータを道具として駆使しているあらゆる分野の専門家に望まれている。しかし, そのようなソフトウェア・システムの汎用化された例はほとんどみない。

* 東京都立航空工業高等専門学校・数学科

本研究は計算理工学(Computational Science and Technology)の対象となる諸問題を数値計算アルゴリズムの面からではなく、ソフトウェア工学及びデータ工学的な面から解決に寄与する概念とその実現法を提案するものである^{1),2)}

まず計算理工学の現状解析を「何をコンピュータにやらせたいか?」という質問言語を構成する作業を軸にして試みた。その考察から導かれた包括的な概念として「数値情報環境(世界)」を提案し、従来のソフトウェアの組立て手法や構成の概念及びデータの位置付けを一新させる。次に数値情報環境を解明の対象(target)に対して構成する、具体的なデータ(数値情報)の配置と転送の一般仕様をSoftware Bus Iとして提案する。Software Bus とは hardware bus のエミュレーションのソフトの実現である。この概念を更に高度な拡張を経てプロセス間 Software bus と言うべき Software Bus II へと、更にその究極として Bus III へと発展させた。数値情報の管理・操作の手法としてデータベース・システム、ファイルアクセス・システム、直接利用法の3種を階層記憶システムの的な位置付けをしながら数値情報環境への展開について検討を行った。その結果、概括的ながらその筋道の見通しが立ち始めている。この検討の途上で、データベース・システムとSoftware Bus の機能及び構成上のアナロジーが存在する事が指摘され、Software Bus の発展に大きく寄与した。更には環境の構成・立ち上げ・制御及び応用についてもいくつかの実現法の提案を行っている。

本稿に於いてはこれらのうち特に応用面について述べる。本稿 2~3 は数値情報環境の理論の概略である。4 が応用機能について述べたもので、まずソフトウェアバスの応用面の特性を述べ、4.3 に於て特にダイナミック DB を採り上げて述べてある。本稿では紙面が限られているので 1 例のみを採り上げた。

2. 数値情報環境

実験による研究において研究対象とは例えばテスト・ピースであり、風洞模型であり、微量の物質材料であり、実験生物である。これらの研究対象(ターゲットと称する)に必要な条件を満たした実験環

境を構成・制御してやることによりターゲットは目的の現象を再現させる。この再現自然現象を(一般的な意味での)センサーを通じ観測(計測)し何らかの処理を行った後、ターゲットの解析・知見の見出しが行われる。このようにターゲットと環境の2つに分類される事が分かる。図1にそれらを示す。

コンピュータを用いた数値的研究においても同様な手続きを踏んで、ただし全てが数値という表現形式(形態)に置き換えられて、行われる。大雑把に見れば上記のターゲットはプログラムに対応し、それ以外のコンピュータ側の全てが実験環境に対応するといつてよいように従来は考えられてきた。図1参照。しかし、上記のプログラムが厳密に例えばテスト・ピースに対応し、それ以外のコンピュータ側の全てが厳密に実験環境に対応するかというとなっていない場合も多い。プログラムがテスト・ピースに対応しない最も顕著な例は I/O ルーチン(含データ表現ルーチン)とかデータ処理ルーチンなどである。テスト・ピースや風洞模型本体は入出力機能など本質的に持ち得ないし、自己の出力データを処理した後実験装置へ渡すなどと言う事もあり得ない。つまり、本来は環境が成すべき事柄を数値的研究の場合はターゲット自身が背負い込んでいると言えよう。それは従来のコンピュータ資源のパワー、機能、その他の面から一応やむを得ないものであった。

しかし、現在においては、特に32ビットのUNIX/Workstation の出現以後においては、こういった未分離な状況は改められても良いと考える。即ち、コンピュータ内部に実験環境(実験装置本体、センサー、計測装置、制御装置、データ処理装置、その他)に対応するそれら全てを数値データと演算メカニズムの set で置き換えた数値環境をターゲットと

	実験的研究	数値的研究
対 象 (target)	(風洞) 模型 テストピース 実験生物	離散化された微分方程式系のルーチン ダイレクトシミュレーションモデルのルーチン 数値データ群
環 境 (対象以外の 全て)	センサー 実験装置 計測機器 データ処理機器	データベース データ管理ルーチン データ処理ルーチン グラフィックルーチン I/O ルーチン

