

本資料は 1997年7月5日付で  
登録区分変更する。 [技術展開部技術協力課]

## 地層処分ヴァーチャル・エンジニアリング・ システムの開発 ( I )

### 概要

(動力炉・核燃料開発事業団 研究委託内容報告書)

技 術 資 料		
開示区分	レポート No.	受領日
特	J1064 97-002	1998. 4. 14.
この資料は技術管理室保存資料です 閲覧には技術資料閲覧票が必要です 動力炉・核燃料開発事業団 技術協力部技術管理室		

1997年3月

日商岩井株式会社

この資料は、動燃事業団の開発業務を進めるため、特に限られた関係者だけに開示するものです。ついては複製、転載、引用等を行わないよう、また第三者への開示又は内容漏洩がないよう管理して下さい。また今回の開示目的以外のことには使用しないよう特に注意して下さい。

本資料についての問合せは下記に願います。

〒107 東京都港区赤坂1-9-13

動力炉・核燃料開発事業団

技術協力部 技術管理室

## 地層処分バーチャル・エンジニアリング・システムの開発（Ⅰ）

— 概 要 —

浅原進也 1)、 Larry BAIRD 2)、  
G.MEGSON 3)

### 要 旨

対話システム技術の研究開発では、地層処分バーチャル・エンジニアリング・システムと利用者との対話（情報交換）技術である情報の視覚化表示、データ入力の為のボタン、プルダウン・メニュー等のユーザー・インターフェイス技術の予備設計検討を行った。作業にあたり、個別に開発する技術の評価が必要な項目については評価用試作プログラムを開発し評価を行った。

評価プログラムの試作開発にあたっては、従来より研究が進められている、処分システムの性能評価に関する方法論を基盤としたシナリオ解析や、関連する個別現象の連成シミュレーション等の数値計算を効率的に統合して実行し、かつ、作図、作表、動画化等の後処理にかかわる操作を解かりやすく効率的に行えるよう考慮して検討した。

研究開発作業の結果、対話機能は地層処分バーチャル・エンジニアリング・システムの重要な機能であると判断された。又、開発にあたっては、オブジェクト指向プログラミング技術、ネットワーク並列処理技術、MIT/X-Windowシステムに関連する技術、等、多くの新しいソフトウェア開発技術が要求されるものと判断した。これらの技術開発課題は、次年度より開始する設計開発作業により詳細に開発蓄積することが必要である。

---

本報告書は日商岩井株式会社が動力炉・核燃料開発事業団の委託により実施した研究内容結果である。  
契約番号：080D0330

- 1) 高度情報化技研株式会社
- 2) Global Technology Associates, Inc. USA
- 3) Department of Computer Science. University of READING. U.K

Shinya ASAHARA 1) , Larry BAIRD 2) ,  
G.MEGSON 3)

## **Development of the Virtual Engineering System for High Level Radioactive Waste Geological Disposal Performance Assessment**

### **Executive Summary**

#### **Abstract**

The Virtual Engineering System for Geological Isolation enhances and integrates the traditional numerical simulation technology and related information database, while incorporating the essence of evolving computer science and technology. Its purpose is to; 1) provide directions for optimizing a safe waste disposal system by simulating predictions of long-term (1,000 years - 100,000 years) system behavior that cannot be simulated in laboratory experiments; 2) providing "a common ground" where, from the perspective of scientists and engineers from various disciplines, ideas can be exchanged and problems discussed in the evaluation of disposal systems ; 3) provide an interface between scientists/engineers and the public through the use of easily understandable visualization technology.

The preliminary scope of studies are interactive Graphic User Interface for future synthetic performance assessment, coupled model computation technology to simulate and visualize a wide range of scientific phenomena, multi-database development technology, and software quality assurance technology.

In fiscal year 1996, fundamental computer technologies were studied as the system development requirements research. During the study, a prototype hardware and software system was developed to evaluate individual technologies, their feasibility and their ability to be integrated into the system. As a result of the studies, we came to the following conclusions; 1) Further study of Interactive Graphical User Interfaces, with more prototypes, is necessary since their design has a direct effect on total system operability and functions; 2) The simulation functions need to be developed based on traditional technology for the start. However, at the same time, new approaches such as model collaborative simulation technology and agent collaborative simulation technology, should be introduced for a better understanding of disposal system behavior; 3) For the database functions, a technology for unifying different type of database system models (e.g. ODB and RDB) is needed to handle the multiple data types. 4) To ensure the quality of the results of simulations, a rigid software development and data quality policy needs to be developed for this unique but complicated system.

The above studies lead us to conclude that it is possible to develop elementally the technologies required for the Virtual Engineering System for Geological Isolation. Therefore, we consider feasible a total system to perform simulations for synthetic performance assessment of disposal systems in a computer virtual world. For the next fiscal year, more detailed fundamental studies will be needed for each function for the basic design of the total system.

---

Work performed by Nissho Iwai corporation under contract with Power Reactor and Nuclear Fuel Development Corporation.  
PNC Liaison : 080D0330

- 1) Advanced Information Technology, Inc.
- 2) Global Technology Associates, Inc. USA
- 3) Department of Computer Science. University of READING. U.K

# 【 目 次 】

1	概 要 .....	1
1.1	対話システム技術研究の概要 .....	3
2	対話技術研究 .....	10
2.1	マルチ・プラットフォームの対話制御機能 .....	18
2.2	システム操作とマルチ・エージェント処理 .....	20
2.3	マルチ・エージェントな対話環境 .....	24
2.4	対話処理機能から見たシステムの操作 .....	32
2.4.1	操作用ユーザーインターフェイス .....	32
2.4.2	試作対話システム制御機能概念 .....	38
2.5	性能評価シミュレーション面から考えたシステム操作要求 .....	49
2.5.1	安全性評価作業の基本的な操作 .....	50
2.5.2	評価期間 .....	50
2.5.3	多重バリア方式 .....	51
2.5.4	評価シナリオ .....	52
2.5.5	評価モデル .....	52
2.5.6	パラメータの不確実性 .....	57
2.5.7	総合的性能評価 .....	58
2.5.8	評価作業の不確かさ .....	59
3	シミュレーション技術研究 .....	62
3.1	仮想世界における実験の概念 .....	62
3.2	性能評価面から見たシミュレーションの目的 .....	68
3.2.1	米国の放射線防護基準 .....	68
3.2.2	ドイツにおける原子力施設に関する放射線防護基準 .....	71
3.2.3	スウェーデンにおける原子力施設に関する放射線防護基準 .....	71
3.2.4	サイト選定作業からの要求 .....	72
3.3	性能評価シミュレーションへの要求 .....	73
3.3.1	性能評価シミュレーションの基本的な考え方 .....	74
3.3.2	性能評価シナリオに関する考え方 .....	77
3.4	性能評価シミュレーション問題 .....	81
3.4.1	シミュレーション期間 .....	81
3.4.2	ナチュラル・アナログ .....	85
3.5	地下水シナリオの例 .....	89
3.5.1	地質環境と地下水シナリオ .....	89
3.5.2	広域地下水の流動 .....	92
3.5.3	局所的地下水の流動 .....	95
3.5.4	地下水の化学的性質 .....	97
3.5.5	鉱物組成及び微視的空隙構造 .....	102
3.5.6	ニアフィールド問題 .....	104
3.5.7	連性問題へのアプローチ .....	109
3.6	数値実験オブジェクトとして考えた評価モデル .....	111
3.6.1	各種のモデル .....	111
3.6.2	パラメータの不確実性 .....	114
3.6.3	システム操作面から見た評価シミュレーションの考え方 .....	116
3.6.4	ヴァーチャル・エンジニアリングの目指す科学技術 .....	117
3.7	オブジェクト指向のソフトウェア開発 .....	120
3.7.1	新たな研究開発の戦略としてのオブジェクト指向 .....	123

3.7.2	クラス階層のプログラム構造	127
3.7.3	オブジェクト指向による数理モデルのアプローチ	129
3.8	並列計算技術	130
3.8.1	計算量の推定	131
3.8.2	並列計算技術	135
3.8.3	並列シミュレーション・プログラム開発の事例	138
3.8.4	並列計算プログラムの開発	140
3.8.5	協調作業のための並列パラダイム	143
3.8.6	並列問題として地層処分性能評価	155
3.8.7	試作並列シミュレーション機能	157
3.9	シミュレーション問題への新たな観点	162
3.9.1	複雑な現象と複雑なモデルとして捉える方法	166
3.9.2	ゲームとシミュレーション	174
3.10	動画化シミュレーション・プログラムの試作	178
3.10.1	プログラム仕様	179
3.10.2	基本操作	181
3.10.2.1	操作開始	181
3.10.2.2	コントロール・パネル	184
3.10.2.3	画面操作	185
3.10.2.4	視点操作	188
3.10.2.5	スライス	189
3.10.2.6	ボリウムレンダリング	191
3.10.2.7	速度ベクトル	191
3.10.2.8	速度の変数	193
3.10.2.9	テキストラベル	193
3.10.2.10	データ検証	194
3.10.2.11	新しい変数の作成	194
3.10.2.12	画像ファイルの保管と印刷	196
3.10.2.13	テクスチャマッピング	197
3.10.2.14	Tclスクリプティング	197
3.10.2.15	キーボードの機能	198
3.10.2.16	制限事項	198
3.10.2.17	v5dimportnのユーティリティ	198
4	データベース技術研究	203
4.1	蓄積されたモデル(コード)の管理	203
4.2	コード選択の基準	204
4.2.2	適用性(公的領域/私的領域、情報源や参考文献へのアクセス)	204
4.2.3	ソフトウェア・エンジニアリング	205
4.3	解析コードの管理	206
4.4	視覚化情報管理	217
4.5	数値計算のためのデータの扱い	223
4.6	情報の処理	226
4.7	オブジェクト指向データベース管理技術	230
4.7.1	データベース・システム要件	233
4.7.2	RDBMSとOODBMS	236
5	ソフトウェア品質管理研究	240
5.1	地下水モデル開発とソフトウェア品質管理	240
5.2	モデル開発過程の留意事項	242
5.2.1	モデル開発過程	242

5.2.2	ソフトウェア開発と保守	2 4 4
5.2.3	コード設計とコード開発の文書	2 4 5
5.2.4	プログラム構造とコードの検証	2 4 5
5.2.5	モデル妥当性検査	2 4 7
5.2.6	検証と妥当性検査規準	2 4 9
5.2.7	文書	2 5 1
5.2.8	科学技術的レビュー	2 5 2
5.2.9	ポスト監査	2 5 4
5.3	アプリケーション検討事項	2 5 4
5.3.1	要件	2 5 4
5.3.2	データ収集	2 5 6
5.3.3	モデル化	2 5 6
5.3.4	モデルの検証と確証	2 5 7
5.3.5	感度解析	2 5 9
5.3.6	シナリオ解析	2 6 0
5.3.7	結果の解釈	2 6 0
5.3.8	監査	2 6 0
5.4	品質管理組織	2 6 1
5.5	品質管理計画	2 6 1
5.6	品質管理評価	2 6 3
5.7	品質管理とコード選択	2 6 9
5.7.1	コード選択過程	2 6 9
5.7.2	モデル開発必要性の公式化	2 6 9
5.7.3	コード特性の記述	2 7 0
5.7.4	コード選択規準	2 7 0
5.7.5	コード選択における品質管理	2 7 4
6	システムの試作	2 7 5
6.1	問題解決へのアプローチ	2 7 5
6.2	CASE	2 8 3
6.3	システムの性格	2 8 4
6.4	システムの規模	2 8 6
6.5	システム開発者とシステムの利用者	2 8 6
6.6	システム開発に潜在的な問題	2 8 9
6.7	プロトタイプング手法の導入	2 9 5
6.8	試作システムの仕様	3 0 1
6.8.1	全体システム概念構成	3 0 1
6.8.2	多重並列分散動作機能概念	3 0 2
6.8.3	ビデオ信号制御仕様	3 0 2
6.8.4	オーディオ信号制御仕様	3 0 3
6.9	シミュレーション・システム	3 0 3
6.10	操作システム	3 0 5
6.11	データベース・システム	3 0 8
6.12	パラメータ表示システム	3 1 0
6.13	オーディオ・ビデオ・システム	3 1 3

付録—A 試作動作シミュレーション画面集

付録—B 参考文献

## 1 概要

地層処分ヴァーチャル・エンジニアリング・システムは、従来シミュレーションや数値計算などと呼ばれてきた物理工学上の現象を、コンピュータにより数値的に計算模擬する技術をもとに、オブジェクト指向、並列計算法等、革新するコンピュータ・サイエンスの成果を地層処分の安全性評価研究に応用し、処分の安全性に係わる技術論拠となる情報を、多方面から総合的に評価することを目的として専用に統合開発するコンピュータ・システムである。従って、プロジェクトの目的を下記の通りに位置づけ、現在実施している個別現象解析レベルのシミュレーション技術研究開発とは別な計算科学技術を基盤とした研究開発とした。

地層処分ヴァーチャル・エンジニアリング・システムを開発・運用することにより、高レベル放射性廃棄物地層処分の安全性評価にかかわる、実験の難しい長期（数千から数十万年）に亘る現象を考慮性に係わる技術を判断し、処分システムの設計の最適化をするための指針等を得るとともに、多分野にまたがる専門家が、自己の領域と他の領域との関係を容易に理解するための基盤を提供することがプロジェクトの目標である。

平成8年度は、基盤となる計算科学技術関連中心に研究・整理し、地層処分ヴァーチャル・エンジニアリング・システム開発基盤情報を整備した。研究にあたり、試作評価の必要なハードウェア・ソフトウェア開発技術については個別に試作を行い、それらを統合化してプロトタイプ地層処分ヴァーチャル・エンジニアリング・システムとして実証評価を行った。

### (1) 平成8年度の作業

平成8年度は、初年度であるため、基盤技術として取り入れていくための幅広い分野について、予備的な検討を行った。また、システムの具体的な動作機能の概要を評価するため、プロトタイプ・システムの試作（図1-1）を行った。

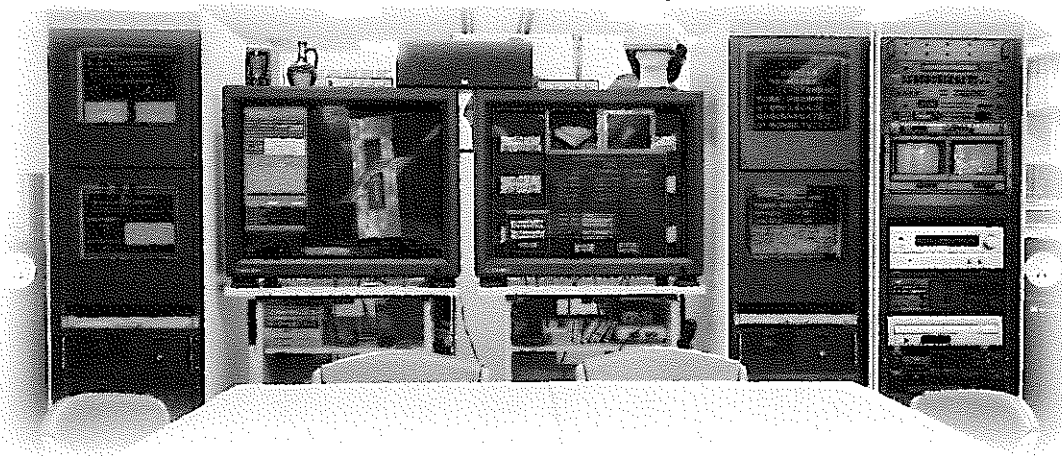


図1-1 試作した地層処分ヴァーチャル・エンジニアリング・システム外観



地層処分ヴァーチャル・エンジニアリング・システムの研究開発は、次の分野に分けて実施した。それらは、対話システム技術、シミュレーション技術、データベース技術、ソフトウェア開発管理技術である。

対話システム技術開発は、地層処分システムの長期的な挙動に係わる、作図、動画、画像処理、対話機能等、コンピュータによる人との対話技術（情報交換）の研究開発を行う作業とした。シミュレーション・システム技術開発は、地層処分システム挙動に係わる解析シミュレーションについて、オブジェクト・シミュレーション、ネットワーク分散並列シミュレーション等、新たな情報処理技術を応用し、並列事象・現象を数値的に模擬する手法の研究開発を行う作業とした。データベース技術開発は、地層処分ヴァーチャル・エンジニアリング・システム全体の基盤となる、膨大な各種の情報モデルを統合化した複合化データベースの研究開発を行う作業とした。ソフトウェア開発管理技術の開発は、地層処分ヴァーチャル・エンジニアリング・システム構築の為に開発するソフトウェアの設計、開発に係わる管理と品質保証を行なう手法の研究開発を行う作業とした。専用コンピュータ・システム構築は、地層処分ヴァーチャル・エンジニアリング・システムの設計・開発作業において必要となる対話機能等、要素技術の試作・評価等に利用するための専用の技術開発・実証用コンピュータ・システムとして整備・構築を行う作業（平成8年度は予備的な試作）とした。

地層処分ヴァーチャル・エンジニアリング・システムは、地層処分技術に係わる情報を統合化し、処分システムを計算機内に仮想的に構築して、長期に亘る地層中での挙動の評価を行う研究のために、専用に開発する仮想環境シミュレーション・システムとして開発するものとした。それは、コンピュータ内部に仮想的に組み立てられた処分システムについて、想定される種々の評価シナリオに沿って、長期に亘る処分システムの複雑な挙動のシミュレーションを対話方式で実行し、多方面にわたる専門家の共通な評価課題を視覚化を共通基盤として提供する機能を重視して開発した。

## （２）平成8年度作業の成果

以上のように、従来シミュレーションと呼ばれてきたコンピュータ利用技術は、物理工学上の現象を数値的に模擬することを目的としていた。地層処分ヴァーチャル・エンジニアリング・システムでは、従来のシミュレーション技術の蓄積を基盤に、従来の拘束条件の多い手続的計算法に比べ、より自由度の高い、試行錯誤的現象の数値計算を最新のコンピュータ・テクノロジーの応用により研究開発するものとした。それにより、従来型数値計算とは異なった、より現実に近い数値計算への可能性を検討した。

その結果、このようなアプローチは、数値計算から数値実験への移行として、仮想世界に於ける中規模の複雑系としての連成問題、動画化シミュレーションやオブジェクト指向ソフトウェア開発手法等による、実験の難しい長期（数千から数十万年）に亘る現象を、従来の個別現象を数値的に計算する技術とは別な、より複雑に連性したモデルやシナリオを（視覚化を基盤として行う技術）開発する為の有効な技術手段であると考えられた。

また、そのような手法により開発された地層処分ヴァーチャル・エンジニアリング・システムは、コンピュータ実験という観点から、将来工学的判断に資する情報を提供する事が期待できると考えら

れた。なぜならば、それは個別現象の解明のためのシミュレーションとは別な、総合的評価に資するシミュレーションの可能性を示すものであり、エンジニアリングと呼称する理由は、この“総合評価”を行なうシステムとして将来機能すると期待されるためである。

従って、地層処分ヴァーチャル・エンジニアリング・システムを開発・運用することにより、高レベル放射性廃棄物地層処分の安全評価にかかわる、実験の難しい長期（数千から数十万年）に亘る現象を考慮した各種のシナリオを複雑に組み合わせて、システムの長期的挙動の評価を行ない、地層処分の安全性に係わる技術を判断し、処分システムの設計を最高にするための指針等を得ることが地層処分ヴァーチャル・エンジニアリング・プロジェクトの目標である。

## 1. 1 対話システム技術研究の概要

対話システム技術研究作業は、地層処分ヴァーチャル・エンジニアリング・システムと利用者との対話（情報交換）技術である情報の視覚化表示、データ入力の為のボタン、プルダウン・メニュー等のユーザー・インターフェイス技術の研究開発作業とした。評価に必要な技術については評価用システムを試作し評価した。対話システム操作機能の開発にあたっては、処分システムの性能評価に関する方法論を基盤としたシナリオ解析や、関連する個別現象の連成シミュレーション等の数値計算を効率的に組立て、実行し、かつ、作図、作表、動画化等の後処理にかかわる操作を解かりやすく、効率的に行えるように人間工学的考慮を操作性にもたせるものとした。対話システム技術研究作業は、地層処分ヴァーチャル・エンジニアリング・システム全体の対話環境構築の為の技術を開発する作業とした。作業に於ては、他の“シミュレーション・システム”、“データベース・システム”、“ソフトウェア開発管理システム”等の各システムにて必要となる対話機能、及び動画処理等の技術を研究開発し、ライブラリーやサブシステムとして標準化し、地層処分ヴァーチャル・エンジニアリング・システムとして、一元統一化したユーザ・インターフェイスを開発する作業を実施した。図1. 1-1に試作した対話システムの画面を示す。



図 1. 1-1 試作した対話システムの画面

### (1) 対話環境基盤技術の整理

対話機能から想定する地層処分ヴァーチャル・エンジニアリング・システムは、地層処分の安全性評価に係わる各種のシナリオを複雑に組み合わせる操作機能を提供し、それらの複雑な現象を仮想的に数値計算により再現（シミュレート）するシステムとした。従って、操作面からシステムの構成を考えた場合、それはデータベース・サーバー、計算サーバー、対話操作コンソール、各種シミュレーション・パラメータ・ディスプレイ、PC等、機能別に分散したハードウェアを統合し、シミュレーションにおいては、それらが並列的に協調して動作する並列ネットワーク分散処理技術により一元統合化されたシステムとして考えられた。

システムの対話環境については、本書では画像処理機能を応用した多重画面制御技術に基礎を置くグラフィック・ユーザー・インターフェイスを対話環境と呼ぶこととした。また、マン・マシン・インターフェイスは、GUI以外の要素（マウス、キーボード、ビデオ、その他画面以外の対話機能に係わるハードウェア）を含むものとして対話環境“GUI”と区別した。平成8年度は、予想されるハードウェア・アーキテクチャについて、グラフィック・ユーザー・インターフェイス・プログラムを開発するための技術、環境基盤を整備し、次年度より開始する予定の統合的ソフトウェア開発のための準備を行なった。

また、作業にあたって、マルチ・プラットフォーム（ネットワークに分散接続された、異なったハードウェア、異なったオペレーティング・システム等）環境において、効率良く対話システムを開発す

るための技術（ソフトウェア再利用、プロトコル変換、自動移植等）を検討した。それらは、ソフトウェア・プロトタイピング・ツール、内部プロセス・インターフェイス、Make環境（CRAY, SGI, DEC, HP, SUN, IBM等ターゲットによる相違）等、ソフトウェア・エンジニアリング分野の技術・手法としてとりまとめ、対話システム開発作業の基盤を整備した。対象とした対話制御環境は、開発する地層処分ヴァーチャル・エンジニアリング・システム構築の基本機能要素として考えられる、下記のアーキテクチャとした。

・UNIX系対話機能

UNIX系対話システム開発技術評価作業は、MIT X11（X Window System Version 11）を基本とした対話システムの設計開発技術と開発環境（開発用のツールソフトウェア、ライブラリ、コンパイラ等）について評価する作業とする。又、X11は仕様上2次元の対話制御システムであるため、3次元の対話制御システムとして業界標準になりつつあるシリコン・グラフィクス社の4D GL系についても評価対象とする。

・PC系対話機能

PC系対話システム開発技術評価作業は、マイクロソフト社の提供するDOS、Windowsの2種のオペレーティング・システムを基本とした対話システムの設計開発技術と開発環境について評価する作業とする。

・Apple Macintosh対話機能

Apple Macintosh系対話システム開発技術評価作業は、アップル社の提供するMac OS（我が国では漢字トーク）を基本とした対話システムの設計開発技術と開発環境について評価する作業とする。

## （2）シミュレーション画面試作と機能評価

シミュレーション・システムGUIは、地層処分ヴァーチャル・エンジニアリング・システムの中核となる並列連成事象シミュレーションの実行を操作するためのGUIを想定した。試作にあたりGUI機能は、シミュレーション・シナリオの決定、シミュレーションの準備、シミュレーションの実行、シミュレーション結果の整理（解析モデルの検証・確認）という4種類の分野に分割して考えた。尚、シミュレーション・システムGUIについては、機能分析段階から分散型対話方式シミュレーション（DIS: Distributed Interactive Simulation）を前提にネットワーク・パラレル・マルチ・プラットフォーム・システムとして設計した。