図1 ターゲットと環境

は全く独立に構築する事が必要になってきたと考える。各研究者各々が勝手な仕様で以て不十分かつ使い勝手の悪い、自己完結的な「閉じた環境」をへばりつかせていたが故にソフトウェアの互換性、移植性、柔軟かつ多様な利用性、使い勝手、ターゲットへの解析・知見の見出し、などに不十分な点がかなりあったのではないかと考えている。粗悪な環境ではターゲットの追求に深みを欠く。

特に物理現象の数値シミュレーションや数値的粒子を用いた(モンテカルロ法による)ダイレクト・シミュレーションなどにおいて、又、試作システム・モデルの数値テスト・数値実験などの遂行においては

1. モデル(ターゲット)の頻繁な取り替え、修正、パラメータ値の変更などが珍しくない。従って、プログラム・モジュールの取り替え、修正が簡単でしかも異質なモデルや異なる数値用メッシュでもよいように作られていなくてはならない。即ち、高度のしかも開かれたモジュラープログラミングが要求される。
2. 得られた未知大量のデータの管理・操作の簡単さや互換性(共通な流通性)が高くなければならない。

しかし同時にみたまには汎用のソフトウェア構成法、データ管理手法が考え出されなくては困難である。我々は汎用の数値(情報)環境という考え方を採ってみた。

以上をまとめると、次のようになる。

- (i) ターゲット(プログラム)は純粋に現象再現に必要なルーチンのみで構成する。
- (ii) 数値環境では入出力ルーチン、データ表現ルーチン(グラフィックなど)、データの種々の処理ルーチン(統計処理など)、プログラム制御・管理ルーチン等は全て環境側で整備する。

少なくとも従来は数値環境をコンピュータの内部にソフトウェア・システムとして構築された例は(厳密に言えば)ないと言ってよいであろう。ただしUNIXはoperating environment(作動環境)であるとされているが、この「環境」は我々が考えた数値「環境」と同一趣旨の概念である。我々はその作動環境の上に「数値」を主な取扱対象として研究を

する人々のための「環境」を新たに提案しようとしているわけである。従って数値環境はコンピュータ・システムのOSのうえにアプリケーション・システム・ソフトウェアとしてターゲットを包み込むがごとくに構築されるものであり、「環境」という名称が示すように多様な機能と汎用性、標準フォーマットを持ちながらもターゲットの要求に応じられる柔軟性を持ち合わせていなければならない。

3. ソフトウェア・バス・システム

「バス」はよく知られているように元来ハードウェア用語であり、VMEバスなどがその例である。ソフトウェア・バスとはバス(hardware bus)システムの利点、即ちコントロール・ボードの抜き差しだけでそのボードのもつ機能をシステムに追加・削除できるという利点を、ハードウェア・バスをemulateすることによりソフトウェアの世界の中で実現しようとするものである。周辺機器に対応するファイル(ツール、モジュール、データ・ファイル、その他)はコントロール・ボードに対応するバス・アダプターを介してバス・ラインに接続し、他の構成ファイルと連動して一連の処理を実行する。ハードウェア・バスのバスラインに対応するのは同一名称のcommon arrayである。もともとcommon arrayは「共用のデータ置場ないしデータ転送共通路」の役割をもつものである。又、配列はハードウェアで言えばRAMメモリーの役割も併せ持っている。ハードウェアのCPUに対応するものはシステム制御プログラムGCTである。ソフトウェア・バス(I)の実現とはcomon arrayの導入であることがわかる。

software bus array : データ・バス配列には数値データ、コメントを収めてある。アドレス・バス配列はデータ・バス中の各データの格納場所の情報が入っている。コントロール・バス配列の情報は各ブロックの制御パラメータとして多様な使われ方をする。

GCP(General Control Program) : ハードウェアにおけるCPUはデータ処理も担当するが、GCPの役割はシステムを構成するファイル(ツール、独立プログラム・システム、サブルーチン、等々)のどれに制御を渡すかを選択する役割のみを持つ。