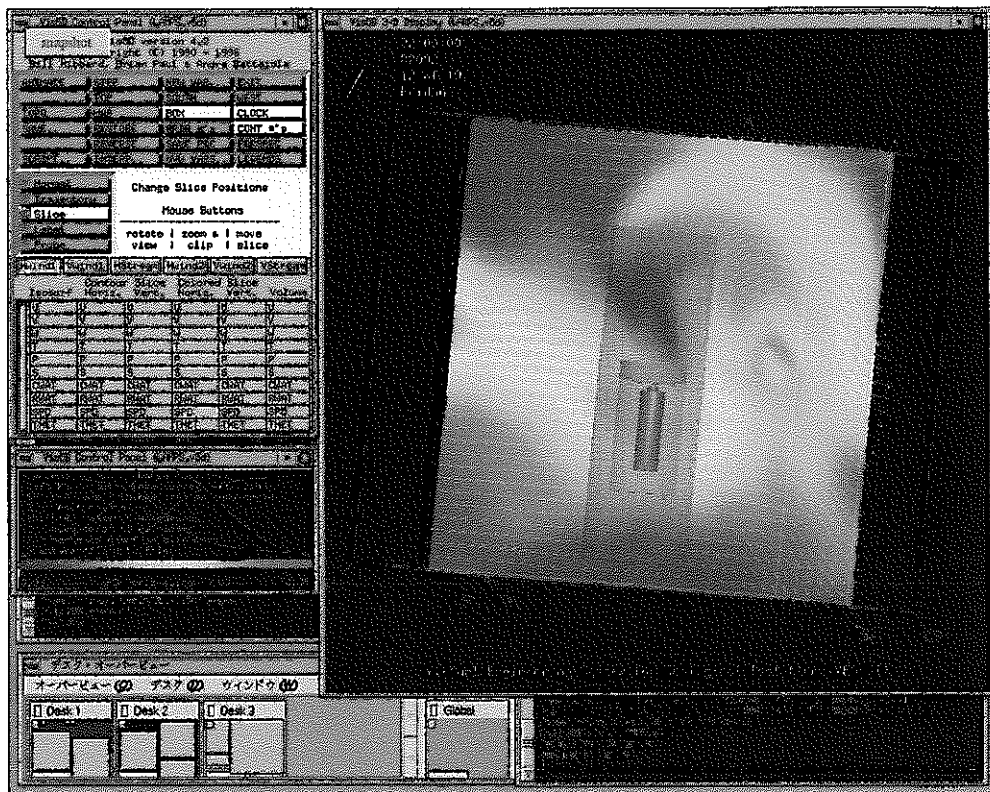


図 1. 1-2 シミュレーション動画処理の例

### (3) データベース画面試作と機能評価

データベースGUIは、システムの情報管理基盤となるデータベース・システムを操作するためのGUIとした。システムのデータベース・システムは、各種の情報を一元管理し、対話ユーザー、アプリケーション・プログラム等からの情報のアクセスを管理する為に構築するデータベースとし、別途研究しているシナリオ解析専用のデータベースとは分離して開発することとした。

但し、それら間の情報の交換は可能となるように開発した。分析にあたり、GUI機能はFEPシナリオ・データベースとの関係、内外の試験データ、設計データ等、エンジニアリング情報、モデル開発関連の文書、ソフトウェア、等数理解析情報という3種類の分野に分割して検討した。

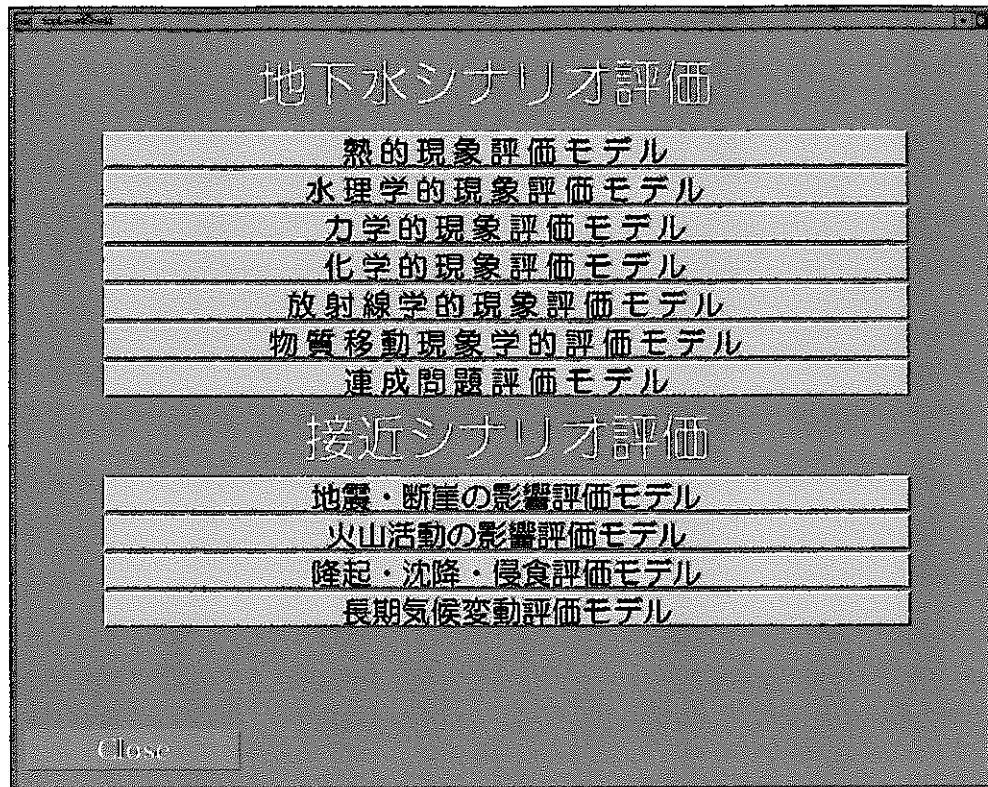


図1. 1-3 シナリオ・データベース画面例1（基本画面）

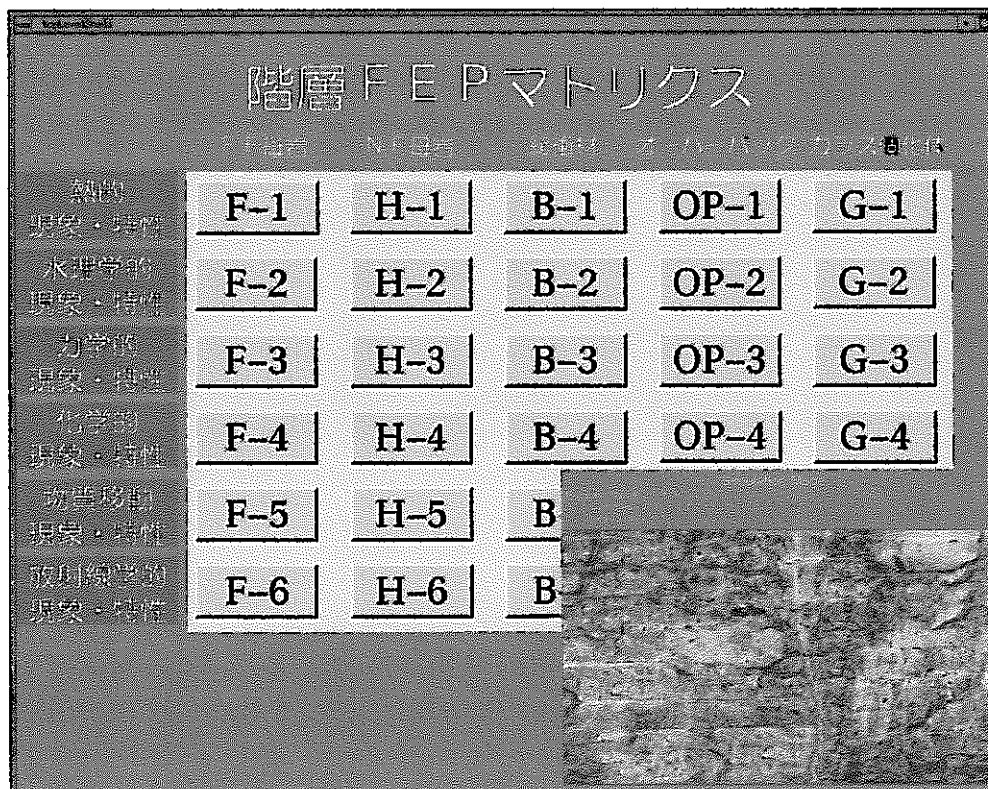


図1. 1-4 シナリオ・データベース画面例2（FEPと関連情報：写真のリンク）

#### (4) ソフトウェア開発管理画面試作と機能評価

ソフトウェア開発管理システムGUIは、地層処分ヴァーチャル・エンジニアリング・システム開発の為に開発される多くのソフトウェア（シミュレーション、データベース、等）のプログラミング作業レベルでの品質を管理し、トータルなライフサイクル管理を行う為のGUI機能として要求を分析した。

作業に当り、図1. 1-5、図1. 1-6に、シナリオに沿って管理されたモデル（コード）とオブジェクトの例を示す。このようにして、個別のソフトウェア開発や整備を総合的なシナリオに沿って管理するGUIを試作した。

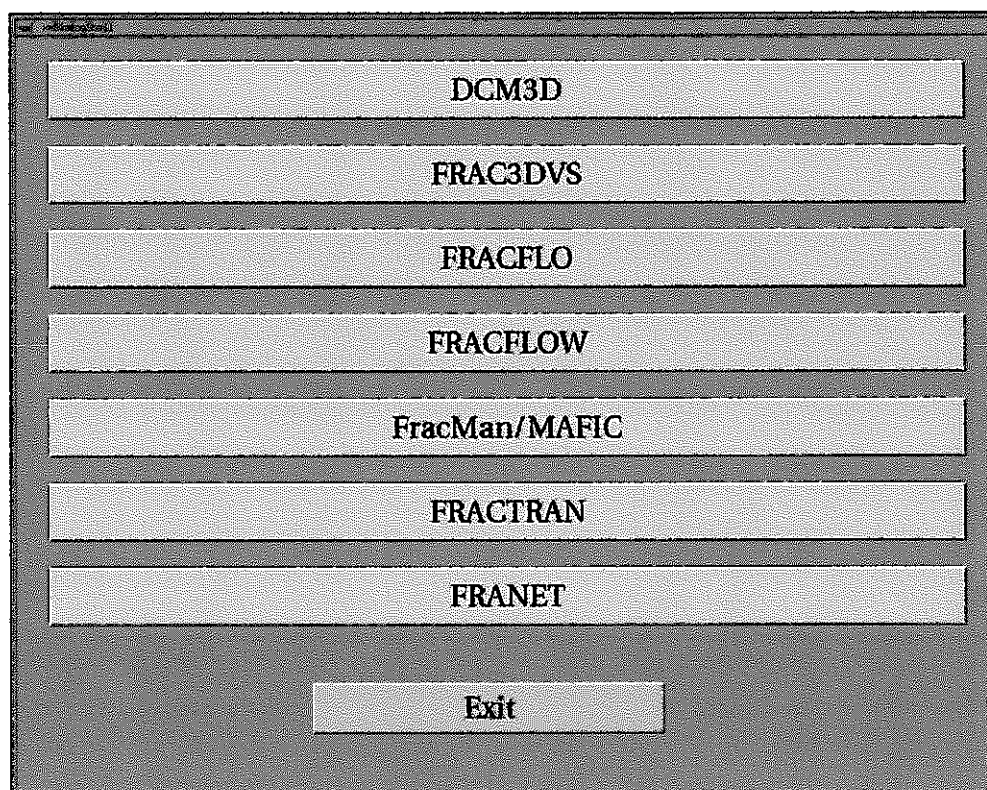


図1. 1-5 地下水モデルの例



図 1. 1-6 形状オブジェクトの例



## 2 対話技術研究

地層処分ヴァーチャル・エンジニアリング・システムの対話技術について予備的研究と試作を行った。対話技術は、地層処分ヴァーチャル・エンジニアリング・システムを利用して性能評価にかかわる操作を行う場合に、直接の操作の対象となるコンピュータ操作画面を制御するシステム・ソフトウェアのことである。この操作画面は、メニュー、作図、3Dオブジェクト表示、動画表現等、コンピュータ・グラフィックスの機能を最適に利用するとともに、地層処分性能評価シミュレーション作業にかかわる一連の操作について、使いやすく、かつ適当と判断される情報の組みに整理されて開発する必要がある。

従って、対話技術は、コンピュータ・グラフィックス技術、性能評価にかかわる一連の評価フレーム、そして、関連するサブシステムであるシミュレーション・システム及びデータベース・システム等との効率的なインターフェイスという、性能評価技術一般と広範な情報処理技術の複合技術から組み立てるものとした。

試作作業は、視覚化表示、データ入力の為のボタン、プルダウン・メニュー等、エンド・ユーザーと直接情報を交換する機能を受け持つソフトウェアについて実施した。試作したGUIプログラムは、目標となるシステムの機能を画面上に示すとともに、複数に渡るGUIプログラム相互の関連の評価等を行えるような相互の呼び出し機能（コールバック）を組み込むことにより、GUIシステム全体の操作を評価した。GUIの開発にあたっては、従来より研究が進められている処分システム性能評価に関する方法論を基盤としたシナリオ解析や、関連する個別現象シミュレーション等のモデル開発作業を効率的に統合し、作図、作表、動画化等の後処理にかかわる操作を解かりやすく行える操作を合わせて考慮した。

平成8年度の作業では、地層処分ヴァーチャル・エンジニアリング・システム全体の対話環境構築の為の基本概念をプロトタイプとして試作した（以下、この対話プログラムをGUI: Graphics User Interfaceと呼称する）。試作システムは、“シミュレーション・システム”、“データベース・システム”、“ソフトウェア管理システム”等の各システムで必要となる対話機能及び動画処理等の技術について、ライブラリーやサブシステムとした。

対話機能から想定する地層処分ヴァーチャル・エンジニアリング・システムは、処分の安全性評価に係わる各種のシナリオについて、計画する処分システムとその環境を構成するオブジェクト、解析コード、検証・確認用試験データ等、複雑な組み合わせ、複雑な現象を数値的に再現（シミュレート）するシステムとした。

システム構成面から地層処分ヴァーチャル・エンジニアリング・システムを考えた場合、それは下記に示す機能別に分散したハードウェアをネットワークにより統合し、シミュレーションにおいては、それらが並列的に協調して動作する並列ネットワーク分散処理技術により、一元統合化されたシステムとして考えた。

- ・シミュレーション用システム (Runtime Visual Simulation System)
- ・操作用システム (Operator Console System)
- ・データベース用システム (Data Base System)
- ・パラメータ・ディスプレイ用システム (Status Display System)
- ・オーディオ・ビデオ用システム (Audio Video System)

検討の結果、計画するシステムが異機種ネットワーク並列処理機能（マシン・パラレル）により統合化されるため、GUI開発環境については、完全な一元化はハードウェアをドライブする基本システムの相違等から難しいと考えられた。従って、システム・ハードウェアをUNIX系とP C系に二分し、UNIX系はMIT/X11を基盤としたXネットワーク・プロトコルによりグラフィック・ウィンドウ制御を統一するとともに、Zバファを画像フレーム・メモリとして有するハードウェアについては、シリコン・グラフィクス社が開発したIRIS4D/GLプロトコルを採用し、拡張することが現在入手可能な技術として適当であると考えられた。

また、X11のウィンドウ管理については、OSF/Motif xm intrinsicを採用することが適当であるとした。なぜならば、地層処分ヴァーチャル・エンジニアリング・システムのUNIXワークステーションが異機種のマルチ・コンピュータ環境となるため、メーカーが共同して開発したOSF/Motifがシステムの統合化に適しているためである。P Cについては、MacintoshとIBM PC系の2種類のアーキテクチャが市場で入手可能である。地層処分ヴァーチャル・エンジニアリング・システムのP Cとしては、仕様の公開されているIBM/PC系を採用し、GUI開発環境はUNIXシステムとサーバー・クライアントの構成が容易なアーキテクチャを提供するWindows95を採用した。

ユーザーインターフェイスの試作作業は、地層処分ヴァーチャル・エンジニアリング・システムのユーザー・インターフェイスについて、操作機能面からの要求を分析するための試作作業とした。作業は、プロトタイピング技法を用いて具体的な対話システムの骨子をプログラミングし、動作条件等を評価しながら進めた。

### (1) GUI基本動作概念

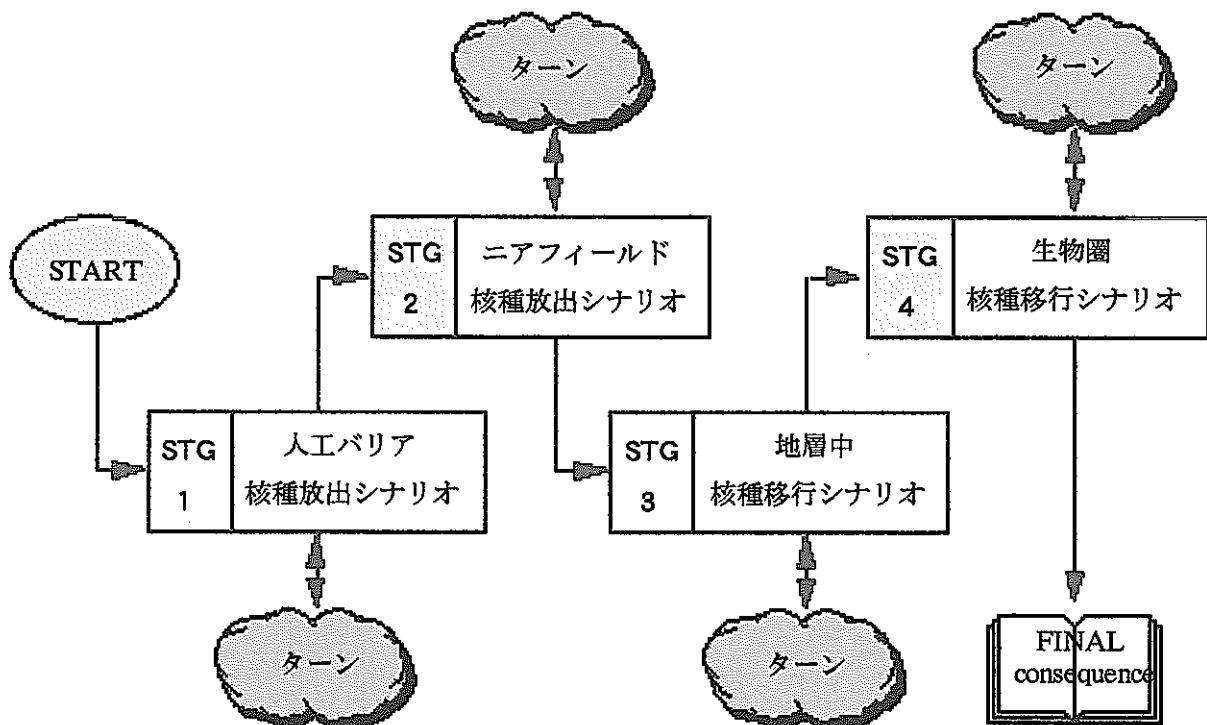


図 2 - 1 ステージの概念

GUIプログラムの基本部分を試作するにあたり、図2-1に示すステージの概念を試験的に考案した。ステージの概念は、地層処分ヴァーチャル・エンジニアリング・システムのシミュレーション機能を操作する基本的な考え方とした。

それは、多重バリア方式を評価するシナリオのサブセットについて分割したものとした。図2-1に試作システムの4段階ステージを示す。図2-1に示すように、シミュレーション・シナリオのサブセットを多重バリアに従って準備していると考えられる。これにより、評価はこれらの多重バリアごとの個別事象についてシナリオに沿って行なうものとした。

この様な考え方は、全体シナリオから考えた場合、各ステージ間のフィードバックや相互影響を含まない点が指摘されるが、個別の現象解析と全体シナリオ解析の中間に位置する手法として、それらを補足することが期待できるため導入した概念である。

操作にあたっては、このステージを単位として、それに含まれる個別シナリオについてFEP等を参考に評価シミュレーションを行ない、シミュレーション結果をターンという概念で保存することとした。各ステージにおいては、ステージ毎のシナリオの組みをステージ・シナリオと呼称することにし、それらの評価結果を定量的に把握できるように、単純な定量化指標としてターンの概念を試験的に考案した。

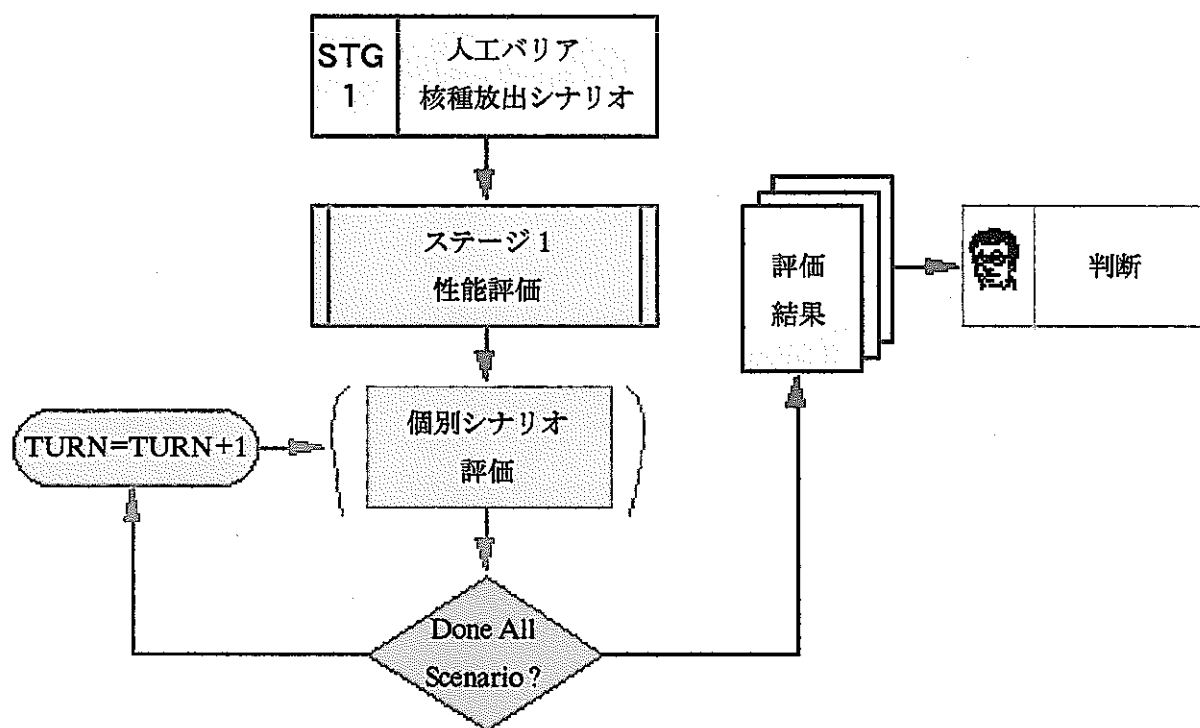


図2-2 ターンの概念

ターンは、各ステージにおける評価の指標とした。それは、各ステージの評価シナリオにかかわる評価の満足度を示すひとつの目安である。各ステージは、個別シナリオから構成されるサブ・グループとした。例えば、ステージ1（人工バリアの核種放出シナリオ）においては、ガラス固化体、オー

バーパック、ベントナイトについての個別シナリオが、現在整備中の階層F E Pマトリクスから抽出すると18のサブ・グループに分割される。これらのサブ・グループは、内部に現象レベルの個別シナリオを含むサブ・グループと考えた。例えば、ベントナイト中の化学的現象・特性というサブ・グループの中には、7種類の個別シナリオが現在提案されている。

このように、ターンとは個別シナリオごとの評価指標とした。例えば、ベントナイト中の化学的現象・特性というサブ・グループについてはターンの価は7である。GUI設計において、ターンは個々のシナリオに組み合わされた下記の状態を示す指標（評価作業の進行を示す状態変数）であり、地層処分ヴァーチャル・エンジニアリング・システムを利用して評価が行われたか否かを表すこととした。

- (完了ターン) ターン=0 : シナリオは発生したが評価は行っていない状態
- (未完了ターン) ターン=1 : シナリオについてなんらかの評価がなされた状態

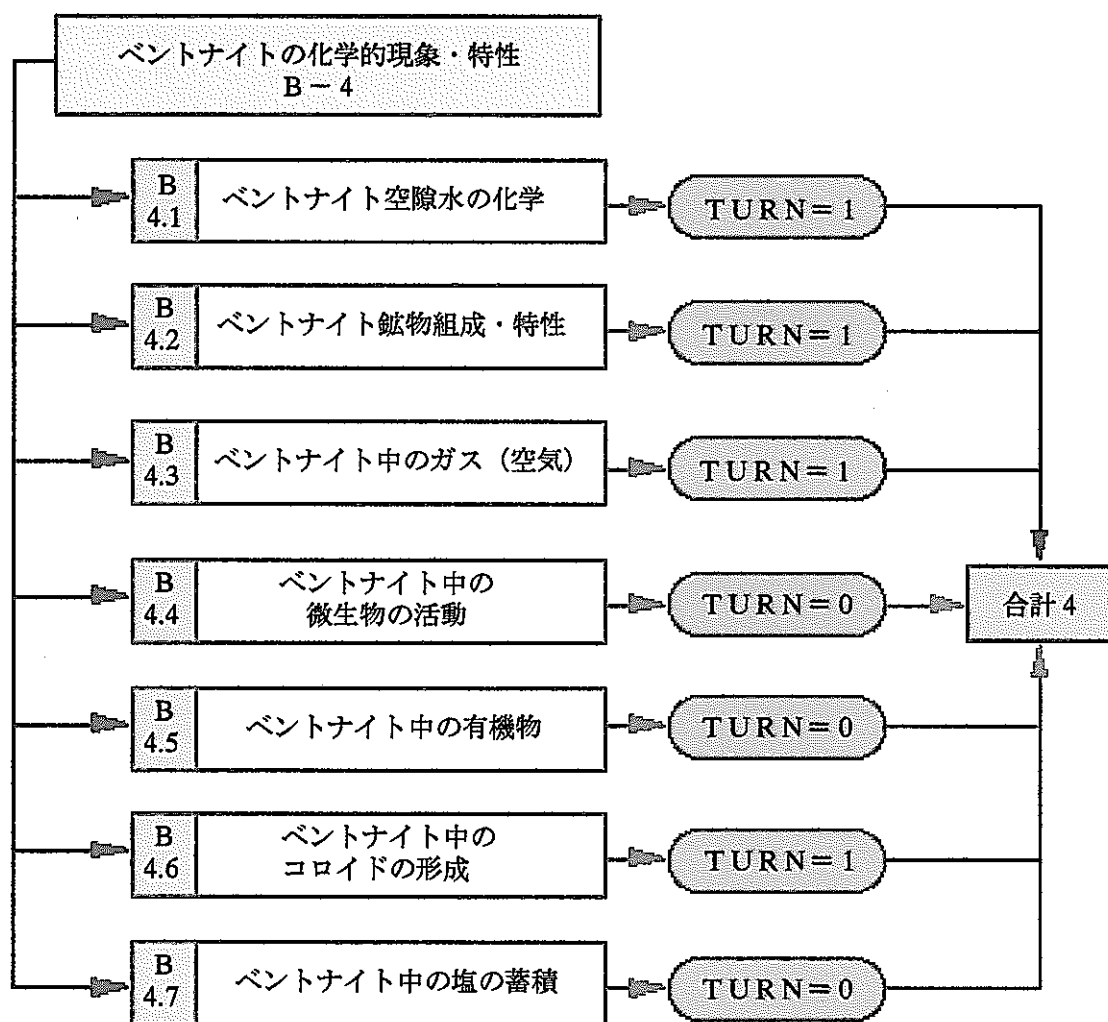


図 2-3 サブ・グループ・ターンの合計概念

現象解析レベルでは、個別シナリオそのものにかかわる評価手法について各種の議論が考えられると予想されたが、本研究は、情報処理技術の応用としてシステムチックな性能評価を目標とした情報シ

システム環境開発技術が目標のため検討対象から除外した。従って、前記のベントナイト中の化学的現象・特性にかかわる評価は、現在のシナリオからは7ターンで評価を完了すると考えた。

それにより、各サブ・グループのシナリオ評価の状態を示す指標を式1)の通りとし、結果をGUI表示して性能評価全体について視覚的に把握するための1指標とした。今後詳細なターン変数表現が必要と考えられる。

$$\text{サブ・グループ・ターン完了値} = \frac{\text{サブ・グループの完了ターン数}}{\text{サブ・グループのターン数} * 100.0} \quad \text{式1)}$$

図2-3にサブ・グループ・ターン完了値の例を示す。この例では、完了値は57%である。この57%の完了値を視覚的に把握しやすいようカラーでGUI上に表示するものとした。

## (2) シナリオ評価GUI

各ターンを実行するためのGUIをシナリオ評価GUIとした。シナリオ評価GUIは、個別シナリオに対応して1ターンの評価を行う作業を支援する機能を提供するものとした。評価を行う作業の支援とは、個別シナリオに記載された評価に関する記述にかかわるモデル、シミュレーション結果、試験結果との検証結果等、シナリオ解析の論拠となる具体的な情報を提供する作業とした。

図2-4に示す通りに、シナリオ評価においては、FEPリストに評価に関する記述を出発点として、モデルの選択、シミュレーションの実施、結果の検証までを対話方式で行うものとした。ターンについては、動作の妥当性を評価することを目的としたため、簡素化して図2-4に示す一連の評価作業の完了をもって1ターンとした。

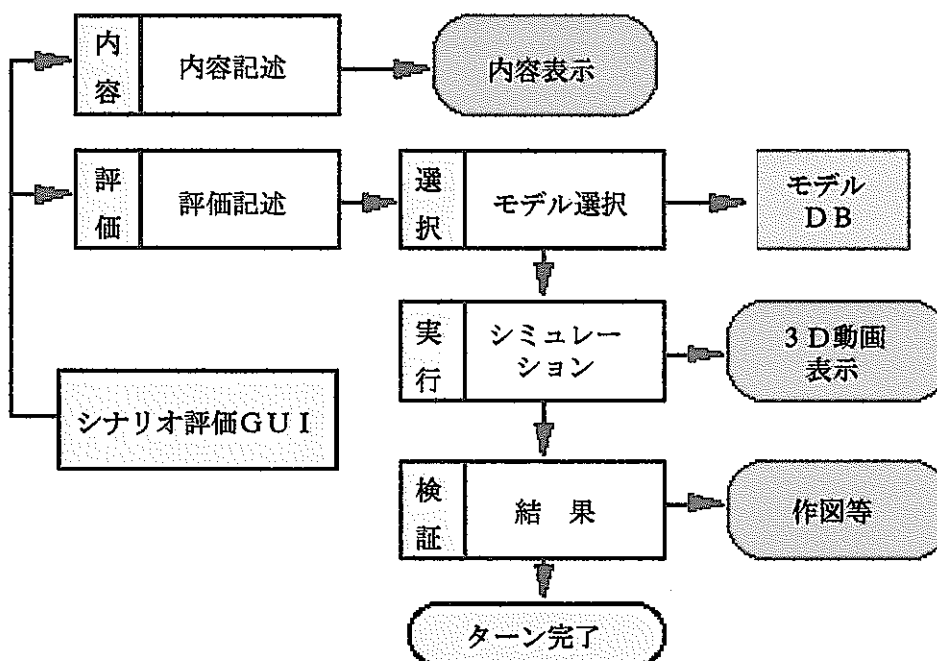


図2-4 シナリオ評価GUI動作機能概念

この点については、今後モデル、検証、試験データ等の多様化を考慮して指標をたてる必要があると考えられる。図2-4から解かるように、その初期の段階では、ターンは従来から研究開発されてきた個別評価コードを利用してシナリオに示される個別評価事象を、数値計算で解析することにより全体を終了する還元論的手法を一元管理するアプローチとしている。

### (3) シミュレーション

シミュレーションは、図2-4に示すシナリオ評価GUIから起動される機能とした。今回の試作では、シミュレーション・コードは個別にデータベース管理されているものとして起動し、計算結果をシステムに共通な3次元動画像表示プログラムにより再現するものとした。この場合のシミュレーションの考え方は、個別シナリオ評価GUIに含まれる機能としている。

また、連成モデルについては、連成するシナリオを関係付け、連成シミュレーションであることを確認できるよう配慮した。これについては、今後詳細な研究が必要である。シミュレーションは、評価シナリオの対象となるオブジェクトが4つのステージについて共通であることから、3Dオブジェクトについて、時刻、及び物理変量という2つの変量次元を加えて5次元オブジェクト（X、Y、Z、t、物理変量）として考えて共通の視覚化サブ・システムに表示するものとした。図2-5に起動時の操作画面を示す。

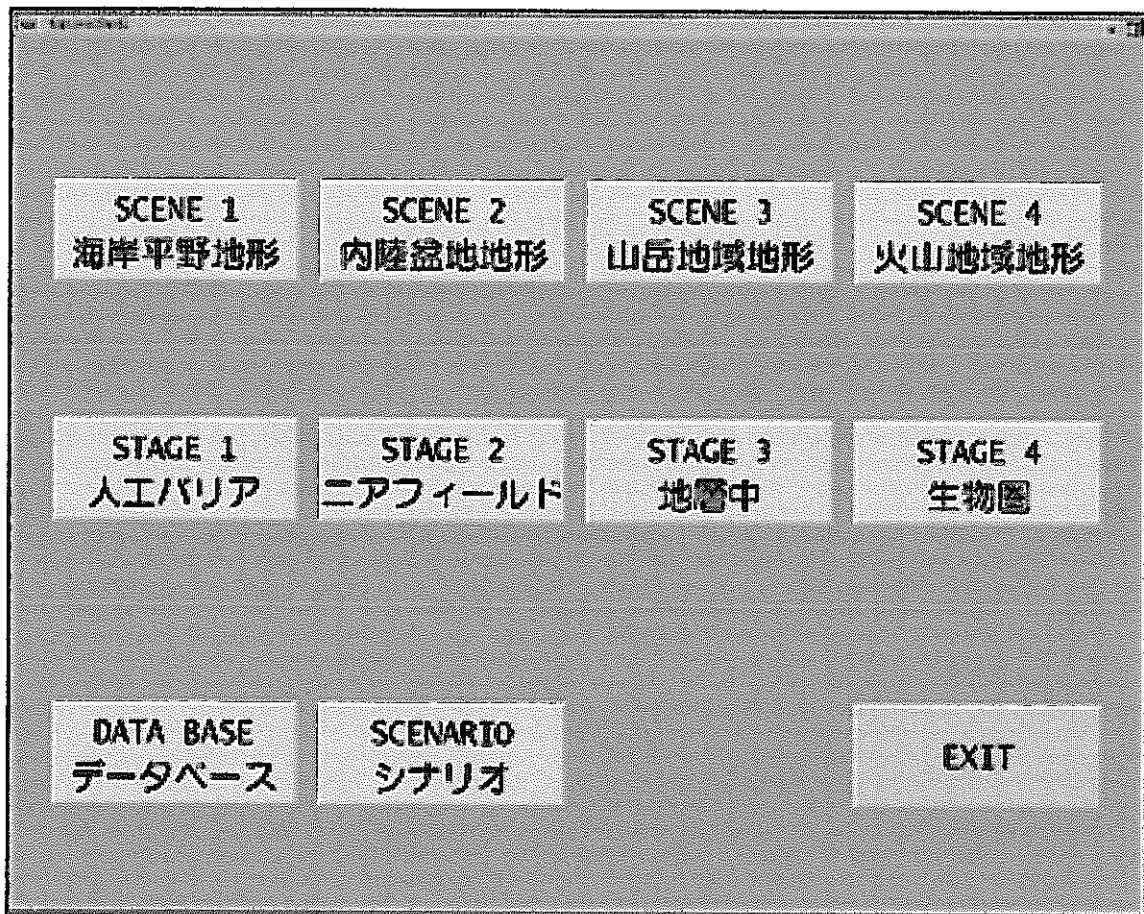


図2-5 初期設定画面の例

図 2-5 に示すように、システムは起動されると全ての画面を初期設定し、主操作画面にステージ等のメニューを表示する。図中の海岸平野、内陸盆地等は、将来のシナリオとして考えられる天然地形に関する情報を格納するデータベースについて選択する操作を行なうために予備的に準備した。図 2-6 にその例を示す。

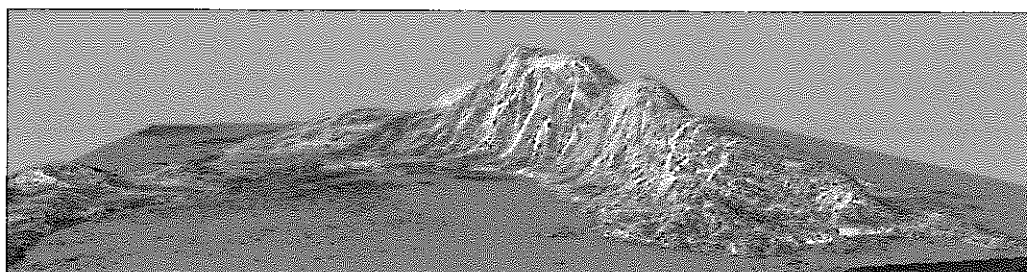


図 2-6 天然地形オブジェクトの例

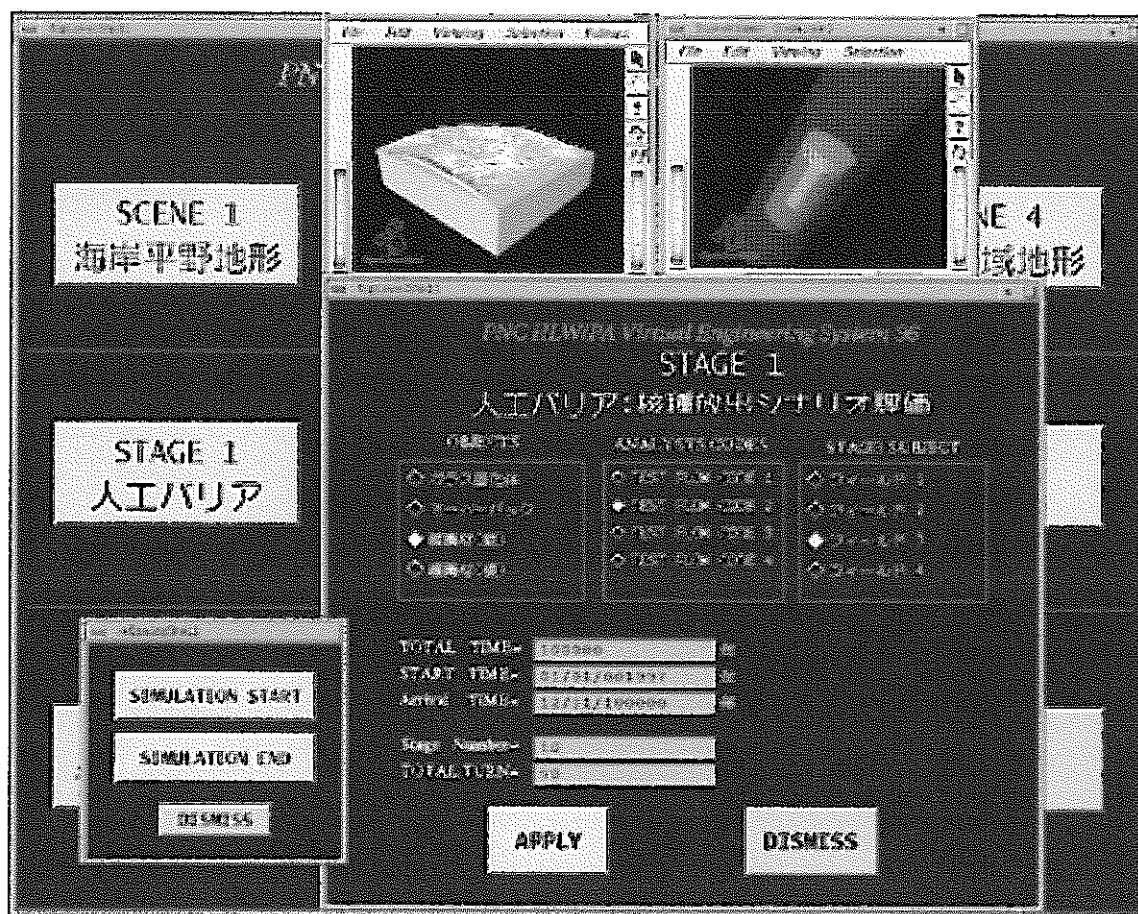


図 2-7 シミュレーション条件選択例

図 2-7 はステージを選択操作した画面の例である。図では、ステージ 1 の人工バリア核種放出シナリオを選択している。図に示すように、ステージを選択すると当該ステージ内の個別シナリオを選

扱することができる。例えば、ガラス固化体の滲出やオーバーパックの腐食等である。このように、ステージ内では、シナリオはより単純な物理化学過程に還元されるものと考えた。

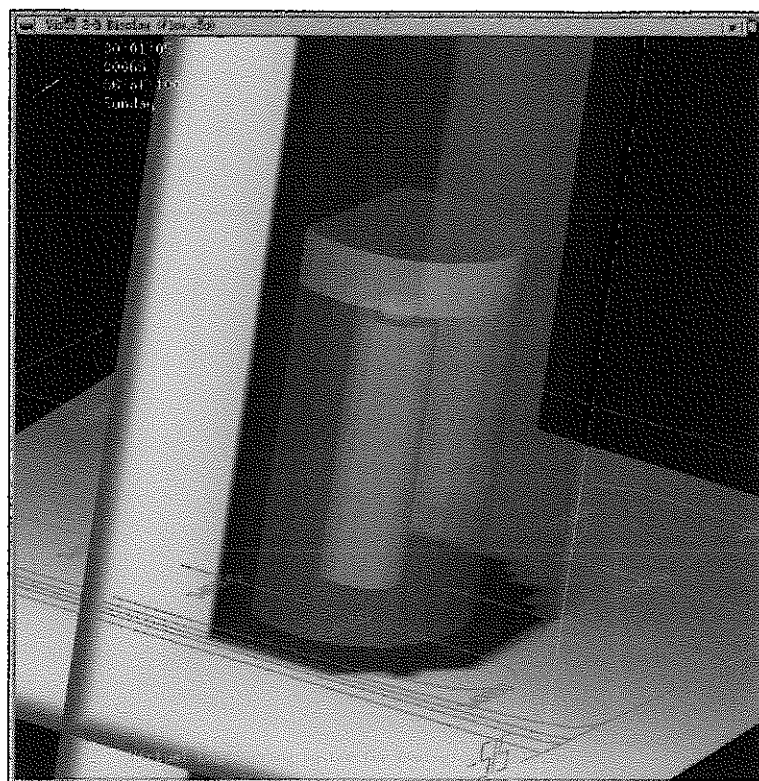


図 2 - 8 3次元シミュレーション動画表現例

図 2 - 8 は、以上のような操作により選定された個別事象の評価計算コードを実行した時点で起動される 3次元の動画化プログラムによる作図表現である。このプログラムは、個別コードから物理化学変量の値と  $x$ 、 $y$ 、 $z$  空間座標の組みを時刻とともに受け取ることにより、3次元で動画表示するプログラムである。これは、動作中でも、拡大・縮小、回転、塗りつぶし、ベクトル表現、等高線表現等を任意の断面について行なえるようにしている。

以上のように、試作した地層処分ヴァーチャル・エンジニアリング・システムの対話機能は、シミュレーションの対象となる問題領域をシナリオに沿って実行し、3次元動画化する機能を提供するものである。

#### (4) ソフトウェア管理GUI

ソフトウェア管理は、地層処分ヴァーチャル・エンジニアリング・システムを開発運用する上で必要となる、多くの試験・検査データ、技術文献、プログラム、解析結果等の情報について、その品質を保守管理することを主目的として開発するシステム運用保守ツールとした。試作においては、次の観点からの情報の管理に大別して実施した。



- 数値データの管理 (例) 熱力学データベース等の試験データ
- 文章データの管理 (例) モデル記述データベース
- 解析コードの管理 (例) プログラム・データベース

図2-9は、地層処分ヴァーチャル・エンジニアリング・システムを構成する情報(ソフトウェア)のインフラ・ストラクチャを示したものである。実際には、このような単純2分岐木構造にはならないが、管理面からは管理の容易さを考え、可能な限り図2-9に示す様な体系になるように情報のインフラ・ストラクチャを構成するものとした。GUIの試作に於ては、図2-9に示す情報構造に向けて、データベース等の構築を進める上で必要となると考えられる技術を利用する為のGUIを試作した。例えば、熱力学データベース分析試験データ等の解析や管理を行う機能、地下水のモデル記述と関連プログラム情報に関する記述等を管理する機能、実際のソースプログラムとその複雑度等の品質管理指標を管理する機能に関するものである。

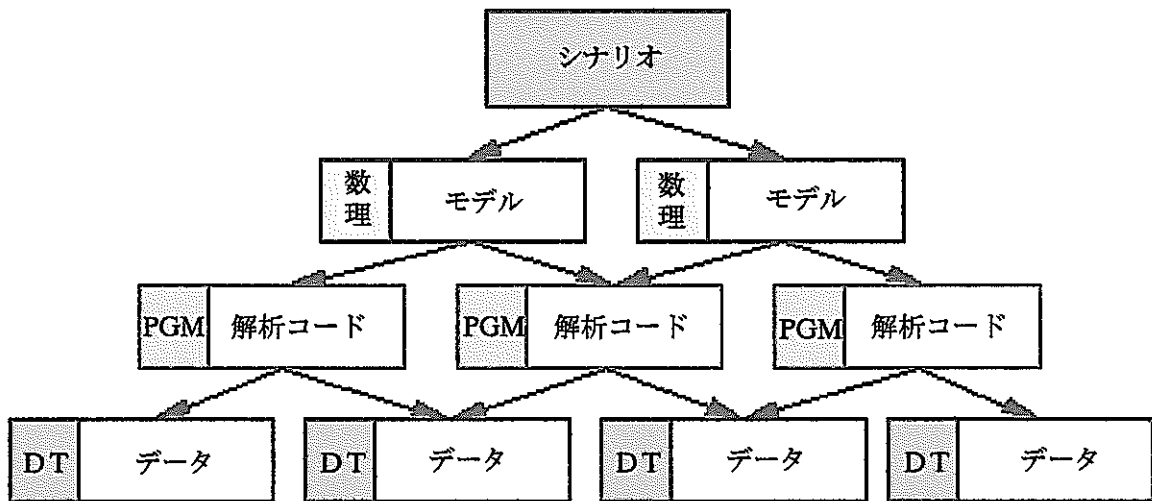


図2-9 情報のインフラ・ストラクチャ

## 2. 1 マルチ・プラットフォームの対話制御機能

従来科学の多くの研究分野で、計算、シミュレートするための新しい方法が発見されるごとに、その分野が進歩するということが繰り返されてきた。例えば、原子力工学、プラズマ物理学、宇宙物理学、分子生物学、地質学等、それらの恩恵を得ている。

計算におけるコンピュータ利用によって、現在科学者が問題を追求し、概念化するスタイルが根本的に変わってきていると思われる。また、計算モデルが、まったく新しい研究領域を開拓しつつもある。実際、科学的な手法のあらゆる変化の中でコンピュータ利用による変化ほど急速なものはないと思われる。

実験から得られるデータはきわめて膨大であり、コンピュータの助けなしには、比較したり結論を引き出したりすることはほとんど不可能であろう。例えば、地層処分性能評価では、膨大な量の地下水についてその地球化学的特性と研究しなければならないが、調査により得られた固有の熱力学データベースやKdデータベースをコンピュータで効率よく利用することにより、予測手法の開発等が可

能になるであろう。また、動画処理等の可視化技術は、これからの放射性廃棄物処分問題等の社会問題を適切に一般大衆や、地域住民に理解してもらうための重要なメディアとなるのである。

地層処分システムの性能評価や、原子力発電所安全性評価、経済やヒト・ゲノムのような複雑系にひそむ問題点は、そのメカニズムについて信頼できるモデルや理論を作るために必要な実験ができないという点である。例えば、原子炉メルトダウン（炉心融解）が起こる限界を知るといった目的で実験することは許されない。同様なことが放射性廃棄物処分システムにもあてはまる。

従来はそのような実験は不可能であったが、現在、この類の複雑系に関する実験が行える条件を備えつつある。また、全体系を理解するために、現実のシステムを細分化し、各部分の知識の統合を目指して、その部分部分を別々に研究する必要もない方向に知見を蓄積する可能性を見いだしつつある。現在、信頼度の高い高品質のコンピュータが使えることから、複雑な現実世界のプロセスを再現する仮想世界を実際にコンピュータ内部に作ることができる。このような仮想世界を実験室として用い、物理や社会とか生物や行動に関する複雑なプロセスについて、発展的な理論を作るための実験が行えることが期待できるのである。

以上のように、コンピュータの進歩は科学の発展を支える重要な基盤となっている。そして、人は、より深くコンピュータと係わることにより、その能力を高めた結果を得ることができる。そのためには、人とコンピュータとの情報交換（対話機能）が期待通りのコンピュータシステムを開発するための重要な技術となっている。

地層処分ヴァーチャル・エンジニアリング・システムは、地層処分された高レベル放射性廃棄物が多重バリア方式により、長期に亘り生活圏より隔離されることをコンピュータを利用しながら確認していく機能を提供することを目的に開発するシステムである。それは、コンピュータ性能の飛躍的な発展に伴い、仮想環境をコンピュータ内部に構築し、考えられるシナリオに添って多方面に亘るシミュレーションを可能にするものとして考えている。そのためには、仮想環境と対話機能に係わる情報処理技術の開発が必要である。それにより人工世界を構築し、幾つものモデルを複雑に組み合わせた複雑系によるシミュレーションを将来実現することを主目標としている。そのようなシステムに於ては、対話機能の質が直接シミュレーションに質に影響するであろう。

試作したシステムは、複数のワークステーションとPCがネットワークに接続され、相互の協調して性能評価シミュレーションを進めるものである。対話システムは、採用するシステム的环境（オペレーティング・システム）により開発技術が異なっている。このようなソフトウェアは、利用者にとって使いやすい反面、ユーザーインターフェイス部分の開発が特に難しい。システムの試作においては、Motifベースの対話システムとWindowsベースの対話システム相互の通信や、各対話システムの背後で個々に独立したタスクとして動作するオブジェクト（以下、このような実行形態とエージェントと記す）の協調等、込み入った機能の開発が必要であった。そのようなシステムをマルチ・エージェント・マルチ・プラットフォーム・ユーザーインターフェイスと呼ぶ。

## 2. 2 システム操作とマルチ・エージェント処理

地層処分ヴァーチャル・エンジニアリング・システムを利用した性能評価作業は、凡そ、以下のような質問にを動機としてなされるであろう。評価にあたっては、試験データやFEP、そしてシミュレーション・コードや視覚化機能等をランダムに操作して、試行錯誤的に作業が進められると考えられる。従って、対話機能は、それらのランダムな操作要求に答えるため、メニューやボタン、スライダ等を装備した、グラフィック・ユーザー・インターフェイスにより提供する必要がある。