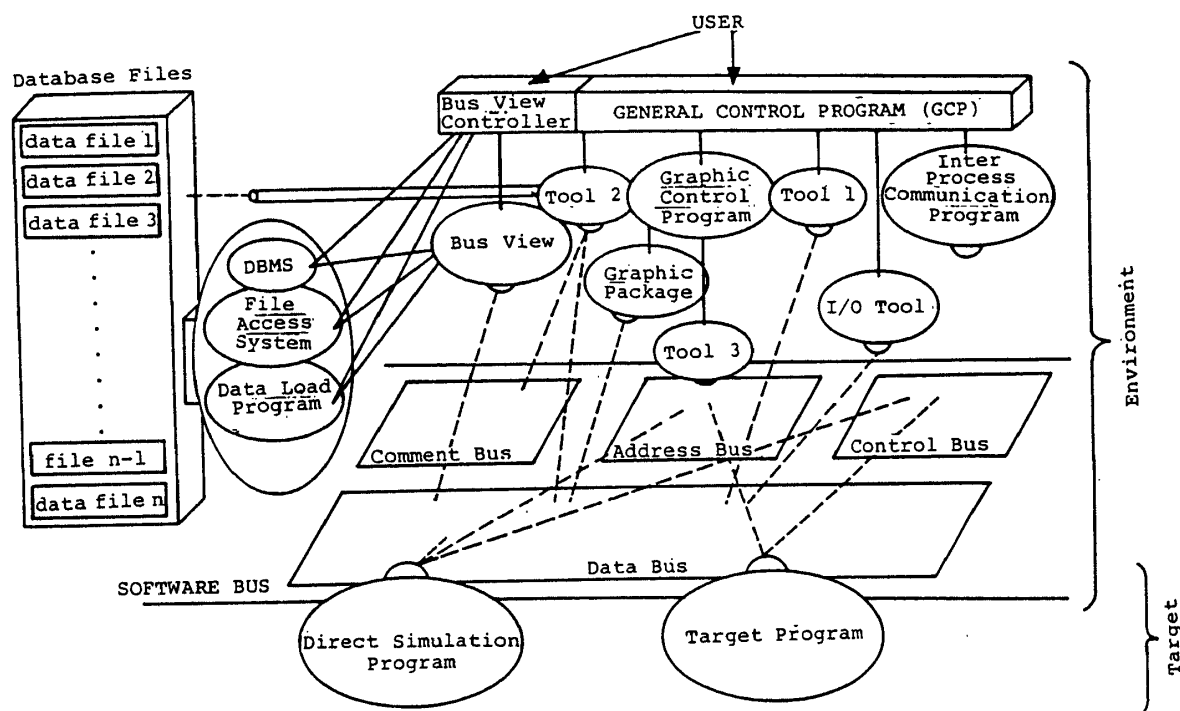


図2 数値情報環境の概観

バス・ビュー：ソフトウェア・バス・システムにのせたツールやプログラムを動かしている途中でデータを見たい (view) 場合が出て来る。配列中のデータ・情報・制御パラメータの出力と加工のルーチンである。数値出力とグラフィック出力がある。

図2に数値情報環境システム全体を模式図で示す。

4. 数値情報環境の応用例

まずDB機能を充分活用できる分野としてデータ解析問題が挙げられよう。大量のデータをファイルから取り出し、解析ツールを使って解析し、必要なデータのみを出力し、結果も同一のファイルに収めておく、という形式で用いる場合などNIEは有利なシステムである。

次に「数値解析分野」への応用である。この分野ではプログラム中のサブルーチン取替えの簡単さや、NIEのデータ監視機能、人工的なデータ操作、データベース機能などは便利な機能として解析による現象追求に深みを加える事ができよう。

三番目は「ダイレクト・シミュレーション問題」への応用である。この問題に対応してはモデル構成途上や、テストランの時のモジュールの交換のし易

さ、データ監視機能、人工的データ操作、データの入出力の際の任意性・データベース機能の随時の発動・データの引用・格納、等々メリットを挙げればきりが無いほどである。恐らくNIEを使いこなせば最もメリットの見出せる方法のうちの一つであろう。

四番目の分野は「ソフトウェア・バンク」としてである。NIEの利用としては不利ではあるが基本データ・ファイルのヘッダー部分の知識機能、データ・コマンダーの利用、UNIXのSCCSの利用などでソフトウェア・バンクを構築できよう。

Dynamic DBシステム

図3にその概念図を示しておいた。まずターゲットとなる(通常)ダイレクトシミュレーション・プログラムと、それに必要なデータを持つデータ管理システム(DBMS, ファイルアクセス・システム, または直接利用のプログラム・サブルーチン)のインターフェイス・ルーチンをソフトウェア・バスに乗せて数値情報環境を立ち上げる。ソフトウェア・バスとデータ管理システム(図の中ではstatic DBとある)の間にはソフトウェアバス管理ルーチン(デーモン)が介在しており、データの入出力を管

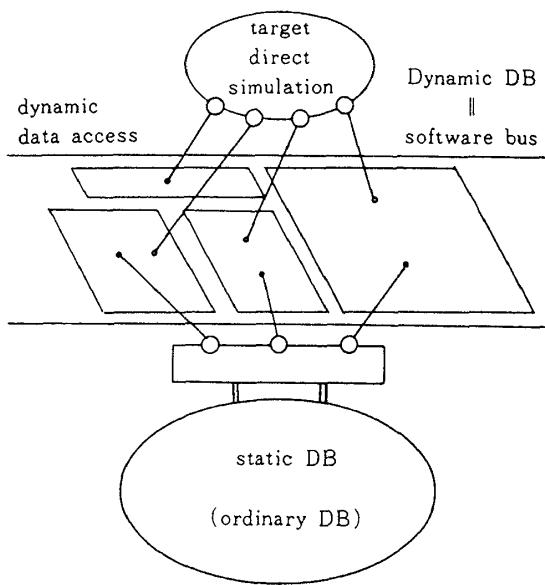


図3 Dynamic DBシステム

理すると同時に未展開データについてはデータ管理システムに出力を要求し、大量の不要データができれば引き上げ要求を出して元のデータファイルに再収容する機能を持つ。さてターゲットとなるプログラムが走り始め、ソフトウェア・バスに入出力を行いながら各ルーチンが情報を処理・表示し始めると、ソフトウェア・バス管理ルーチンにデータ・アクセスの通知をしつつデータをアクセスするが、もしソフトウェア・バス上にないときはソフトウェア・バス管理ルーチンがデータ管理システムに対してデータの要求を出す、またはユーザがソフトウェア・バス管理ルーチンに対し直接にデータ管理システムに対しコマンドを出してデータ処理と入出力を要求することもある。以上のような使い方により、従来にはなかったダイレクト・シミュレーション・プログラムとデータベースシステムの同時使用により、計算とデータ処理へのユーザのフリーハンドを与え、現象解析の効率化を実現している。

つまり図2からわかるように、広義DBシステムとしてのsoftware bus（原busと拡張bus）と任意にデータのやりとりし、入出力しながら、targetプログラムは自然現象又は人工構築物を数値的にNIE内で再現しつつける、というイメージである。そし

てSoftware busはstaticなDBと多量のデータをやりとりしつつtargetのdirect simulationを支えているわけである。

更に重要な事はこのdynamic DBにおいてはsimulateしつつコンピュータ内で動いているシステム内再現物理現象をいついかなる時間において、いかなる配列値や変数値（物理量）、等をもいかなる精度にても自在に取り出せるという数値計算の特長とともに必要データはすべて保存してあるというDBの特長もやはり備えている事である。それはつまり実験においてexcellent probeを実験対象の中に突っこんで（probeをつっこんだ事が実験対象への影響なしで）いかなるデータをも正確に取り出せる事と同等である。

6. 現状および結論

ソフトウェア・バス・システムは未解明の現象の数値的解析、ダイレクト・シミュレーションの際のデータの作成・管理・処理・表示の全過程において従来より高い処理効率をもたらすのみならず、数値データを通していま一步現象内部に踏み込み、新しい知見を得るのに適したソフトウェア・システムであると考えている^{1),2)}

本環境システムの具体的な開発研究は文献にあるように共同研究として研究会組織をもって行われつつある²⁾

参 考 文 献

- 1) 畠山正行：「DBSSのシステム理論とWorkstation/UNIX」東京都立航空工業高等専門学校昭和60年度研究紀要第23号，pp.85-136. 昭和61年5月。
- 2) 畠山正行（航空高専），三神 尚（東工大），高山文雄（電総研），関 光平（日立），久野茂（日本電気），矢吹道郎（上智大），小島功（電総研），日比一喜（清水建設）：「数値情報環境の理論」，東京都立航空工業高等専門学校研究紀要第24号，pp.151-252. 昭和62年6月。