- 質問01：地層処分システムは、最終閉鎖が完了するまで、放射能被爆、放射能レベル及び非管理区域への放射性物質の放出が、環境保護機関等の定める環境規制等の許容値以内にいつでも納まるように設計がなされているか？
- 質問02：地層処分システムの地質は、予想される熱応力の影響下で放射性核種移行の抑制能力が増加するような鉱物質に変化したり、またはそのまま変わらずにいる鉱物質を有するか。
- 質問03：地層処分システムの地層は、過去において処分システムが重大な影響を受けるような地震が発生したことがあるか？
- 問題04：地層処分システムの地層は、地下水流れに悪影響を及ぼす隆起、陥没、褶曲、断層のような構造変形があるか？
- 問題05：地層処分システムの地層は、動水勾配、割れ目での平均流速、蓄電率、導伝率、自然再充電、電位差レベル、放電位置のような、人の近づきうる環境への放射性核種移行に影響を及ぼす水理学上の変化の可能性を有するか？
- 質問06：処分区域は、放射性核種の吸着を減少させたり、岩場のもつ強度を弱めたり、人工バリアシステムに悪影響を及ぼしたりする地球化学的過程を有するか？
- 質問07：処分区域は、地下施設や坑道やたて坑の封鎖機構の設計、施行で複雑な工学手法を必要とする岩盤状態や地下水状態を有しているか？
- 質問08：処分区域は永久閉鎖中、安定な地下開孔部を設計できないような地盤力学的特性を有するか？
- 質問09：地層処分システムの方角、形態、レイアウト、深さや地下施設の一部である人工バリアは放射性核種の格納や隔離に有効か？
- 質問10：地層処分システム中の通路は、作業が安全に実施され回収ができるように設計されているか？
- 質問11：地層処分システム中の通路の掘削方法は、人の近づきうる環境へ地下水や放射性廃棄物が移行する優先通路の発生を抑えるような方法か？
- 質問12：地層処分システムは、水やガスの侵入を制限できるように設計されているか？
- 質問13：地層処分システムは、最大勢負荷、破碎帯やその他の不連続の条件、及び母岩、周囲の地層、及び地下水の熱輸送特性、地盤力学的、帯本地質学的、地球化学的システムの予想応答を考慮して目標を満足するように設計されているか？

以上の質問は、予想される質問のほんの一部（のシナリオ）であると考えられる。計画されている地層処分システムの性能評価シミュレーションのための技術は、このように非常に多岐に亙る専門領域の問題が複雑に組み合わさったものであり、特にランダムな試行錯誤的問題解決に陥りやすいと考えられる。地層処分システムの性能評価シミュレーションには、およそ3種類のソフトウェア・モデ

ルが必要であると考えられる。

第1は、処分システム施設および地表の景観の物理的な記述、処分システムに処分される廃棄体の位置、種類および量に関する数値的な記述、それにシミュレーション対象地域の地層の特性記述等からなる物理モデルである。

第2は、廃棄物がら漏れ出た放射性核種が、処分場近傍、及び地層中のそれぞれの場に及ぼす効果を予測する汚染物質移行モデルである。

第3は、地震や火山の噴火など、サイトの通常の操業にとっては外的要因である気候変動等の効果を計算に入れるための一連のモデルである。

それぞれの種類のモデルは、空間的、時間的に幅広いスケールで利用されるものでなければならない。また、空間的には、廃棄体スケール、処分ピット周辺スケール、処分場全体スケール、地層スケールの4つのスケールを考えると良いであろう。

シミュレーションの時間区分に関しては、短期間の定義として100年という期間がしばしば使われるが、これは制度的な規制をそれよりも長い期間に亘って保証することはできないと考えるためである。長期間については100年～1万年の期間を想定することができる。これは、これまでに行われた研究によって、放射性核種、および環境上重要な意味を持つ微量元素の漏出の大部分は1万年以内に起こり、1万年以降は気候学的変動が廃棄物管理工学よりも大きな意味を持つことになる可能性が高いと考えられているためである。

### (1) 処分システムの情報

処分システムの情報には、処分システムの物理的構造、および処分システム内の廃棄物の分布を記述するのに必要なすべての情報が含まれる。物理的構造の記述には、処分システム内のすべての廃棄体、ボアホール、立坑、通路および室の位置ならびに寸法が含まれている必要がある。また、地表の景観情報も含まれている必要がある。

### (2) 地質学的情報

地質学的情報は、シミュレーション対象地域の地層の特性を記述するものである。このモデルには、地山の地層の他、その地域の鉱床や人工的な構造物の位置や規模についてのデータが含まれている必要がある。地山の地層の特徴は、厚さ、深さ、水平方向の広がり、熱伝導率、粒間地下水流、収着の性質、地山と地山に含まれる地下水、および処分した廃棄体のパッケージの間での化学的相互作用の程度などのパラメータに関して記述される必要がある。

### (3) 物質移行情報（性能評価情報）

汚染情報は、処分システムからの漏出核種の拡散を予測するために使われる情報である。それは、主に個別の物理化学現象仮定の記述をモデルにより開発されたシミュレーション・コードと関連するデータである。この種のモデルは、処分システムのために岩盤、およびその間隙流体に生じる外乱による水文学的、熱・力学的な力の間での相互作用を考慮に入れたものでなければならない。

各種の力の間における相互作用の内容を理解するためには、まず熱的な力の発生源から行うことが一般的操作として考えられる。熱的な力は、処分される廃棄体から発生する大量の熱によって生じる。

この熱入力は、相当の期間続き、地下水の対流ならびに母岩の膨張を引き起こす。比較的わずかな温度上昇が大きな体積で長期にわたって起きた場合には、かなりの対流を生じることになる可能性がある。また、処分システムの建設においては、地盤に大きな空洞を作ることによって当初の応力分布を変えることになる。この応力分布の変化によって、母岩に力学的な変形が生じ、既存の割れ目が開いたり、閉じたりする現象が考えられる。このような割れ目パターンの変化は、流体の流れや地盤中の溶質の輸送に大きな影響を与える。

更に、廃棄物の拡散ならびに関係する水文学的、熱力学的な相互作用についての的確な予測をするためには、人工バリアと天然バリアの両方の空間的スケールを対象とした別々のモデルが必要となる。

#### ア. 人工バリア

人工バリアは、次の3つの事項に関する廃棄物パッケージの達成能力のシミュレーションを行うものである。

- ・長期にわたって放射性核種を封じ込めること
- ・地山への放射性核種の漏出速度を低減すること
- ・廃棄体への水の接近を最小限に抑えること

人工バリアには、処分場の廃棄物処分孔に収容するすべてのもの（すなわち、固化体、キャニスタ、オーバパック、緩衝材、埋戻し材等）を含めるものとする。人工バリアは、オーバパック設計寿命、当初からの欠陥によるオーバパックの早期破損率、破損したオーバパックからの部分浸出率、オーバパックの破損から浸出期間の最後までまでの時間などのパラメータを使ってオーバパックの物理的性質の特徴を記述する。

#### イ. 天然バリア

天然バリアに対しては、地下水と人工バリアの間の相互作用のほか、それに続いて核種の地下水中への移行が起きる場合の様子をモデル化することが要求される。天然バリアは、対象とする区域全域にわたって流れの経路と流速を特定するとともに、処分場に関係する水文学的、熱的および、力学的な力の相互作用を考慮に入れたものでなければならない。天然バリアで特に必要となるパラメータとしては、以下が考えられる。

- ・処分システムの建設によって生じる岩盤の機械的变化
- ・処分する廃棄物による熱負荷の増大
- ・天然の動水勾配および透水率
- ・地下水流の流速および水量
- ・地下水のpH、酸化電位、イオン強度、含有酸化剤、処分した廃棄物が関係する化学的な変化などの化学的性質。

#### ウ. 生物圏情報

生物圏事象モデルは、処分システムの通常の操業にとっては外的な要因である。それは生物

圏における事象の影響をシミュレーションするモデルである。例えば、地震、火山の噴火、大洪水、隕石の衝突、地図にない大きな割れ目、ボアホールを塞いだ栓の破損の他、爆弾の爆破、火災または処分サイトへの侵入などの人為的な事象を挙げるができる。

#### (4) 基本対話機能要件

地層処分ヴァーチャル・エンジニアリング・システムの操作においては、利用目的に応じて主として2つの形式があると考えられる。第1に、多数の観衆にシミュレーション画像が提供され、観衆がフロアから「～ならばどうなる」式の質問を行いながらシミュレーションの経過の規定に参加する形式がある。第2の形式として、性能評価研究のための利用があり、研究者が現象の局所のおよび広域的な影響について評価する場合である。

これらの利用方法では、シミュレーション結果を視覚的な直観に訴える単純化された描写にまとめるコンピュータ・グラフィックスによって、対話方式で自由に操作できる大型のビジュアルな環境をサポートすることができるとともに、結果の詳細な分析と科学的な視覚化によって、あらかじめプログラムされた複雑なシミュレーション・シナリオの実行をサポートすることができるシステムのアーキテクチャ開発が必要である。

個別シナリオ等の評価においては、シミュレーションは場合により数時間に渡ることも考えられるため、必ずしも完全な対話速度で動作する必要はないと思われる。例えば、小スケールの結果を高い精度で得ることに研究者の関心がある場合は、シミュレーションは対話速度よりも、ゆっくりと動作して詳細な計算結果を作り出してもよく、その上で、その結果に対して研究者が対話方式で質問を与えるようにすることができるようにすることが必要である。これをアーカイブ処理と言う。

将来、テラフロップス級の超高速コンピュータの利用により、全面的な対話方式シミュレーションが可能なシステムを作り上げることができるのであれば、このような動作機能は不要と思われるかも知れないが、実際にはそうではない理由として互いに関連し合った2つの要因がある。

第1の要因は、モデルのデータ空間のイメージを捉えるのに必要な大量のデータを記録することは、たとえ95%以上のデータ圧縮ができたとしても、それによって記録されるシミュレーションの規模（すなわち持続時間）に厳しい制約が課される可能性があるということである。従って、実時間性が保証されないであろう。

第2の要因は、たとえデータ空間のイメージを保持するのに十分な2次記憶装置があったとしても、そのような装置の入出力（I/O）速度の遅さのために、2次記憶装置への転送がシステム性能における重大なボトルネックとなるということである。これも前記と同様に、実時間性を保証しない要因である。

また、シナリオに沿った地層処分の性能評価問題に係わる詳細なシミュレーションを行う場合は、シミュレーション・システムが「実時間」システム、すなわち外部からの割り込み、ないし、イベントに対してマイクロ秒、ないしミリ秒の時間枠で応答するシステムとして動作する必要はないと考えられる。何故ならば、そのような問題はシナリオの評価が目的なためである。従って、対話機能は、目立つ休止のない迅速な動作が要求されるが、本来的な要求としての「実時間」要件は存在しない。このように考えることによって、シミュレーション・システムのハードウェア、およびソフトウェアの「実時間」特性の指定にかかわる問題は、システム設計面から回避することができ、

より現実的なシステムの実現が可能となる。

目標とするシステムは、高度な対話機能を通じて複数の独立したプログラムが、場合により協調して要求に答えるための処理を行うシステムとなると考えられる。そのようなシステムの動作は、従来の逐次的にコードを組み合わせて計算する手法と本質的に異なったものであると考えられる。以上のようなシミュレーション手法（数値実験）は、複雑な現象の理解にとって非常に大切なものと考えられる。数値実験では、コンピュータは素過程として論理的に動いている。しかし、コンピュータ内で起こる現象を理解したり、定式化したりするとき必要になることは、それら論理の筋をすべて追うことではなく、コンピュータ内の現象を通して養われた新しい直観に基づくことであると考えたほうが、より自然なアプローチであろう。

このとき、最初に重視すべきことは、素過程での論理の正当性である。これを信じた後、今までの物理学的直観とは異なるかもしれない新しい直観を養う必要がある。たとえ、ある事実が我々の物理的直観に反したとしても、このような意味で「数学的事実」であれば、それを真理とみなすことは計算科学的方法と考えられる。

以上のように、ヴァーチャル・エンジニアリングの思想から、地層処分性能評価のような複雑な現象に対する数学的定式化をコンピュータの中で形式化する（コンピュータ実験）ことが期待できる。そのようなシミュレーションは、複数の独立したタスクとして動作するオブジェクトである“エージェント”の協調により実現すると考えることができる。従って、試作した対話システムは、マルチ・エージェント（かつ、マルチ・プラットフォーム）な環境に対応するものとした。

## 2. 3 マルチ・エージェントな対話機能

理想的な地層処分ヴァーチャル・エンジニアリング・システムの対話機能の基礎的考え方として、一個のユーザーインターフェイスを複数のエージェントの集まりとして構成する、マルチ・エージェント・ユーザーインターフェイスとして考え、そのアーキテクチャを試作した。マルチ・エージェント・ユーザーインターフェイスは、個々のユーザーインターフェイス・エージェントが単独で動作し、ユーザとの対話が可能であるユーザーインターフェイスとした。マルチ・エージェントを用いる利点は、各ユーザーインターフェイス・エージェントを個別にプログラミングできること、テストできることである。ユーザーインターフェイス・エージェントの協調は二通り必要である。

協調 1：複数のユーザーインターフェイス・エージェントが一体となって、一個のユーザーインターフェイスを構成するには、それらが協調する必要がある。

協調 2：複数のユーザーインターフェイス・エージェントが同一のものに対するユーザーインターフェイスを提供しているとき、それらの表示の一貫性を保つためにも、協調が必要である。

### (1) マルチ・エージェントモデル

ユーザーインターフェイス管理システム（ウインドウ・マネージャと呼ばれる事もある）とは、ユーザーインターフェイスを構築するためのソフトウェアシステムである。ユーザーインターフェイスの開発においては、このユーザーインターフェイス管理システムの機能を検討することが重要な課題となる。

ユーザーインターフェイス管理システムは、ウインドウシステムの上位に位置し、ウインドウシス

テムの提供する入出力機能と、アプリケーションプログラム本体との間の橋渡しを行なうシステム・サービス機能である。ユーザーインターフェイス管理システムを用いて構築されたユーザーインターフェイス部とアプリケーション本体とが組合わさって、ウインドウシステム上で動作するユーザーインターフェイス全体を構成する。

ユーザーインターフェイス管理システムに実装する制御アルゴリズムに関する有力な考え方として、シーハイムモデルがある（図2. 3-1参照）。シーハイムモデルでは、対話部はプレゼンテーション部、対話制御部、アプリケーション・インターフェイスの三つの要素から構成される。各要素はコンパイラにおける字句解析部、構文解析部、意味解析部にそれぞれ対応する。ユーザーの入力は、三つの構成要素によって、順に解析され計算部への入力となる。逆に計算部の出力は逆順で三つの構成要素によって処理され、ユーザへの出力となる。シーハイムモデルの問題点の一つは、ダイレクト・マニピュレーションが実現困難なことである。

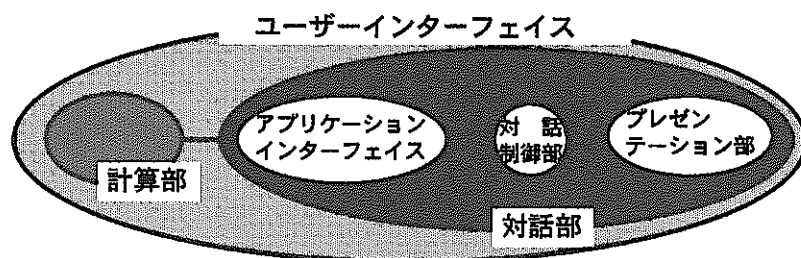


図2. 3-1 シーハイムモデル

ダイレクト・マニピュレーションを行なうには、対話部の末端であるプレゼンテーション部を管理する情報から、計算部の管理する情報まで、全ての構成要素の情報を必要とする。プレゼンテーション部と計算部との間で、何度も制御を行き来させれば不可能ではないが、現実的には実現困難である。

例えば、グラフィックス・エディタにおける、ラバーバンディングはプレゼンテーション部単独で実現することはできず、計算部の助けによって、ラバーに囲まれた範囲の図形を特定させなければならない。同様に、視覚しながら計算を進めるにはマウス・カーソルの情報をプレゼンテーション部が取り込むとともに、計算部にコールバックをかけて結果を待たなければならない。

そこで同期問題が生じる。同じことは、性能評価に係わる個別コードを、インフルエンシ等に関連させて実行する場合も言える。マルチ・エージェントモデルは、シーハイムモデルへの反省から提案された。機能の層別化によってユーザーインターフェイスを分解するシーハイムモデルと異なり、マルチ・エージェントモデルでは、画面領域の分担によってユーザーインターフェイスを分割する。

マルチ・エージェントモデルでは、一個のユーザーインターフェイスの占める画面領域は、いくつかの小領域（例：テキストサブウインドウ）に分割される。各ユーザーインターフェイス・エージェントは、小領域の一つにおける対話操作を担当する。個々のユーザーインターフェイス・エージェントは単独で動作しユーザとの対話が可能である。これらのユーザーインターフェイス・エージェントが集まって領域全体での対話操作を提供する。



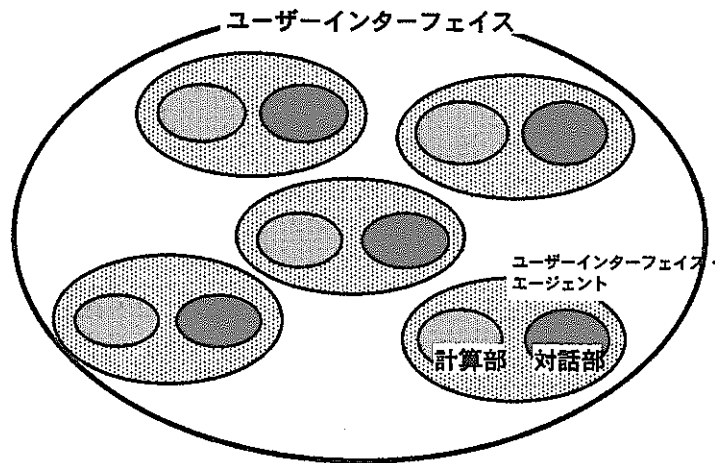


図 2. 3-2 マルチエージェントモデル

直観的にいうと、シーハイムモデルが横割りであり、マルチ・エージェントモデルは縦割りである。個々のユーザーインターフェイス・エージェントは、さらに対話部と計算部に分割される。対話部は、そのユーザーインターフェイス・エージェントの担当領域における対話操作を提供する部分である。計算部はアプリケーション全体の中の一部であって、そのユーザーインターフェイス・エージェントが操作すべき部分である。

このように、対話部と計算部を組みにしたエージェントを考えることにより、従来型の同期関係に支配されたシミュレーションから、並列的に協調動作するマルチ・エージェントによる複雑な協調計算を行うアーキテクチャを実装することが期待できる。すなわち、ユーザーインターフェイス全体の計算部（アプリケーション全体）も、ユーザーインターフェイス・エージェントの計算部の集まりとみなすことができる。マルチ・エージェントモデルには次のような利点がある。

- 利点 1 : 個々のユーザーインターフェイス・エージェントを独立に作成できる  
個々のユーザーインターフェイス・エージェントを独立にプログラミングすること、対話操作を行なってテストすることができる。
- 利点 2 : ユーザーインターフェイス全体の構成を柔軟に変えられる  
ユーザーインターフェイス・エージェントは他の部分と独立しているため、ユーザーインターフェイス・エージェントの追加や削除が容易である。
- 利点 3 : ダイレクトマニピュレーションが容易に実現できる  
対話部と計算部の分離は、シーハイムモデルのように全体で一括して行なわれるのではなく、ユーザーインターフェイス・エージェント単位で無理なく行なわれる。あるユーザーインターフェイス・エージェントがダイレクトマニピュレーションを実現するには、そのユーザーインターフェイス・エージェントの対話部と計算部の情報を用いなければならない。
- 利点 4 : マルチスレッド・ダイアログが容易に実現できる  
マルチスレッド・ダイアログとは、ある系列の操作の途中で、他の系列の操作を実行することである。例えば、文書を格納するファイル名をエディタで指定している途中で、他のウィンドウでファイル名の一覧を表示する操作を実行することである。

マルチ・エージェント・モデルでは、あるユーザーインターフェイス・エージェントが、一つの系列の操作を行なっている時、並列に他のユーザーインターフェイス・エージェントにもう一つの系列の操作を実行させることによって、マルチスレッド・ダイアログを自然に実現することができる。ただし、ユーザーインターフェイス・エージェントを並列に動作させるためには、並列オブジェクト指向での実装を要する。

## (2) ユーザーインターフェイス・エージェントの協調

ユーザーインターフェイス・エージェントが、それぞれ独立して動作するだけでは十分ではなく、それらが協調して動作する必要がある。本研究では、必要となる協調を一体化のための協調と一貫性のための協調の二つに分けて考える。一体化のための協調とは、複数のユーザーインターフェイス・エージェントが一体となって一個のユーザーインターフェイスとして振舞うための協調である。

例えばスクロールバー（ユーザーインターフェイス・エージェント）とテキストウィンドウ（ユーザーインターフェイス・エージェント）からなるユーザーインターフェイスにおいては、スクロールバーでスクロールを行なうとテキストウィンドウでの表示範囲が変わる。

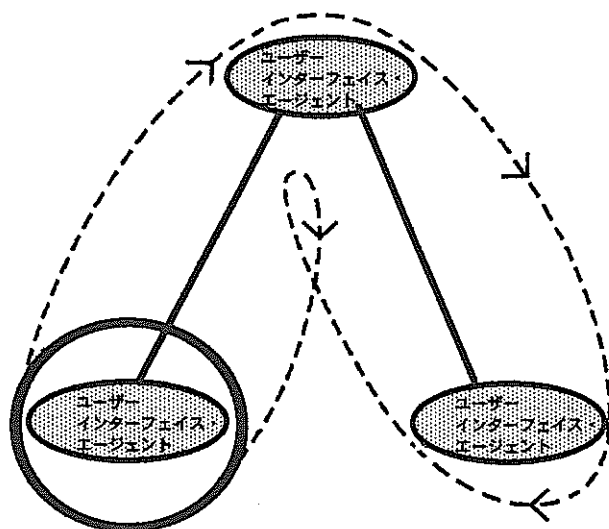


図 2. 3-3 階層構造による協調

一貫性のための協調とは、複数のユーザーインターフェイス・エージェントが、同一のものに対するユーザーインターフェイスを提供しているとき、それらの表示を一貫させるための協調である。すなわち、一個の計算部を共有する複数のユーザーインターフェイス・エージェントは協調する必要がある。例えば、同一のテキストに対して二つのテキストウィンドウが存在するとき、一方で編集を行なうと、もう一方でも表示を行なわなければならない。

同様に、例えば地球化学コードPHREEQEの熱力学データベースを整理する対話機能設計を考えた場合、PHREEQEは検証されたものを一つに計算部として扱い、同一の熱力学データベースについて二つ以上のウィンドウを表示しながらデータ解析作業がすすめられることが考えられる。いずれ

にしろ、このような一貫性のためのエージェントの協調機能は、今後システム開発の様々な局面で生じるであろう。マルチ・エージェントを用いた従来の研究では、この二つの協調を実現するのに、階層構造による方法か逆階層構造による方法のいずれか一方のみを用いている。

### (3) 階層構造による実現方法

階層構造による方法では、ユーザーインターフェイス・エージェントを階層的に構成し、上位のユーザーインターフェイス・エージェントに、下位のユーザーインターフェイス・エージェントを制御させる(図2. 3-3参照)。

この方法では、一体化のための協調を容易に実現することができる。下位のユーザーインターフェイス・エージェントの挙動のうち協調を必要とするものは、上位のユーザーインターフェイス・エージェントによって検出される。上位のユーザーインターフェイス・エージェントは、関連するユーザーインターフェイス・エージェントに通知し協調を行なわせる。

例えば、スクロールバー(ユーザーインターフェイス・エージェント)とテキストウインドウ(ユーザーインターフェイス・エージェント)を持つファイルエディタ(上位ユーザーインターフェイス・エージェント)を考える。スクロールバーの挙動は、ファイルエディタによってテキストウインドウに通知されスクロールを引き起こす。

各ユーザーインターフェイス・エージェントは、上位ユーザーインターフェイス・エージェントや、兄弟ユーザーインターフェイス・エージェントに依存しない。上位や兄弟がすべて削除されても、そのユーザーインターフェイス・エージェントは単独で動作しうる。言い換えると、ユーザーインターフェイス・エージェントは、上位ユーザーインターフェイス・エージェントや兄弟ユーザーインターフェイス・エージェントからの入力を必要とせず、これらに対する出力も行なわない。上述の例では、テキストウインドウ(兄弟ユーザーインターフェイス・エージェント)とファイルエディタ(上位ユーザーインターフェイス・エージェント)が削除されても、スクロールバー(ユーザーインターフェイス・エージェント)が単独で動作が可能である。

階層構造による方法の欠点は、一貫性のための協調に向かないことである。一貫性を保つべき複数のユーザーインターフェイス・エージェントは、一個の計算部を共有することによってのみ関連づけられている。したがって、協調を制御すべき上位のユーザーインターフェイス・エージェントは存在しない。一貫性の保持の目的で、上位のユーザーインターフェイス・エージェントを設けることによって、この問題を解決することは可能ではある。この上位ユーザーインターフェイス・エージェントは、一貫性を保つべきユーザーインターフェイス・エージェントすべての上位に位置付けられる。

いずれかのユーザーインターフェイス・エージェントが、計算部の状態を変更するような挙動を行なうと、上位ユーザーインターフェイス・エージェントが、他のユーザーインターフェイス・エージェントに通知して一貫性を保たせる。

しかし、この解決方法で一貫性を保つためには、どのようなユーザーインターフェイス・エージェントにより、どんな操作が行なわれるのかを予め想定しておく必要がある。したがって、新たなユーザーインターフェイス・エージェントが追加されるたびに、一貫性を保たせるための変更を上位ユーザーインターフェイス・エージェントに行わなければならない。この作業は煩雑であるため、実際には行なわれないことが多い。また、ユーザーインターフェイス・エージェント以外のプログラムによ

って計算部の状態が変更されると一貫性は保たれない。

#### (4) 逆階層構造による実現方法

逆階層構造による方法では、計算部を共有する複数のユーザーインターフェイス・エージェントの協調を行なう。逆階層構造とは、階層構造の逆転したものであり、共有される計算部と、各ユーザーインターフェイス・エージェントの対話部とが形作る構造である（図2. 3-4参照）。

この方法では、一貫性のための協調を容易に実現することができる。あるユーザーインターフェイス・エージェントの対話部がユーザとの対話の結果、計算部の状態を変更すると、どのユーザーインターフェイス・エージェントの対話部もこの変更を検出して再表示を行なう。

例えば、一個のテキスト（計算部）を共有する複数のテキストウインドウ（対話部）を考える。あるテキストウインドウからテキストを編集すると、全てのテキストウインドウがこの変化を検出して再表示を行なう。

この方法の欠点は、一体化のための協調に向かないことである。一体となって振舞うべき複数のユーザーインターフェイス・エージェントは、上位のユーザーインターフェイス・エージェントを構成することによってのみ関連づけられている。協調の中心となるべき共有される計算部は存在しない。

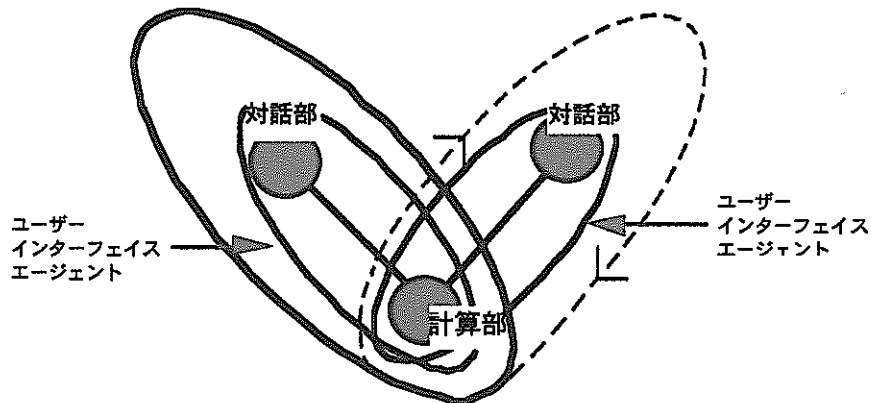


図2. 3-4 逆階層構造による協調

一つの解決方法は、全てのユーザーインターフェイス・エージェントの状態を表すデータを設け、各ユーザーインターフェイス・エージェントの計算部として共有させることである。その結果、各ユーザーインターフェイス・エージェントの挙動は、この計算部の状態変化として現れる。他のユーザーインターフェイス・エージェントは状態変化を検出し、必要な協調を行なう。この計算部は、アプリケーションの一部ではなく、他ユーザーインターフェイス・エージェントについての情報であるので、厳密には計算部とはいえない。この解決方法では、ユーザーインターフェイス・エージェントの独立性が損なわれる。実際、各ユーザーインターフェイス・エージェントの計算部が、他ユーザーインターフェイス・エージェントの情報を多く含んでいる。

現実には、逆階層構造方式と階層構造方式を組み合わせたアーキテクチャをプログラミングすることができ、以下のような長所を得ることができる。

**長所1：複雑なユーザーインターフェイスの一貫性を保つことができる。**

今回作成したブラウザは、同一のシステム環境に複数存在しても、完全に一貫性を保つ。計算部を共有する、ユーザーインターフェイスの一貫性をとる例は、従来いくつか報告されているが、ほとんどは、テキストなどの簡単な計算部をもつユーザーインターフェイスに限られている。Smaltalkのクラス/メソッド定義のように複雑な計算部をもち、ブラウザのように複雑な機能をもつユーザーインターフェイスにおいて一貫性が実現された例はなかったと思われる。

**長所2：想定したユーザーインターフェイス以外のものによって、計算部の状態が変更されたときも、ユーザーインターフェイスは一貫性を保つことができる。**

今回作成したブラウザが開かれている状態で、ファイルからクラス定義を読み込むと、ブラウザの表示は自動的に更新される。各ユーザーインターフェイス・エージェントは、クラス定義の変更を検出するのであり、他のユーザーインターフェイス・エージェントには、依存しないのがその理由である。ブラウザの作成時には、ファイルからのクラス定義の読み込みを考慮していなかったにもかかわらず、一貫性は保たれている。

**長所3：各ユーザーインターフェイス・エージェントを独立に作成し実行できる。**

この利点はマルチ・エージェントモデル自体のものである。今回のブラウザの例では、クラス・リスト・ビューなどのサブビューをそれぞれ完全に独立して作成し、動作確認を行なうことができた。従来のブラウザの構成では、常にブラウザ全体の動作を考慮しなければ、作成することも動作確認を行なうこともできない。

**長所4：各ユーザーインターフェイス・エージェントの記述が適度な大きさである。**

記述はユーザーインターフェイス・エージェントに分散して行なわれ、各ユーザーインターフェイス・エージェントの記述量は数百行程度である。ユーザーインターフェイス全体を理解するには、これらのユーザーインターフェイス・エージェントの記述をそれぞれ独立に理解すればよい。ほとんど全ての記述が、クラスブラウザに集中していた従来の実装と比べ、プログラムの理解が大きく容易になる。

そのような協調計算における逆階層構造と階層構造の手法を導入するには、複数のサブシステムが何らかの形で相互作用を及ぼしながら、計算を進めて行くシステムの構成をとり、凡そ以下の2通りの設計手法が考えられる。

**手法1：元来1つの問題を複数のサブシステムへと分割し、作業の分散化を図るタイプ。トップダウン的な設計が行なわれるため、どのようなサブシステムによってシステムが構成されるかは固定されている。また、各サブシステムの作業結果を元々の問題に対する作業結果へと統合する必要があるため、すべてのサブシステムの運用、制御を司りシステム全体の整合性を実現する集中管理ノードの存在が前提となる。**

**手法2：基本的に独立した処理を行う自律的なエージェントが、必要に応じた情報のやり取りを行い、それによってシステム全体に協調的動作を実現しようとするタイプ。通常、すべての処理を一元的に管理するエージェントは存在せず、システムを構成するエージェントも予め特定することはできない。**

このように、手法1のタイプにおいては、本来1つのシステムで解かれている問題をいかに複数のサブシステムへと分割するかが、その興味を中心であるのに対して、手法2のタイプでは複数のエージェントをいかに協調させるかに重点が置かれる。従って、システムの動作としては、ともに複数のサブシステムで並列（並行的）に処理が行われるのであるが、問題解決の方向性は逆である。

手法1のタイプは、いわゆる並列型システムと呼ばれるものである。このタイプは、集中管理ノードの存在によって、スケジューリング、および解の最適化が期待できるシステム構成である。しかし一方では、新たなサブシステムをシステムに追加するような場合に、集中管理ノードの変更も同時に必要となることから柔軟性、拡張性の点では不利である。手法1のタイプのシステム構成に対して与えるべきシステムの構造は、本質的に階層構造である。ある1つの集中管理ノードは、その管理下にある複数のサブシステムに対し上位の階層に位置し、そのような集中管理ノードをさらに集中管理するノードはその上位に位置する。このようにして階層構造が成立する。この構造によって手法1のタイプのシステム構成に対して、確実な管理、制御が可能なシステムが実現できる。本研究では、1つのユーザーインターフェイスを、いくつかのユーザーインターフェイス・エージェントへと分割した。この分割は手法1のタイプのシステム構成である。したがって、一個のユーザーインターフェイスをまとめる必要が生じ、集中管理を行なう全体ユーザーインターフェイス・エージェントを新たに作成することによって、階層構造による統合化のための連動を行なった。しかし、集中管理は通常の集中管理とは異なる。通常の集中管理では、サブシステムは上位に隷属する。すなわち、入力上位から与えられ結果も上位に対して出力する。これに対し本研究の集中管理では、サブシステムは上位から独立である。ユーザーインターフェイス・エージェントは、上位ユーザーインターフェイス・エージェントとの間の入出力のやりとりを必要としない。上位は必要に応じて介入するのみである。すなわち、各ユーザーインターフェイス・エージェントが、上位ユーザーインターフェイス・エージェントに全く依存しない。これは、マルチ・エージェントの指向性に沿った協調型の階層構造であるといえる。

一方、手法2のタイプは分散（協調）型システムと呼ばれるものである。このタイプは、自律的エージェントの相互作用によって、システム全体の協調が実現されるため、柔軟性、拡張性に優れる。しかし、集中管理エージェントが存在しないため、最適な制御を行なうことは困難である。手法2のタイプのシステム構成は、本質的に逆階層構造となる。自律したエージェントが協調するためには、協調の媒介となる共有物が不可欠である。

本研究では、各ユーザーインターフェイス・エージェントが、この計算部の状態を監視し、それに対応して自らの状態を変更することにより、ユーザーインターフェイス・エージェント間の協調を実現する手法について検討した。これによって、いくつかのユーザーインターフェイス・エージェントが連動し合うのか、予め決定することができない状況に適応することが可能となった。逆階層構造による一貫性のための連動は、各ユーザーインターフェイス・エージェントが、集中管理ユーザーインターフェイス・エージェントの存在なしに自律的に行なうものである。この連動は、従来から協調問題解決と呼ばれてきたものである。

マルチ・エージェントによるユーザーインターフェイス構築技法は、階層構造に基づく方法と逆階層構造に基づく方法が提案できる。本研究では、エージェントの協調を一体化のための協調と一貫性のための協調とに分けた。階層構造に基づく方法、逆階層構造に基づく方法は、それぞれ一体化のた

めの協調、一貫性のための協調の記述に適し、もう一方の記述には適さない。現実のユーザーインターフェイスにはいずれのタイプの協調も含まれることから、両方の構造を用いてユーザーインターフェイスを記述すべきである。この仮説から下記のユーザーインターフェイス制御機能等を期待できる。

複雑なユーザーインターフェイスの一貫性を保つことができる。  
ユーザーインターフェイス以外によって計算部の状態が変更されたときも、ユーザーインターフェイスは一貫性を保つことができる。  
各ユーザーインターフェイス・エージェントを独立に作成し、実行できる。  
各ユーザーインターフェイス・エージェントの記述が適度な大きさである。

という利点を持つことを確認した。階層構造または逆階層構造の単独の適用では、これらの利点は同時に得ることはできない。1ユーザーインターフェイスに限らず、互いに影響し合う複数のエージェントで構成されるシステムは、元来1つの問題を複数のエージェントへと分割したタイプのシステムと独立した処理を行なうエージェントが必要に応じて情報をやり取りしてシステムの協調動作が行なわれるタイプのシステムの両方の特徴を合わせ持つ。これらのタイプのシステムには、それぞれ階層構造と逆階層構造による記述が適している。従って、階層構造と逆階層構造の併用によるシステム構築法は、マルチ・エージェントシステム一般に対する構成法をも与えている。

ユーザーインターフェイスは、システム開発の成功の鍵を握る最も重要な要素の一つである。システムが使い易いか使い難いかを決めるのはこのサブシステムであるためである。いかにそのシステムが高度なシミュレーション機能を有していようと、またいかに多くの特別なオプションを有していようと、操作する者にとってそのシステムが、使い難ければ、システムが最大限に活用されることはない。地層処分ヴァーチャル・エンジニアリング・システムの対話機能用として検討すべき対話機能は、マルチ・エージェント・ユーザーインターフェイス方式によることが将来への可能性の高い開発手法と考えられる。

## 2. 4 対話処理機能から見たシステムの操作

試作したグラフィック・ユーザー・インターフェイスは、将来の地層処分ヴァーチャル・エンジニアリング・システムの中核となる、並列複合事象シミュレーションの実行を操作するためのグラフィック・ユーザー・インターフェイス機能を想定した。機能は大別して、シミュレーション・シナリオの決定、シミュレーションの準備、シミュレーションの実行、シミュレーション結果の整理という4種類とした。尚、システム構成については、異機種分散型対話方式としてマシン・パラレルな環境を試作した。

### 2. 4. 1 操作用ユーザーインターフェイス

図2. 4-1に試作システムの構成を示す。試作システムは、4台のコンピュータと6台の大小モニタ画面を、1システムとして協調動作できるよう統合化したものである。



図 2. 4 - 1 試作システムの構成

図 2. 4 - 1 に示すとおり、本システムは 6 台のディスプレイを接続、それぞれのディスプレイは個別の機能を提供する独立した機能セグメントである。それらは下記の機能を提供する。

(1) 主操作機能

図 2. 4 - 2 に主操作画面を示す。これは、システムを初期設定した時表示される画面である。

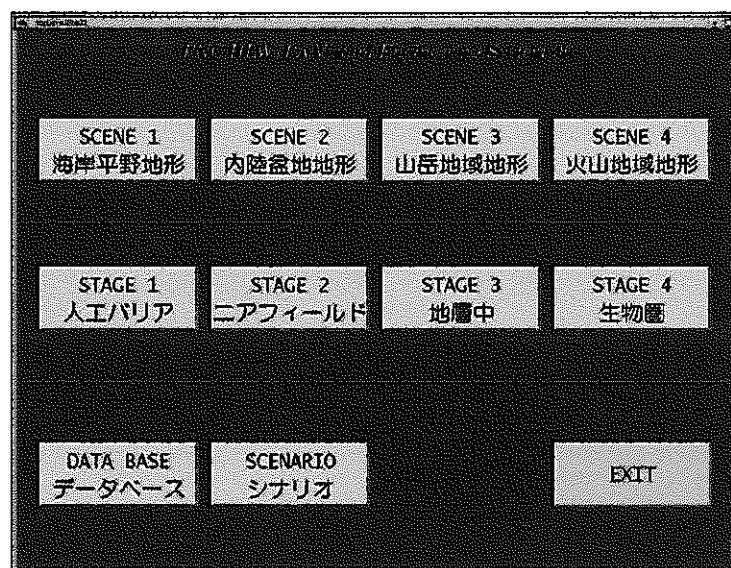


図 2. 4 - 2 主操作画面

図 2. 4 - 2 に示すとおり、主操作画面は簡単なメニューから構成した。メニューは下記の分類に添ってボタン・ウィジェットを実装した。図 2. 4 - 3 に天然地形情報画面を示す。



天然地形シーン想定ボタン・グループ

シーン1：海岸平野地形選択（3次元海岸平野地形オブジェクトを構成する情報）

シーン2：内陸盆地地形選択（3次元内陸盆地地形オブジェクトを構成する情報）

シーン3：山岳地域地形選択（3次元山岳地域地形オブジェクトを構成する情報）

シーン4：火山地域地形選択（3次元火山地域地形オブジェクトを構成する情報）

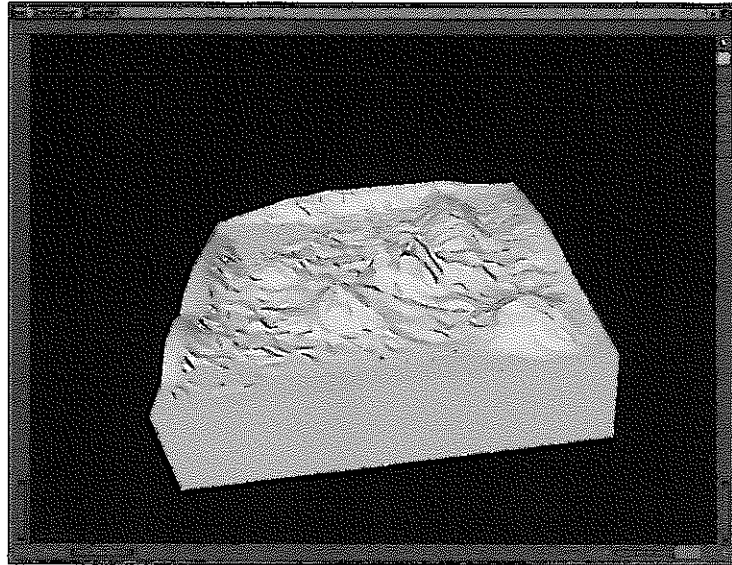


図2. 4-3 天然地形情報の例

ステージ選択ボタン・グループ

ステージ1：人工バリア核種放出シナリオ

人工バリア性能評価シミュレーション操作を開始するボタンウィジェット

ステージ2：ニアフィールド核種放出シナリオ

ニアフィールド性能評価シミュレーション操作を開始するボタンウィジェット

ステージ3：地層中核種移行シナリオ

地層中核種移行シミュレーション操作を開始するボタンウィジェット

ステージ4：生物圏核種移行シナリオ

生物圏核種移行シミュレーション操作を開始するボタンウィジェット

データベース保守等の補助ボタン

データ等の保守操作を開始するボタンウィジェット

システムの終了ボタン

本システムの操作を終了するボタンウィジェット

図2. 4-4 にステージ1：人工バリア核種放出シナリオ選択画面を示す。

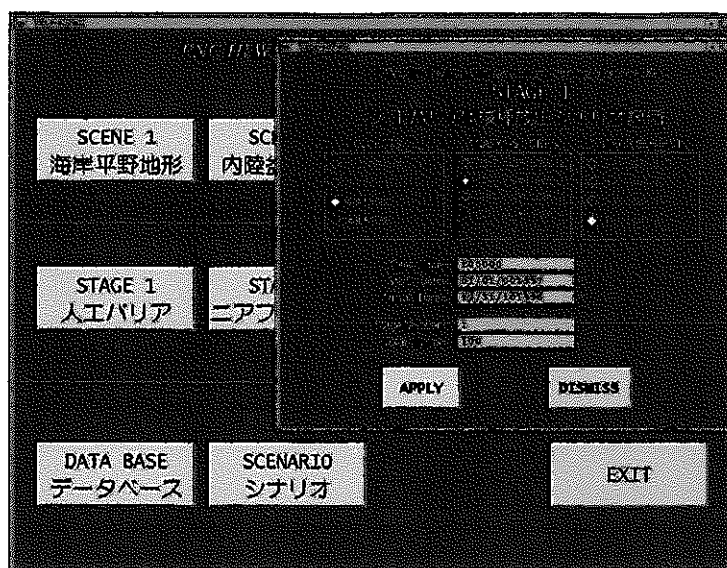


図 2. 4-4 ステージ1：人工バリア核種放出シナリオ選択画面

### (2) 3次元視覚化機能

図 2. 4-5 に 3次元視覚化機能を示す。図 2. 4-5 ではシミュレーション中の画面を表示している。

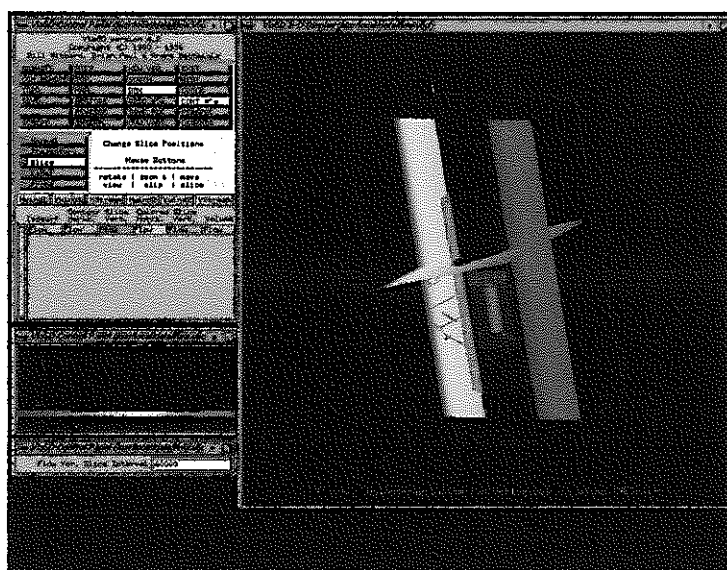


図 2. 4-5 3次元視覚化の例

### (3) シミュレーション・パラメタ表示機能1：タイム・パラメータ

図 2. 4-6 にシミュレーション・パラメタ表示機能1：タイム・パラメータを示す。図 2. 4-6 ではシミュレーション中の画面を表示している。

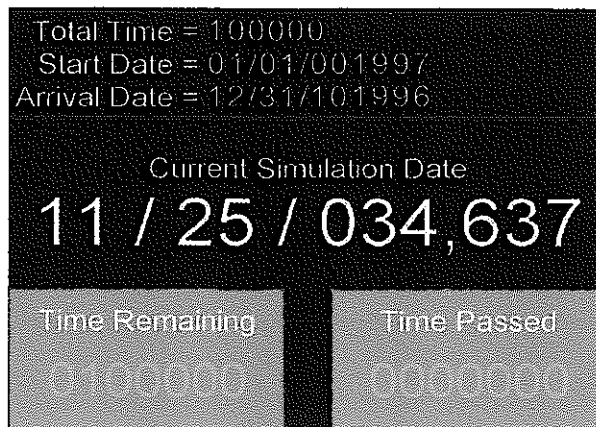


図 2. 4-6 タイム・パラメータ表示機能の例

(4) シミュレーション・パラメタ表示機能 2 : 選択シナリオFEP

図 2. 4-7 にシミュレーション・パラメタ表示機能 2 : 選択シナリオFEPを示す。図 2. 4-7 ではシミュレーション中の画面を表示している。

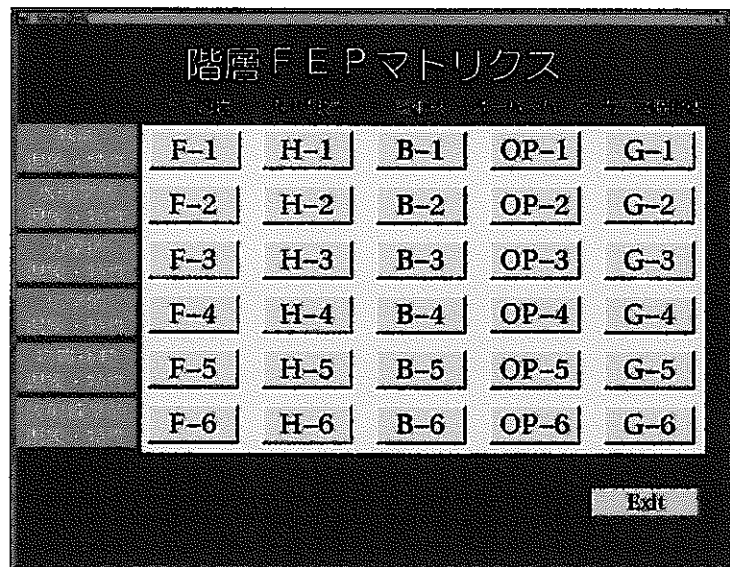


図 2. 4-7 選択シナリオFEP表示機能の例

(5) シミュレーション・パラメタ表示機能 3 : 計算条件、等

図 2. 4-8 にシミュレーション・パラメタ表示機能 2 : 選択シナリオFEPを示す。図 2. 4-8 ではシミュレーション中の画面を表示している。



図 2. 4-8 計算条件等、表示機能の例

(6) シミュレーション・パラメタ表示機能 4 : ステージ・パラメータ

図 2. 4-9 にシミュレーション・パラメタ表示機能 2 : 選択シナリオFEPを示す。図 2. 4-9 ではシミュレーション中の画面を表示している。

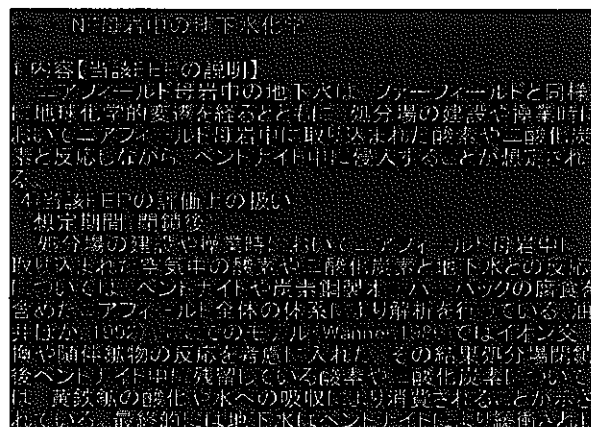


図 2. 4-9 ステージ・パラメータ表示機能の例

(7) データベース操作機能

図 2. 4-10 にFEPデータベース・マトリクス画面例を示す。図 2. 4-10 は試作した分散型マルチメディアFEPデータベース・システムの例である。

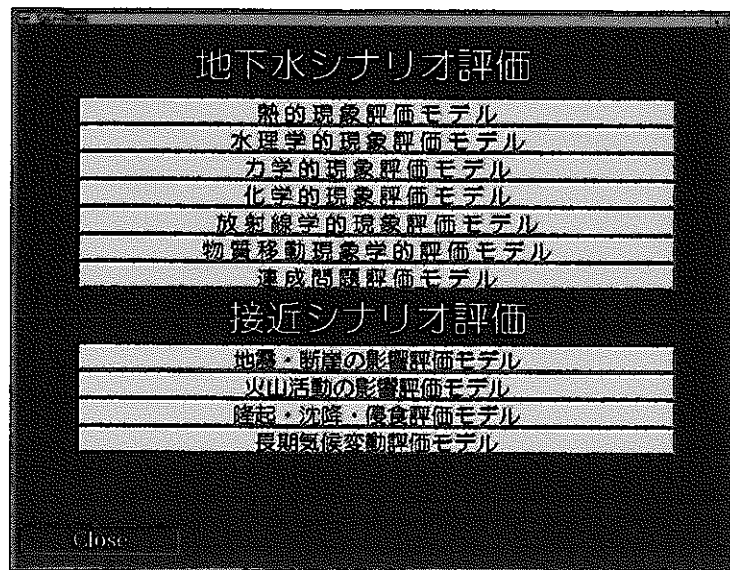


図 2. 4 - 1 0 データベース操作機能の例

## 2. 4. 2 試作対話システム制御機能概念

### (1) 主操作制御仕様：Main Console Panel

機能：地層処分性能評価ヴァーチャル・エンジニアリング・システムの起動

目的：ユーザー、ハードウェア等リソースを確認してシステムを起動する

仕様：試作システムでは、User ID, Password等を確認し、システムを初期化するログイン手続き用 GUIを準備する。尚、User ID, Passwordについては固定とし、それ以外であれば警告を出して終了。

表示：1) コンソール システム・コンソールGUIを表示

2) ピュアー クリアー後スタンバイ・メッセージ表示

3) P C クリアー後スタンバイ・メッセージ表示

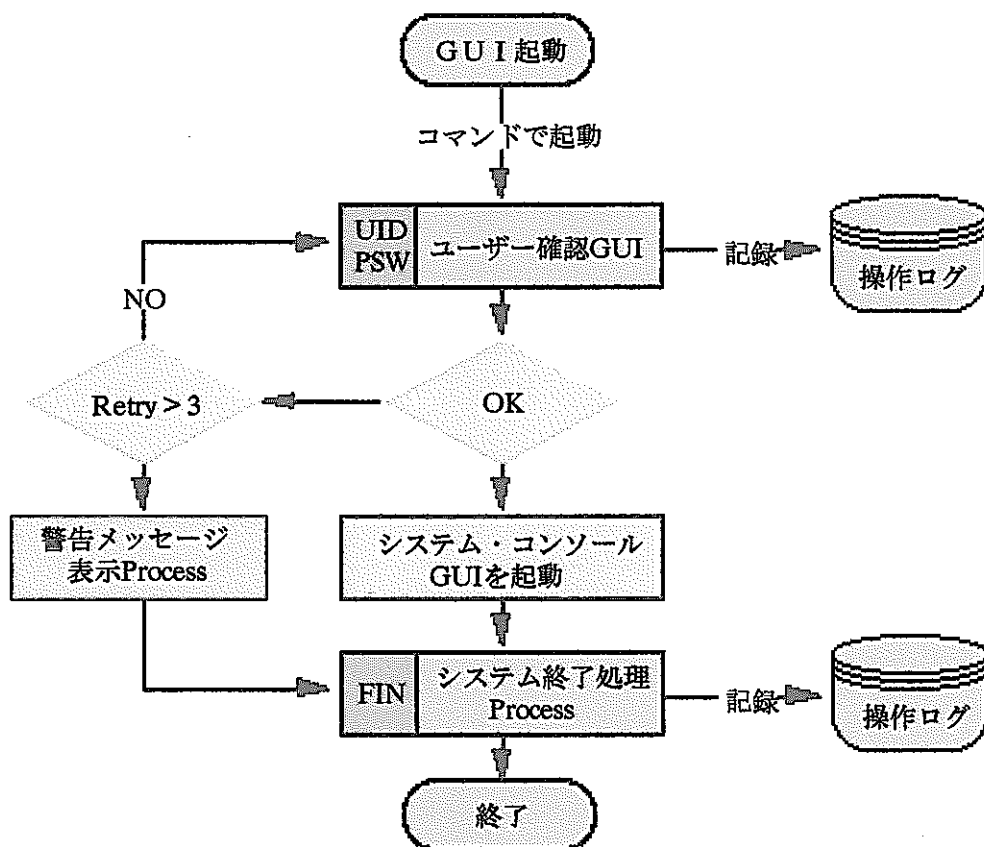


図 2. 4 - 1 1 主操作制御仕様

## (2) システム・コンソール制御基本ウィジェット

機能：システム・コンソール・ボタン・ウィジェット・コール

目的：シーン、ステージ、保守、終了処理機能を起動する

仕様：試作システムでは、天然地形を選択することにより、動作をリセットする。オブジェクト・ビューアーに選択シーンの天然地形を表示する。平成8年度に試作するシステムでは、天然地形シーンのオブジェクトはステージ4のオブジェクトと同等であるとする。ステージ1から4まステージ・ボタンは評価するステージに入るボタンとする。従って、ステージ・ボタンを選択すると選択されたステージのGUIが表示されるものとする。

表示：1) コンソール：ステージGUIを表示

2) ビューアー：シーン、ステージで選択された3Dオブジェクトを表示

3) PC-A：ステータス情報表示(選択ウィジェット等)

4) PC-B：シミュレーション情報表示(初期状態)

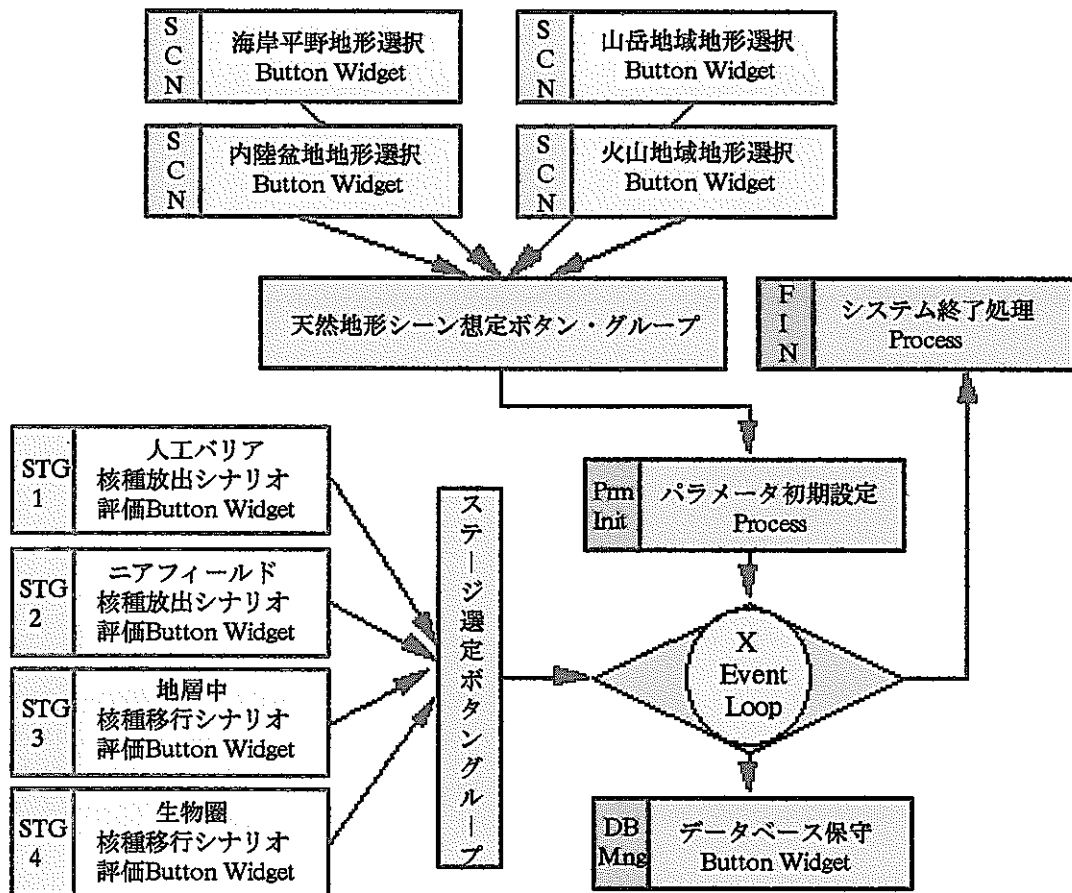


図 2. 4-12 システム・コンソール制御基本ウィジェット仕様

(3) シーン選択ボタン海岸平野：内部処理Button Widget

機能：シーン選択ボタン（海岸平野内部処理概要）

目的：想定される処分システムをとりまく天然地形を海岸平野に設定する。

仕様：試作システムでは、単に海岸平野地形に建設された処分システムの概観を3D表示するにとどめるがシミュレーション情報ファイルに、半島地形が選定されていることを示すフラグを立て、次のシーン選択アクションが行われるまでの間の性能評価プロセス等における参照に利用する。これは、将来的に、天然地形の相違にかかわるシナリオの相違が考えられるためフラグを立て、各ステージ、ターンにて、現在選択中の天然地形を確認する必要が生じることがシステム開発上予想されるためである。

表示：半島オブジェクトを大型モニタに表示。（改良Vis5D）

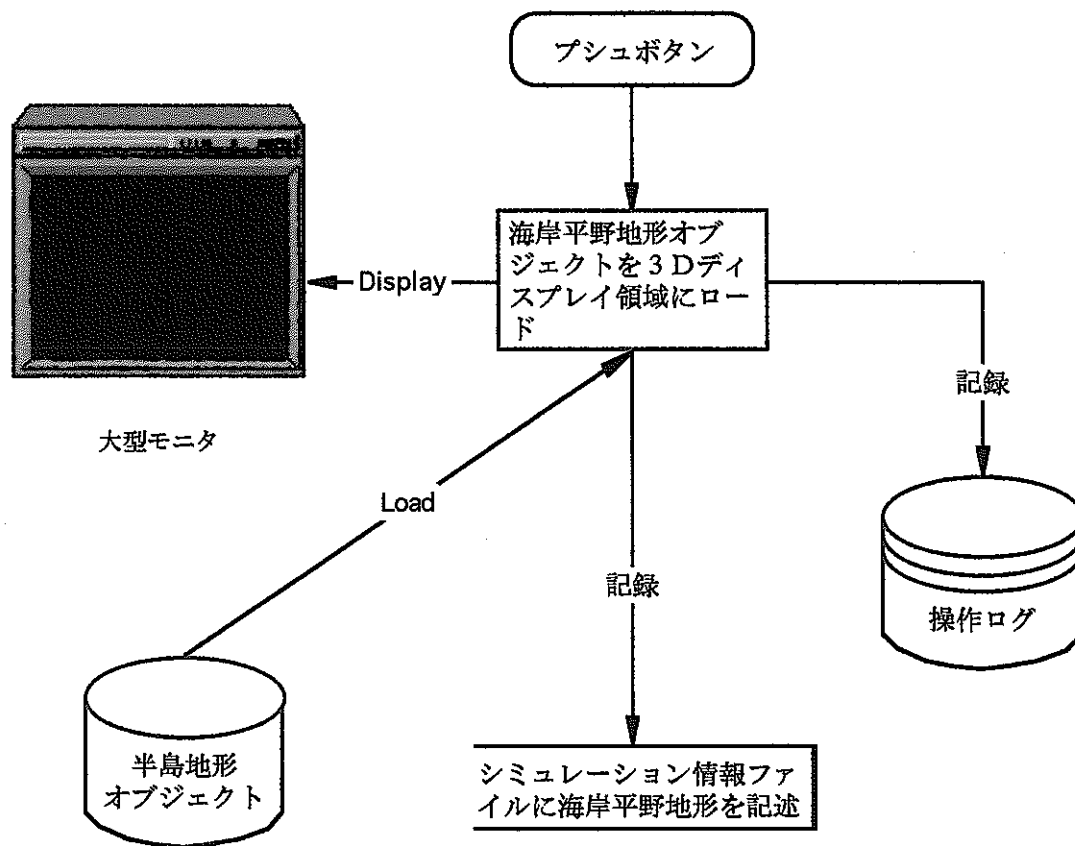


図 2. 4-13 シーン選択ボタン海岸平野：内部処理

(4) シーン選択ボタン内陸盆地：内部処理Button Widget

機能：シーン選択ボタン（内陸盆地内部処理概要）

目的：想定される処分システムをとりまく天然地形を内陸盆地に設定する。

仕様：試作システムでは、単に内陸盆地地形に建設された処分システムの概観を3D表示するにとどめるが、シミュレーション情報ファイルに平野地形が選定されていることを示すフラグを立て、次のシーン選択アクションが行われるまでの間の、性能評価プロセス等における参照に利用する。これは、将来的に、天然地形の相違にかかわるシナリオの相違が考えられるため、フラグを立て各ステージ、ターンにて、現在選択中の天然地形を確認する必要があることがシステム開発上予想されるためである。

表示：平野オブジェクトをオブジェクト・ビューワー画面に表示。（改良Vis5D）



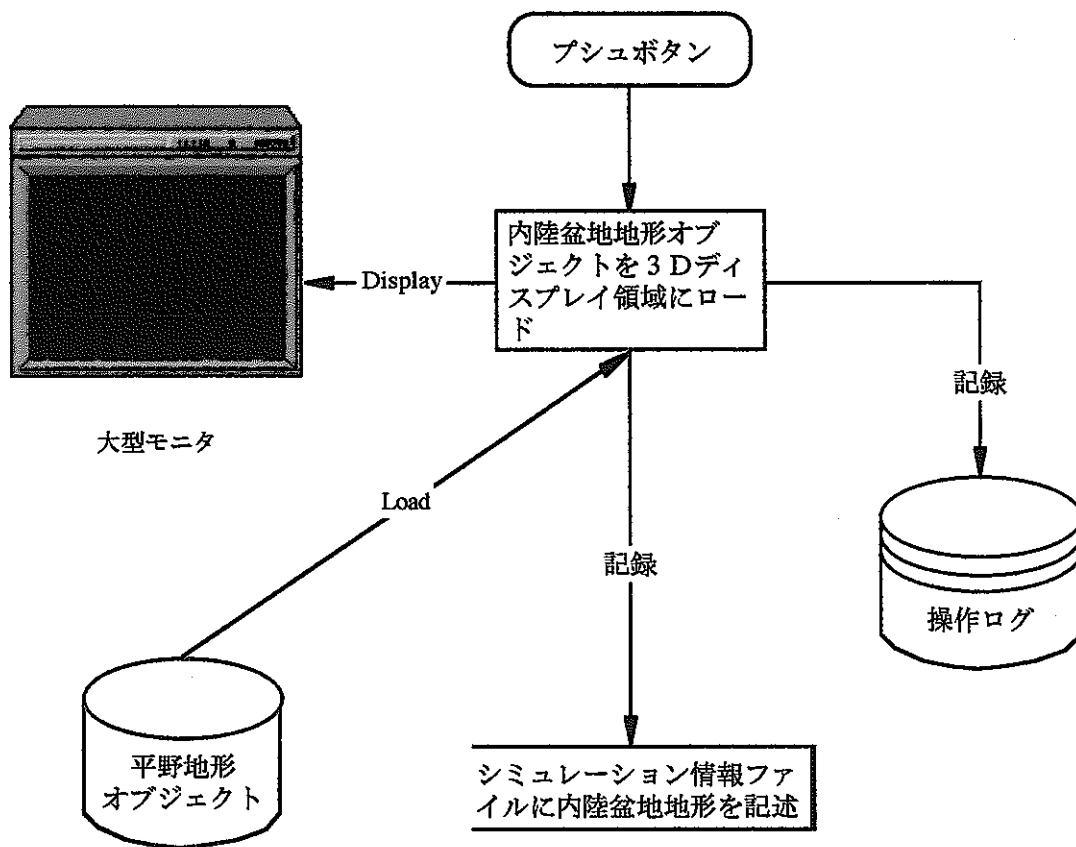


図 2. 4-14 シーン選択ボタン内陸盆地：内部処理

#### (5) シーン選択ボタン山岳地域地形選択Button Widget

機能：シーン選択ボタン（山岳地域内部処理概要）

目的：想定される処分システムをとりまく天然地形を山岳地域に設定する。

仕様：試作システムでは、単に山岳地域地形に建設された処分システムの概観を3D表示するにとどめるが、シミュレーション情報ファイルに山岳地域地形が選定されていることを示すフラグを立て、次のシーン選択アクションが行われるまでの間の、性能評価プロセス等における参照に利用する。これは、将来的に、天然地形の相違にかかわるシナリオの相違が考えられるため、フラグを立て、各ステージ、ターンにて、現在選択中の天然地形を確認する必要があることがシステム開発上予想されるためである。

表示：山岳地域オブジェクトをオブジェクト・ビューワー画面に表示。（改良Vis5D）

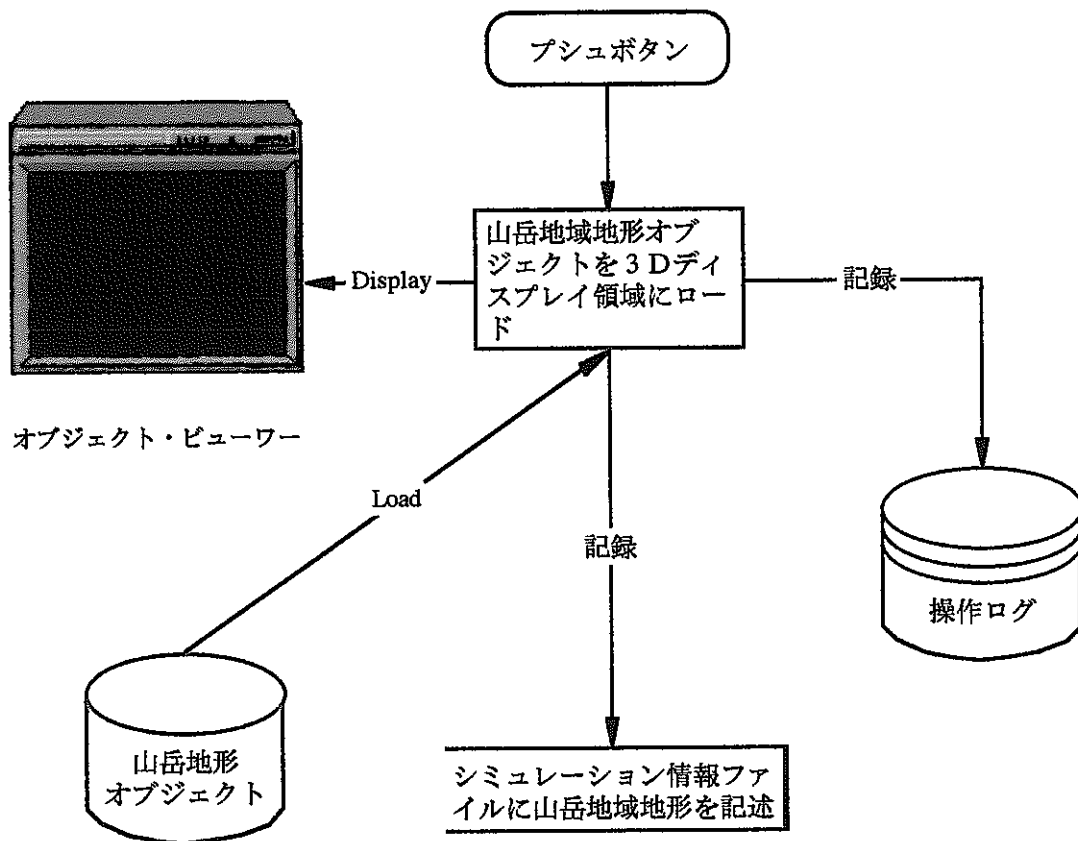


図 2. 4 - 1 5 シーン選択ボタン山岳地域地形選択内部処理

(6) シーン選択ボタン火山地域：内部処理Button Widget

機能：シーン選択ボタン（火山地域内部処理概要）

目的：想定される処分システムをとりまく天然地形を火山地域に設定する。

仕様：試作システムでは、単に火山地域地形に建設された処分システムの概観を3D表示するにとどめるが、シミュレーション情報ファイルに火山地域地形が選定されていることを示すフラグを立て、次のシーン選択アクションが行われるまでの間の、性能評価プロセス等における参照に利用する。これは、将来的に、天然地形の相違にかかわるシナリオの相違が考えられるため、フラグを立て、各ステージ、ターンにて、現在選択中の天然地形を確認する必要があることがシステム開発上予想されるためである。

表示：山岳オブジェクトをオブジェクト・ビューワー画面に表示。（改良Vis5D）

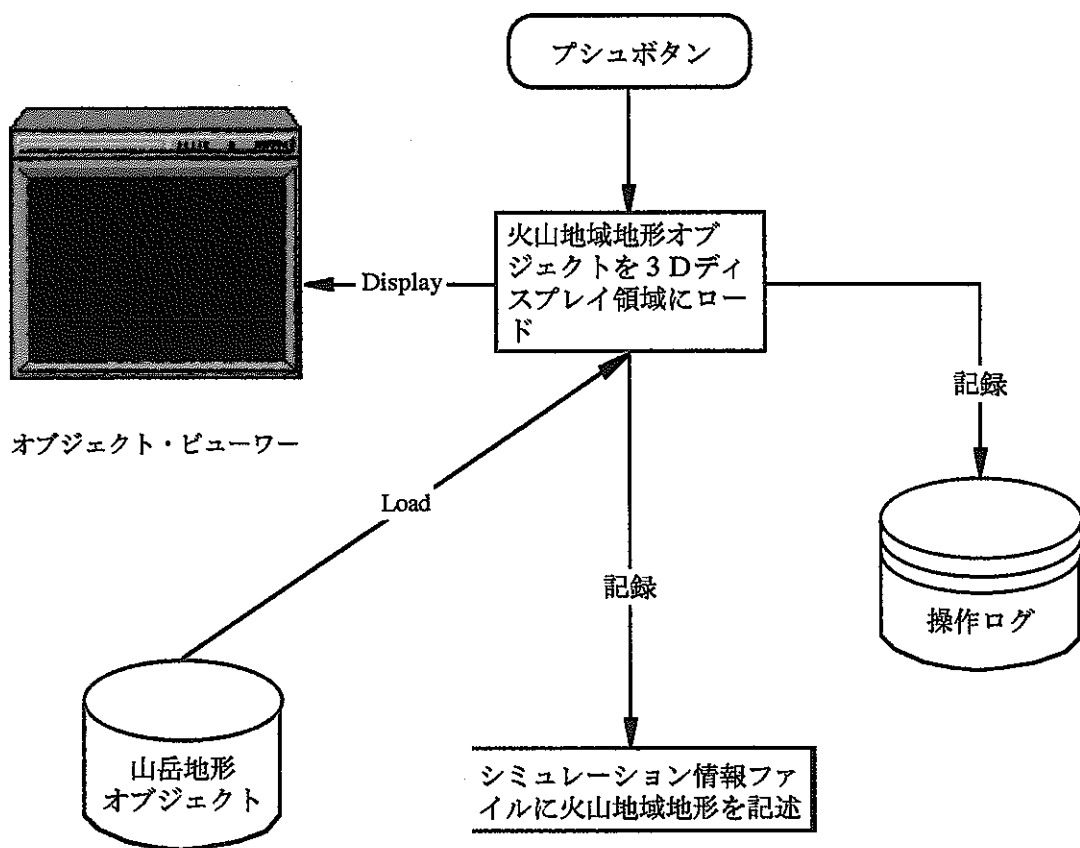


図 2. 4-16 シーン選択ボタン火山地域：内部処理

#### (7) パラメータ初期設定内部処理

機能：システム・パラメータ初期設定

目的：シミュレーションに必要なシナリオ、ステージ、カレンダー等共通情報を初期設定する

仕様：試作システムでは、1ターンの間保持するシミュレーション情報について初期化する。これらの情報は、パラメータ・ディスプレイ・パネルに常時表示される。実際にはコンソール・システムの/usr/tmpにファイルが作成され、各パラメータ・ディスプレイ・システムにより、指定したタイム・インターバルで参照するものとする。

表示：Panel-A) シミュレーション用、カレンダーと時刻パラメータの表示。

Panel-B) 現在、選択中のシナリオ関連の表示。

Panel-C) シミュレーション・パラメータ関連の表示。

Panel-D) ステージ、ターンの表示。

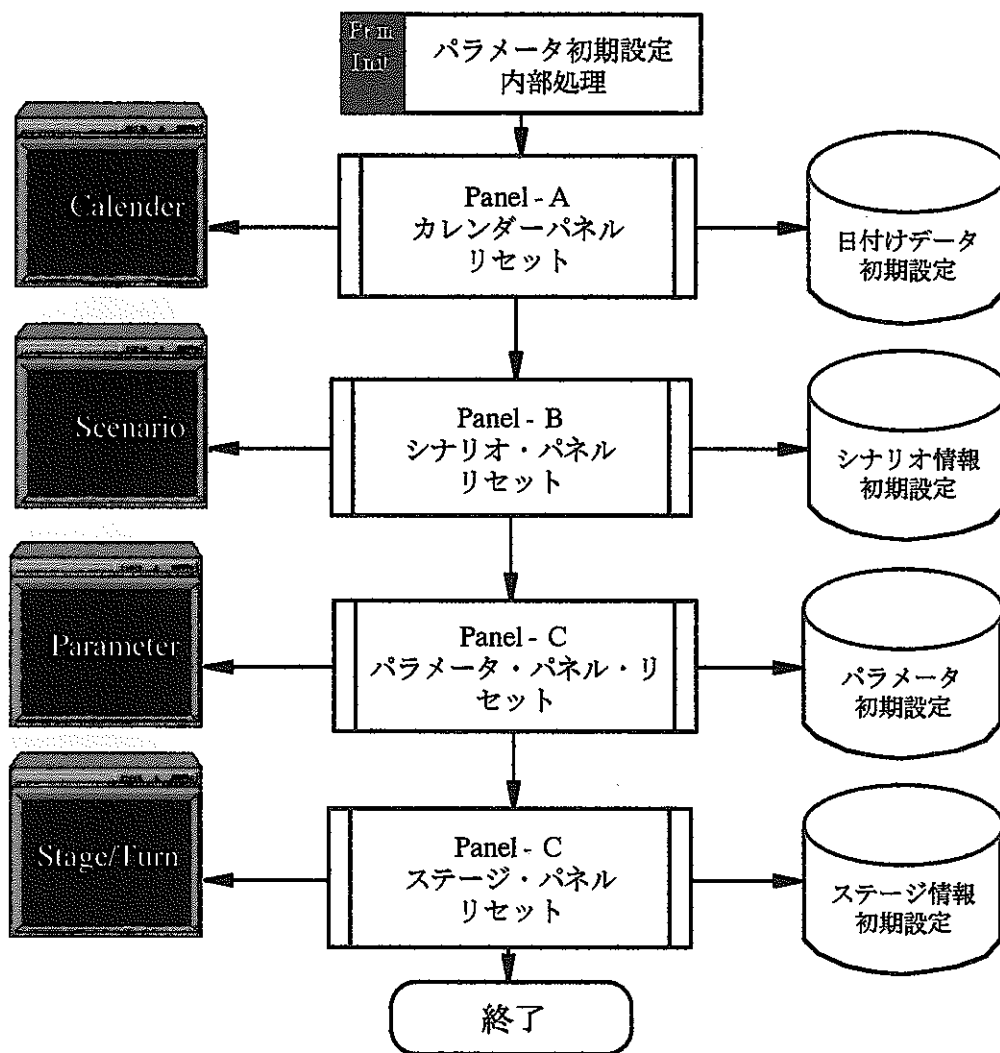


図 2. 4-17 パラメータ初期設定内部処理

## (8) ユーザー確認GUI

機能：ユーザー確認、システム初期設定

目的：ユーザー、ハードウェア等リソースを確認してシステムを起動する

仕様：試作システムでは、User ID, Passwordを固定とし、それ以外であれば警告を出して終了とする。

表示：1) コンソール システム・コンソールGUIを表示

2) ビュアー クリアー後スタンバイ・メッセージ表示

3) PC クリアー後スタンバイ・メッセージ表示

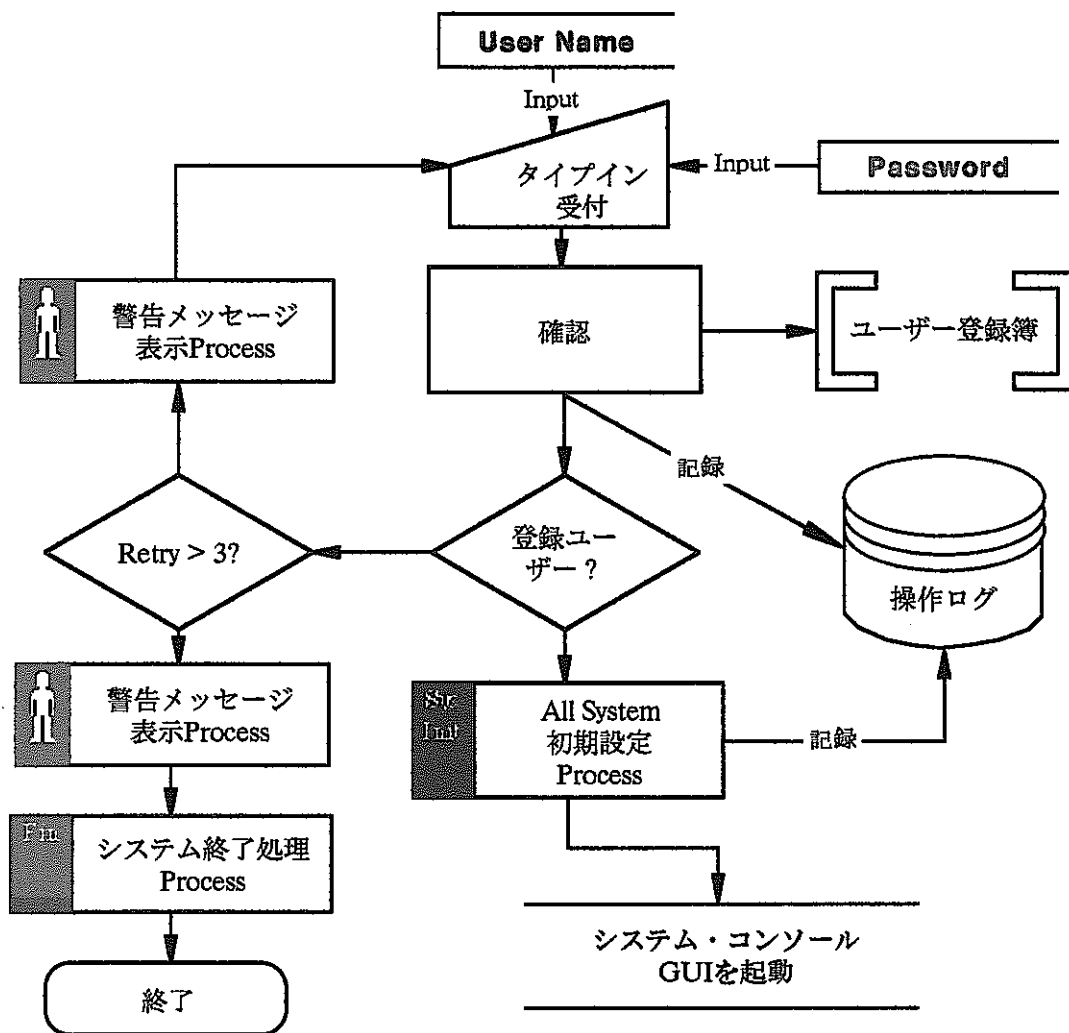


図 2. 4-18 ユーザー確認GUI内部処理

### (9) 全システム初期設定

機能：地層処分性能評価ヴァーチャル・エンジニアリング・システム初期設定

目的：ユーザー、ハードウェア等リソースを確認してシステム初期設定する

仕様：試作システムでは、各ディスプレイ画面等を初期設定する。準備完了メッセージを表示する。

表示：1) コンソール システム・コンソールGUIを表示

2) ビュアー クリアー後スタンバイ・メッセージ表示

3) PC クリアー後スタンバイ・メッセージ表示

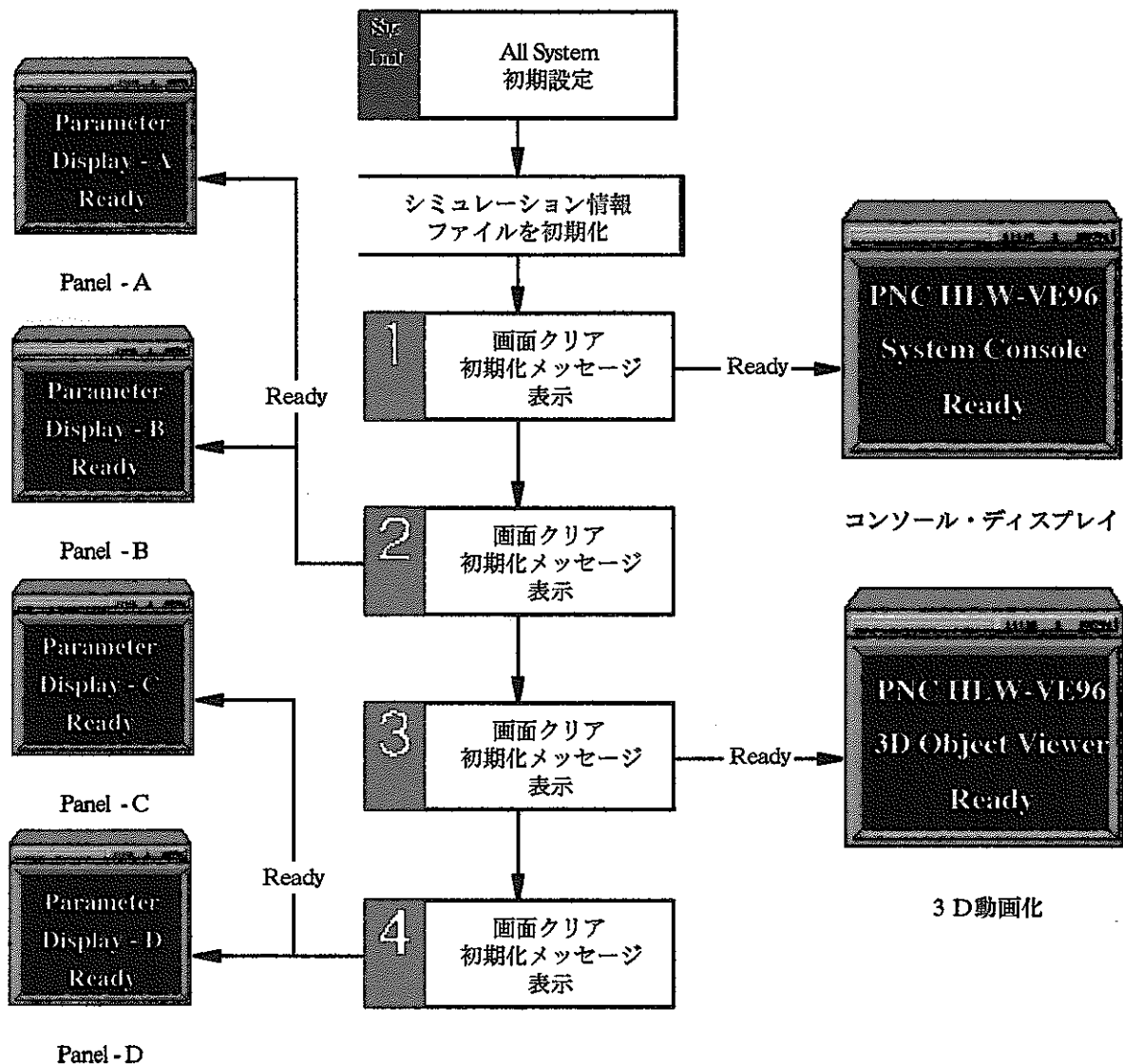


図 2. 4-19 全システム初期設定内部処理

### (9) Panel - A: カレンダー・パネル

機能：パラメータ表示装置 系統-A

目的：シミュレーション等の時刻、カレンダー情報の表示

仕様：試作システムでは、現在、実時刻、シミュレーション現在時刻、等を表示しシミュレーションの時間依存性の視覚的効果を高める。

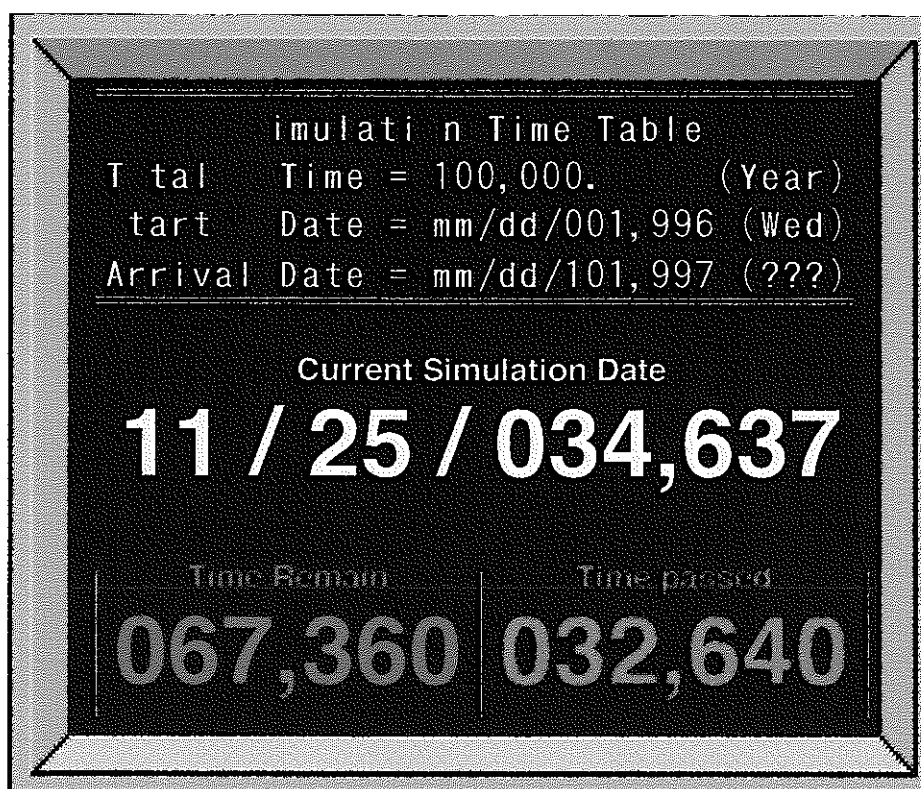
備考：スライダー等のビジュアルな動図による表現を併用することが好ましい。

表示：1) シミュレーション終了時刻（取敢ず西暦10万年、1000世紀）

2) 現シミュレーション時刻（実行時にはカウント・アップ）

3) 実時間（実際の時計時刻）

4) 表示仕様は下図を基本案として視覚的に見やすくスマートな表現を行う



## Parameter DISPLAY - A

図2. 4-20 全システム初期設定内部処理

### (10) Panel - B : シナリオ・パネル

機能：パラメータ表示装置 系統-B

目的：選択した評価シナリオに関する情報の表示。

仕様：試作システムでは、現在、評価中のモデルに関連するFEPリストを表示する。

表示：1) 選択したシナリオをFEPリストから表示する

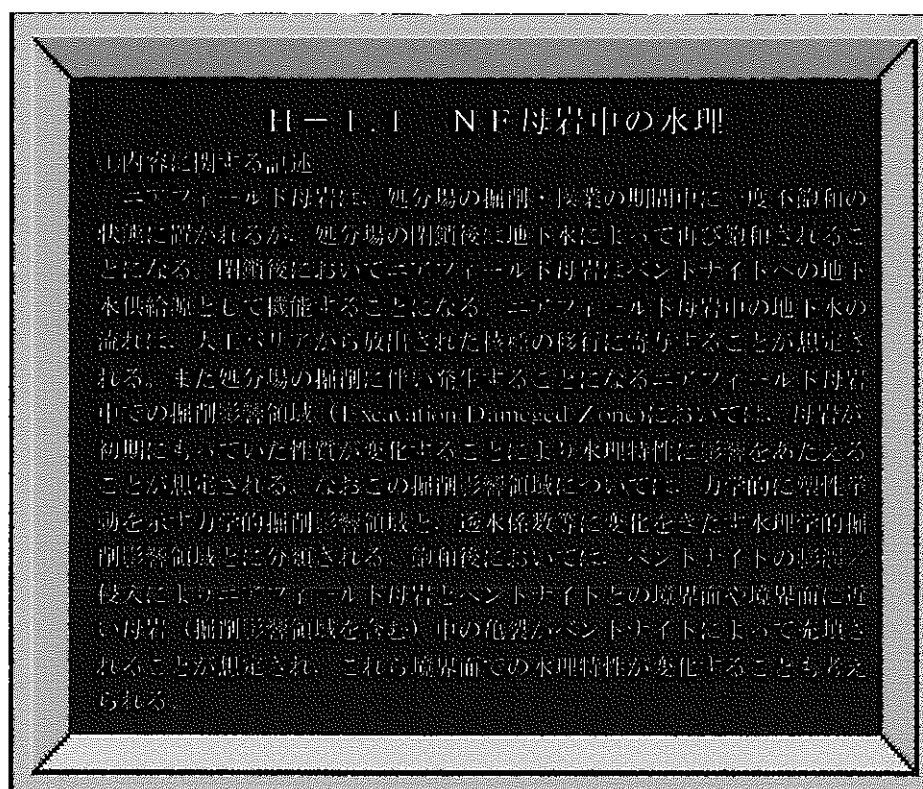


図 2. 4-21 Panel-B: シナリオ・パネル

## 2. 5 性能評価シミュレーション面から考えたシステム操作要求

ユーザーとの対話処理操作をより効率化すると共に、ユーザーによる意思決定を支援する為の参考となる情報を提供する機能について検討した。

地層処分ヴァーチャル・エンジニアリング・システムとユーザーとの対話は、評価シナリオの決定、シミュレーションの実行、実行結果の整理等、シミュレーションに直接関連する操作に始まり、シミュレーション結果の妥当性等を検証・確認するための関連情報をデータベースより検索し、解析結果と比較評価する、等のさまざまな性能評価計算に係わる操作が予想される。また、シミュレーション・コードやデータ等、ソフトウェアの管理に係わる操作も考えられる。対話システムを利用しやすいシステムとして構築するためには、その設計思想に使いやすさが吟味されていなければならない。

例えば、ある系統立った評価シミュレーションを実施する場合、シミュレーションは評価シナリオにそって進められるべきであり、個別の評価シナリオに関連するFEPリストやインフルエンス・ダイアグラムはシミュレーション操作の支援知識となるであろう。

使いやすいユーザーインターフェイスを設計するためには、操作全体の流れと、画面のレイアウト、メッセージの解かりやすさ等が吟味されなければならない。そのためには、システムの目標や機能について整理し、操作のゴールを明確にすることが重要である。



高レベル放射性廃棄物処分システムの安全性評価は、処分システム開発プロジェクトの各段階について行われる。地層処分ヴァーチャル・エンジニアリング・システムは、それらの各段階について、目的に適合した結果を提供できなければならない。このため、その各開発段階に応じて評価結果の使用目的が異なる。当面の地層処分ヴァーチャル・エンジニアリング・システムの目的は、処分システムの開発における第2段階で実施される処分サイトの特定、及び処分システムの概念設計の妥当性を判断する情報を提供することにあると考えられる。具体的には、想定サイトでの処分の環境影響の概要、処分システムの設計オプションに関する多方面からの安全性評価比較の2点にあると考えられる。

### 2. 5. 1 安全性評価作業の基本的な操作

安全性評価の基準としては、処分が公衆にもたらすリスクが、社会的に十分許容されるレベルかどうかで判断することが基本であると考えられる。また、安全性評価のためのシナリオ選定として効率的な評価を行うためには、FEPによる手法と、合わせて事象の発生確率と影響の程度に応じて評価方法を合理的に選定する必要があると思われる。従って、個別シナリオと確率論的シナリオを用いて、評価シナリオをより簡素化し、下記の分類に添ってシミュレーション等を行い、影響を評価することが安全性評価の基本的考え方と思われる。また、この考え方は、地層処分ヴァーチャル・エンジニアリング・システムの操作の基本的な考え方となるものである。それは、リスクの影響度と頻度から下記のクラスに分類することにより、情報を構造化することができる。

Class 1：影響大頻度大 設計で対応する

Class 2：影響大頻度小 定性的な評価を行う

Class 3：影響小頻度大 定量的な解析評価を行う

Class 4：影響小頻度小

定性的な検討を行う。ここで大小の判断基準が問題になるが、このアプローチでは

\* 「影響大」、「頻度大」の場合には両者は比較的良好な精度で推定できること、

\* 影響の大小より、事象の発生確率の大小の推定の方が技術的に困難であるという考えがベースにあると考えられる。

高レベル放射性廃棄物処分に係る安全性評価は、計画する多重バリアシステムによる放射性核種の長期間に亘る隔離能力を対象に行われなければならない。従って、システムの操作の基本は、この多重バリアの性能を数値的に評価し、今度の処分システム開発に参考となる情報を提供することを目的とした作業に適していることが望まれる。これらの問題を効果的に解決するには、次の様な考え方を基にすることが妥当であるとしている。

### 2. 5. 2 評価期間

いずれの評価を行う場合にも、評価期間により評価手法も評価に必要な精度も異なる。そこで著書らは次の3期間に評価期間を分けるのが良いと考える。

#### (1) 短期（約1000年まで。処分後の地下水浸潤等比較的過渡的な期間）

廃棄物中の全放射エネルギーの時間変化を見ると、初期の1000年で約1/1000になることが知られてい

る。それ以降は、清練に使ったウラン鉱石全体中に存在する量と大差ないと考えられる。放射能濃度でみるとウラン鉱石の約10倍程度になる。従って、前述の『影響大』のケースClass 1, 2が該当する事象が起きる可能性のあるのは1000年までであり、この期間は『影響大』の事象が発生しないように設計することが望ましい。システムの開発に於ては、この間は複雑な地球化学的過程を中心とした現象に関心が集まることを重視して設計すべきである。歴史的資料や考古学的建造物等の情報をデータベース化し、工学的手法参考になる情報を提供することにより設計作業を支援できる。また、事象の発生確率の大小をPSA等により、根拠をもって定性的に予測することが可能である。

しかし、現代の経験で確かなことが言えるのは100年程度であり、科学的予測の精度を上げるためには更に基礎研究が必要である。そのためには、この期間の現象解明に係わるより包括的な評価を支援する対話機能が必要である。

## (2) 長期 (約1000 - 10,000年)

放射能濃度は1万年経過するとウラン鉱石と同じオーダーになると考えられている。従って、長期の評価に於ては影響は少ないと考えられるが、発生の確率が高いと思われる事象シナリオについて評価の関心が高まると思われるため、対話システムの機能は、より多くの気候変動や地質等のデータベースについて充実したものが求められると考えられる。

例えば、氷河期の周期は数万年オーダーであり、約1万年程度であれば、自然条件、生物活動等が現在と同じと考えても不自然ではないと思われる。生物圏に漏出する放射能による影響を現在の生活条件を根拠に定量的に被曝線量として評価しても不当ではないであろう。

また、第四期 (約200万年) の地殻変動データを使うことにより、地中における約1万年以内の事象は比較的定量的に評価できる。

## (3) 超長期 (1年以上)

1万年を越える超長期の評価においては、生物活動について人類の存続の疑問も含めて不確定要素が多くなり、被曝線量を評価する意義が疑わしくなるかも知れない。ウラン鉱床中の放射能濃度と処分システム中の平均放射能濃度は大差なくなると思われる。このように、長期の評価について当面は天然に存在する放射性鉱物の放射線影響との比較において、定性的な検討にとどめるのが適当と思われる。従って、対話システムは、より優れたナチュラル・アナログ・データベース等の機能を充実すべきであると考えられる。

### 2. 5. 3 多重バリア方式

多重バリアシステムを採用した場合は、どの部分を各バリアの境界と考えるかが重要となる。例えば、米国NRCの基準では廃棄物パッケージと地層の2つのバリアに分けている。設計ベースで考えると、オーバーパックと緩衝材を2つのバリアと考え、更に地層バリアが加わり3重のバリアシステムと考えることができる。この場合、容器は密封機能、緩衝材は移行遅延機能、地層は隔離機能 (接近できなく、移行距離が長い) と、それぞれ異なった機能で核種移行を阻止する働きをする。従って、この異なった機能を期待する評価シミュレーションにおいては、これらのバリア構成要素を明確に3次元オブジェクト等で視覚化して区別することにより、整理されたシステムを構築する必要があると

考えられる。

また、長期の評価においては、高放射線被曝の事象の発生はほとんど考えられないので、極めて低い発生確率であることを定量的に評価することは必ずしも要求されないと思われる。単独で閉じ込め性の高いバリアを重ねる必要はない。しかし、我が国の地層は必ずしも安定ではないので、設計目標としては、地層の不安定性に起因する不確定要素を排除することを目的とした、多重バリアを採用した方が評価が容易となることが多重バリア方式の採用論拠である。この多重バリア方式の処分システムの機能を、周辺母岩と合わせて機能別に分類評価できる対話画面構成が重要である。

また、地層データベースについても詳細な調査結果から構成でき、確実な性能評価が可能な処分システム周辺の岩盤と距離による隔離機能を期待する広い範囲の地層の多重バリア・システム設計の支援機能を考えることができる。

#### 2. 5. 4 評価シナリオ

当面の評価シナリオとしては、処分システムの建設時における掘削影響等からの地下水環境変動シナリオ、及びそれぞれが原因となった核種の放出シナリオを開発する必要がある。地層処分ヴァーチャル・エンジニアリング・システムの開発においては、シナリオは別途推進されている、地層処分シナリオ・データベースの開発プロジェクトにより得られたものを応用する。処分システムに及ぼす影響事象を、サイト特定、及び設計によりかなりを除くことができると考えられている。更に、除くことのできない事象のうち、放射能の漏洩に結び付くシナリオを抽出する方法として、それぞれの事象に定性的発生確率を与え、相対的発生確率から重要事象を求める方法がとられる場合が考えられる。

また、定性的な数値は、科学的根拠に基づいて与えることができるとは限らない。それらの数値の見積りに当たって、根拠となる科学的データが得られない場合は、まず専門家の見識に頼ることになる。また、同時に、公衆の感覚的判断も大切になる。そこにいわゆるPublic Criteriaの必要性が生じる。合理的なPublic Criteriaを形成するためには、まず、関連する科学的事実を公衆に伝達することが大切である。そして、長い時間をかけ、多くの人が理解できるよう整理する必要がある。以上のことから、シナリオに添ったシミュレーションを操作する機能には、確率論的評価面から重要事象が示されることや、将来のPublic Criteriaとの関係、社会システムへの影響といった包括的な機能が設計当初より含まれていることが望ましいと考えられる。

#### 2. 5. 5 評価モデル

処分システムの安全性評価のためのモデルは、放出核種の移行過程等を表現する複数のモデル（サブモデル）から構成されている（評価モデル・システム）。これらのモデルは、対話システムから見ればバックグラウンドで独立したタスクとして動作するオブジェクトとして扱うことが好ましいと考えられる。また、それらは、対話システムのウィジェットや他のタスクからのメッセージで起動され、並列的に協調して動作する必要がある。

サブモデルは、基本的には核種の移行過程をできる限り正確に表現するものであるべきであるが、一方、評価モデル・システムは、総合的な解析を行える機能が必要なため、サブモデルは簡単なものが望ましい。詳細モデルに於ては、我が国では火山による飛散、地殻変動による露出等の直接被曝シナリオも他国より慎重に評価する必要があるが、基本的にはサイト特定でそれらの要因を削除できる

と考えられる。

例えば、地殻のゆっくりした上下動等、必ず起きる地殻変動として評価しなければならない事象もあるが、その場合は、境界条件の時間的变化による地下水の動きへの影響を考慮することになる。これらのことから、我が国においても、地下水による核種移行モデルが評価の基本となると考えられる。

従って、モデルの選定とシミュレーションの実行においては、サイト特定で削除できる要因等を明確に表示し、評価シナリオを簡素化する操作を視覚的に表現することが好ましいと考えられる。

### (1) 容器の耐久性評価

人工バリアは、固化体（キャニスタ入り）、容器、緩衝材からなる。容器の健全性は、核種の移行の開始時期を決めることになる。容器は完全封入機能を持ち、腐食等の破損による開口部が生じた時点でバリア機能を消失するとして評価する。小さな開口部であっても、一旦開口部ができるとバリア機能が他のバリアの機能に比べて極度に低下する。容器は、キャニスタとオーバパックの組み合わせを考え、代表的材料として、キャニスタはステンレス鋼を考え、オーバパックは炭素鋼を考える。容器の耐久性の評価は、腐食速度の評価と地震等の瞬間的な破壊につながる要因の有無の評価にわけて考えることができる。全面腐食、及び局部腐食の速度に影響する要因として、温度、酸性度、地下水の組成、放射線線量率、酸化還元電位等について考える。シミュレーションに於ては、これらの環境条件の時間変化を求め、その環境条件での腐食速度の積分値として、腐食の厚さを計算する手法等が用いられている。オーバパックの腐食現象に係わる評価は、化学過程を中心としたシミュレーションになるものと思われ、シミュレーション結果の視覚化表現には新たな手法の開発が必要である。

また、瞬間的破壊としては、応力腐食割れと機械的破壊が考えられる。これらの破壊については、あらかじめ割れの生じる環境条件を実験的に明らかにし、その環境条件が処分環境にないことを確認する。機械的破壊については、破壊応力以上の外力が加わらないことを評価する。これらの要因が生じた時点で容器は破壊し、バリアとしての能力はなくなるものとする。これらの破壊についても、対話システムは優れた視覚化機能が要求されると考えられる。

図2. 5-1から図2. 5-3に試作したオブジェクトより視覚化した人工バリア構成物を示す。

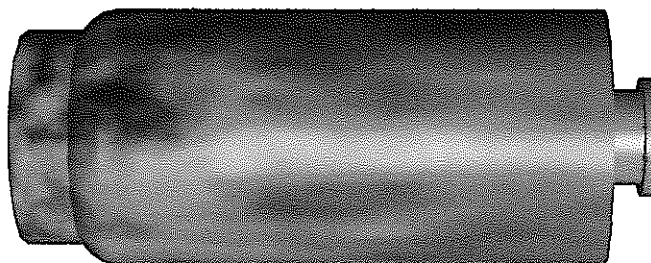


図2. 5-1 ガラス固化体キャニスタ（試作オブジェクトより）

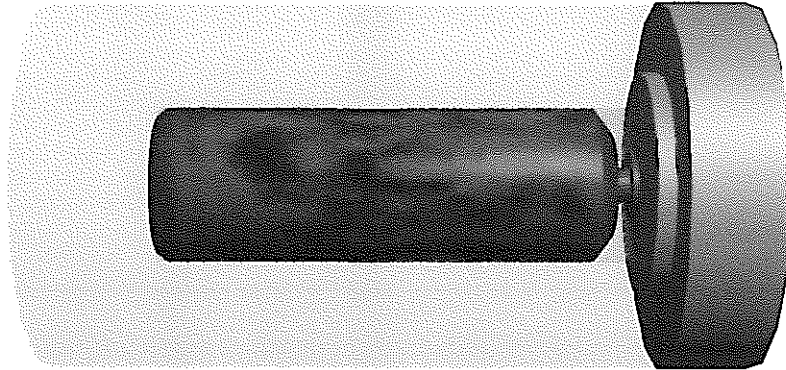


図2. 5-2 オーバーパック（試作オブジェクトより）

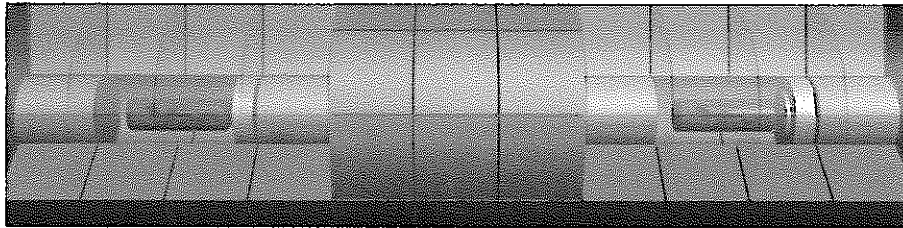


図2. 5-3 緩衝材（試作オブジェクトより）

## （2）核種浸出評価

貯蔵時、輸送時の評価においては、時間に比例した浸出が生じるとして、浸出率により評価する。処分システム操業後のソースターム評価においては、処分環境の地下水中で平衡反応から決まる溶解度モデルと拡散、及び地下水の動きに構造物物質移動により評価する手法が取られている。

溶解度モデルによる評価のパラメータとしては、処分環境の温度、地下水の組成・pH・Eh、共存固相（ガラス、容器、緩衝材等）の組成が挙げられる。放射線は水の放射線分解をもたらし、Eh・pH等を定める2次要因として重要である。これらの平衡反応による評価要因に加えて、より溶解度の低い結晶性化合物にゆっくりと変換し、実質上核種移行評価系から除かれる、いわゆる鉱物化も重要な評価シナリオである。このような地球化学シミュレーションについて対話システムは、熱力学データベースと連結したパラメータ・サーベイや感度解析機能を提供し、評価作業を支援する機能が求められる。また、化学反応を視覚化する手法の開発も大切であると考えられる。

地下水の流れについては、人工バリア内での地下水の動きは極めて遅いことから、主な物質移行機構は拡散になるので、ガラスの表面での核種の溶解度、及び緩衝材中の分散係数が物質移行評価の重要な評価パラメータである。ガラス固化体、及び人工構造物の特性は人工的に決まるため、この部分には不確定要素は少ないが、浸出評価における不確定要素は主に自然環境の不確定性による。従って、長期に亘る地下水流れを視覚化し、前述した化学過程を重ね合わせる等の表現技術の開発が対話シス

テムに必要である。それにより、非常に緩慢な拡散中心とした流れに添って反応する過程の視覚化表現が期待できる。それは、核種の放出と輸送の過程をシミュレーションすることを目標としている地層処分ヴァーチャル・エンジニアリング・システムの対話機能開発の重要な課題と考えられる。また、その手法は次項で触れる地層中での核種移行評価についても同様である。図2. 5-4から図2. 5-6に試作した仮想処分システムデータより視覚化した、処分場近傍の仮想景観を示す。

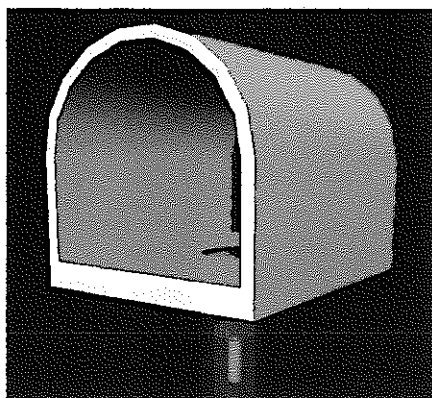


図2. 5-4 処分場近傍（5 m立方）

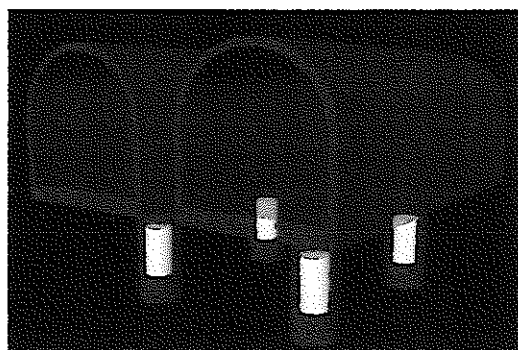


図2. 5-5 処分場近傍（10 m立方）

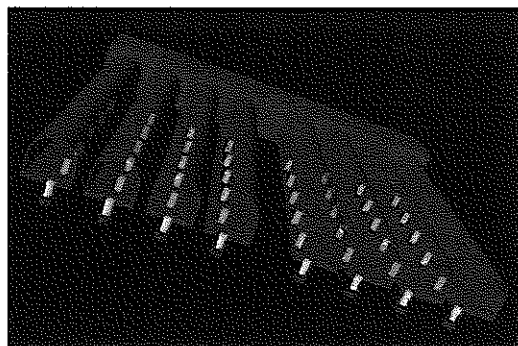


図2. 5-6 処分場近傍（100 m立方）

### (3) 地層中での核種移行評価

地層中での核種移行評価も基本的には核種浸出評価と同じ構成である。しかし、地層中においては評価上最も重要な部分は地下水の動きである。特に、破碎帯等の母岩に比べ極端に水が通りやすい、いわゆる水みち（水理）の分布である。また、母岩の間隙水は一種の貯留タンクであり、母岩の中への分散速度も重要な評価要因である。従って、地形、地層構造、母岩及び水みちの透水性、母岩中のイオンの分散係数が主なパラメータであると考えられる。

地層中での化学反応モデルでは、可逆反応モデルと鉱物中に取り込まれ、実質上、溶液系から不可逆的に取り除かれる鉱物化モデルの組み合わせで考えることができる。原子炉によって著しく溶解度の異なるNp, U, Trにとって還元性物質の存在は評価上重要である。即ち、溶解度、鉱物化の反応速度、還元物質の含有量とその還元特性が重要な評価要因である。これらの要因にまで立ち入った情報を自然環境で得るには、更に反応の種類、基礎データ（熱力学データ、反応速度等）に関する研究が必要であり、評価環境を想定して必要な吸着分配係数、溶解度を実測し、より汎用性のある基礎データとしてまとめていくことが必要である。

ソフト面では、PHREEQUE等で代表される多成分系での核種の安定な存在状態を求める計算コードの開発を進め、更に、反応速度、濃度依存性等を考慮した吸着移行評価コードを充実し、組み合わせていく必要がある。データベースについては、熱力学データや吸着係数の収集が考えられる。

図2. 5-7に試作した仮想処分システムデータより視覚化した、処分場全景の仮想景観を示す。

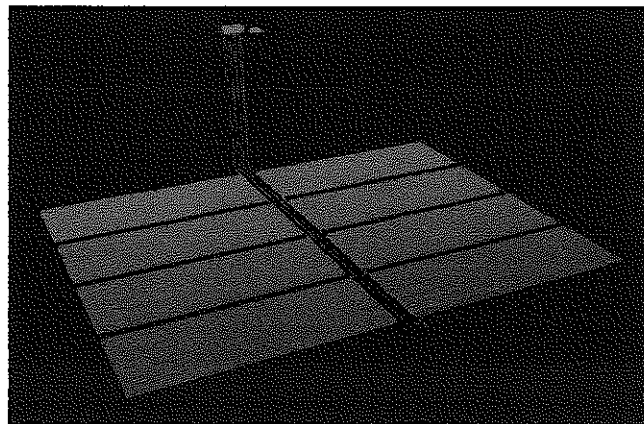


図2. 5-7 処分場全景の仮想景観

### (4) 生物圏中の核種移行評価

生物圏に到達した放出核種が人体に至るまでの過程は、原子炉、再処理施設等の地上施設から放出される核種の挙動と類似であり、そのモデル化及び関連パラメータについては多くの研究が行われている。そのほとんどが適用可能と考えられる。図2. 5-8に地層中での核種移行経路概念をしめす。

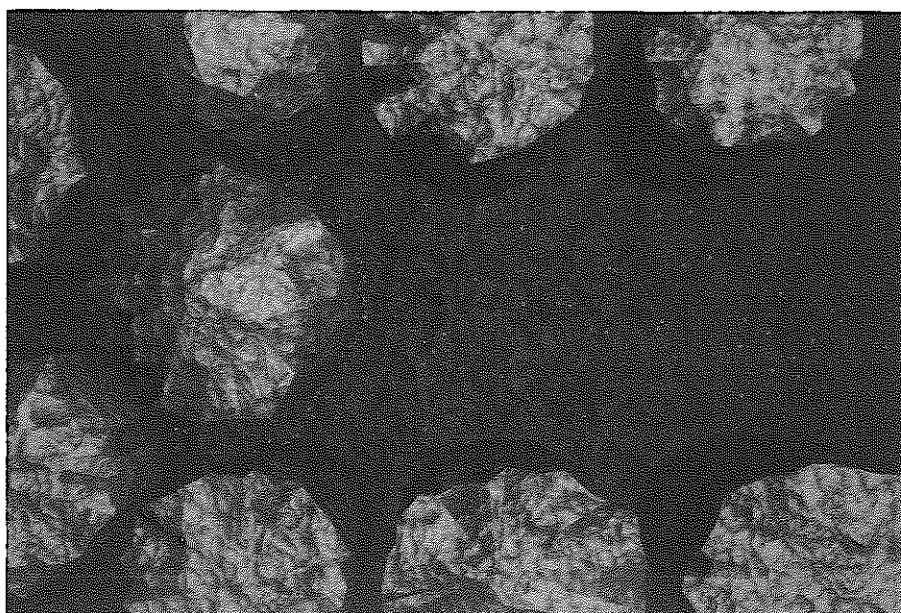


図 2. 5 - 8 地層中での核種移行経路概念

ただし、TRU等の長半減期核種に関する情報、データはまだ十分ではないので、この分野に関する一層の研究が望まれる。従って、生物圏への評価に備えて核種の崩壊等のデータベースや移行率、量等の作図機能等が対機能に求められる。

#### 2. 5. 6 パラメータの不確定性

地殻変動、地震等の接近シナリオと呼ばれる範疇に含まれる自然現象は、確率的事実であり、いつどこで発生するかを決定論的に予想することはできない。しかし、過去の発生の統計とプレートテクトニクス仮説から、ある程度推定することができる。その手法として、第四紀における地殻変動の統計的データを使い、長期評価に適合するデータを得る。これらの現象による大きな影響は、サイト特定で避けられるが、地震や雑続的隆起沈降の影響は考慮しなければならない。これらは、地下水による核種移行シナリオに影響する不確定性である。従って、対話システムには、将来これらの地殻変動や地震等の事象を与えられる機能が求められる。

地層の透水性、地下水の組成、岩石の組成等、地層の特性は、原理的には処分サイトが決定されれば測定によって求めることができるが、サイトを破損しないで行う測定精度には限界がある（掘削影響）。そこで、一般的に、類似地層での観測結果の統計処理により、それらの特性値を求めざるを得ない。具体的候補サイトが決まっていない現時点での評価では、地層の大分類の特性の統計的扱いをすることになると考えられる。従って、対話システムは、仮想のサイトにて評価シミュレーションを進めるための情報として、前述の地下水や岩石の組成等、幅広く情報をデータベース化して参照を可能とするとともに、地形のオブジェクトや設計する処分システムを仮想環境として、コンピュータ内に、それらデータベースと関連させて構築できるものであることが望ましい。それらを構築する上での不確定な事項として下記が上げられる。これらの不確定さの原因については、当面の間、事実情報を詳細に記述するデータベースを構築して、優れた対話機能により随時評価することが望まれる。



### (1) 人工バリア

材料及び施工等の品質管理上の不確定性がある。地層処分システムでは、地層特性の不確定性がそれに比較して大きいので、その不確定性は無視できる。更に重要なことは、人工バリアの健全性の不確定性は、環境条件の不確定性に起因すると言うことである。特に経時変化が重要である。例えその経時変化が予測できても、長い時間の経時変化であれば、その条件での実証試験ができなく、モデルを使った予測に頼らざるをえないと思われる。

### (2) 生活環境

遠い将来の人類の居住状況を予測することは非常に困難である。そこで、現在の社会が変わらないものと仮定する。従って、サイト固有のデータを使うことは難しいであろう。陸地か海洋か、砂漠か農耕地か、井戸の利用か川の利用かと言う程度の区分で被曝経路シナリオを作り評価することになると思われる。

### (3) 測定精度

以上のように、パラメータそのものの特性として幅を持っている。また、測定技術が不十分である場合、技術はあってもサイトの損傷を避げるために使えない場合がある。後者の場合は類似サイトを詳細に調査することでその精度を高めることができる。

### (4) 長期の不確定性に対する保証

長期の事実を限られた期間で実証することはできない。それを補う一つの方法としてナチュラル・アナログ研究がある。評価事象の機構のモデルを作り、その事象のナチュラル・アナログを探し出し検討することにより、実証的データを得る。この方法の難しさは、それが置かれた環境条件の経時変化を把握することが困難である点である。

## 2. 5. 7 総合的性能評価

前述のように、処分システムの安全性評価の判断基準は、処分に起因するリスクが社会的に許容されるリスクレベル以下であるかどうかにおかれるものと考えられる。そして、この判断のために総合評価が考えられる。総合評価は2種類あると考えられる。

第一に、代表的な(適切な)処分シナリオについて適切な移行シナリオを選定し、決定論的評価を行う手法である。決定論的評価に於ては、シナリオが適切かどうか、評価結果の信頼性が十分かどうかについて議論がなされるであろう。それは長期にわたる諸条件の変化やモデル・パラメータの持つ不確かさを考えねばならないからである。そのためには、多方面に渡る情報を評価とともに参照しながら作業をすすめる必要がある。このような決定論的評価を検証するための情報を効率的に提供することは対話システムの重要な機能要件である。

もう一つの方法に確率論的評価がある。それにより、処分システムの安全性の規制的観点からの評価は、決定論的評価手法によって行われ、確率論的評価手法は、決定論的評価を補完するものとして位置付けられると考える。確率論的評価手法は、放出核種の移行過程の変化確率、及び移行過程を支配するパラメータの自然的、社会的条件の変化に伴う変動、または情報の不十分さによる不確かさに

よる最終のリスク評価結果の不確かさを定量的に解析するものである。

更に、評価結果の不確かさに大きく寄与するパラメータを明確に示す機能、いわゆる感度解析機能も持つものであると考えられる。また、確率論的手法は、シナリオを簡単にネットワークで画像表現し、操作を視覚ベースで行うシステム化の手法を提供するものとしても有効と思われる。

例えば、評価モデルとしては、廃棄物からの核種の浸出は一定割合で一定期間続き、緩衝材中の移行は拡散のみに支配され、地層中は一次元吸着平衡を保ちながら移行し、岩盤中の間隙水へ拡散し、飲料水として人間に到達して放射線被爆をもたらすという簡単なシナリオである。確率論的評価手法の開発では、手法そのものの開発とともに、入力データベースの開発も重要な問題である。そして、入力データの確率分布をできるだけ正確に知ることが必要である。

## 2. 5. 8 評価作業の不確かさ

本システムを操作して性能評価シミュレーションを行うにあたり、最終的な目標となる生物圏からの放射性核種の隔離を多重バリアが機能として確保する過程の作業に下記の人的な不確かさが含まれると思われる。それらはシナリオの選択と内容に係わる事項や、数学モデルの数式化と計算等、凡そ以下に示す領域に含まれる不確かさである。

### シナリオの選択と内容

#### 処分システムの将来における状態

評価モデルと計算（数学モデルの数式化と実際の計算）の数学モデルの数値解析上の不確かさや、

プログラミングの不確かさ、精度計算上の限界

計算パラメータや熱力学データ等のデータ自信に内在する不確かさ

データの不足、測定誤差、専門家の仮定、等

これらは、確率的な変動性による不確かさと知識不足からの不確かさに大きく分類されている。性能評価においては、容器の腐蝕や長期に渡る放射性核種の地層から生活圏への移行など、将来に互る条件についての知識不足に依る不確かさが重要な研究課題となる。

また、不確かさを見積もる確率論的不確かさ評価は、主観的な確率を用いて行われていると考えられる。さらに、シミュレーション・プログラムの品質やデータの信頼性等、ソフトウェアに含まれる不確かさは、地層処分ヴァーチャル・エンジニアリング研究のような、情報処理に基盤をおいた性能評価にとって重要な課題となる。従って、ソフトウェアの不確かさは、前記の2つの不確かさ以外の第3不確かさと考えられる。

不確かさの評価は、サイトの選定と貯蔵所の設計研究などに用いられ、システム性能評価の結果において、予想の信頼性と精度、及び信頼度を示すことに特徴がある。確率論的不確かさ評価は2種類の情報を同時に与える。それは主観的な信頼性の有る期間と不確かさのランキングである。前者は、サイトや貯蔵所の候補間の比較に要求されるもので、後者は、線量など評価値における不確かさに対して最も寄与する定量化された不確かさのランキングを示すものである。また、ソフトウェアの不確かさについては、別途複雑度解析等の定量的ソフトウェア品質評価技術の研究が進められている。

シナリオやモデル化における不確かさは、主に定性的な方法において扱われる。例えば、FEP手法が上げられる。それは組織的に良く文書化された過程が要求される。計算モデルに対する定性的な保

証過程の展開的な適用が重要であり、性能評価の結果が規制の目的に使用される場合、特に問題となる。パラメータとデータにおける不確かさは次の方法を用いる際に定量的に扱われる。

- ・表面応答解析
- ・モンテカルロ解析
- ・微分解析
- ・確率論的解析

不確かさ評価に至る前にスクリーニング解析を行うことにより計算時間を最小化できる。不確かさ評価結果に含まれる感度情報は、データの不確かさを減らす目的で行う研究のガイドとなる。この手法の採用により、次第に、より複雑な方法を用いる必要性が出てくるにつれ単純性が擁護されている。

例えば統計的サンプリング法がそれぞれ特徴を持って開発されている。

- ・単純なランダムサンプリング : 単純で、健全な方法だが、多大な計算時間を要する。
- ・ラテン超多面体サンプリング : 単純なランダムサンプリングよりも計算時間は少ないが使用する際に注意を要する。
- ・インポートランスサンプリング : 更に計算時間を減少でき、解析をより関心ある領域に向えることができる。

不確かさの評価については、類似分野として、原子力プラント、再処理、廃棄物処理の評価における不確かさ評価で用いられた方法には共通点が見られる。すなわち以下のものがある。

- ・入力データを含むパラメータにおける不確かさに関する方法
- ・主観的な確率密度関数を用いることによりパラメータの不確かさを表わす方法
- ・単純なランダムサンプリングないしはラテン超多面体サンプリングを用いて確率密度指数からサンプルしたパラメータセットを出力するモデルの計算法
- ・メジアン、ミーン及び95%の信頼値により表わされる不確かさを出力する方法

以上のように不確かさは、性能評価シミュレーションにとって重要な対象である。従って、対話システムは、この不確かさについて出来るだけ多くの情報を提供するように設計開発されなければならない。それは、情報の検索、知識ベース等による判断、補足的な数値計算を含むものである。

従って、それらの情報処理機能をもって、およそ以下に示す議論について支援することが求められる。それにより、性能評価シミュレーション全体の有する不確かさをスクリーニングする機能が必要である。

議論1 : 決定論的、確率論的、性能評価法の相対的な役割は何であるか？

決定論的な最良の評価と、パラメータの不確かさについての確率論的な処理法を結び付けて考察することが必要

議論2 : 処分システムの性能評価を実行する時、確率論的不確かさと決定論的不確かさを区別して扱う必要があるか？それとも合わせて扱うべきか？

議論3 : 定量化する必要のある全ての源に含まれる不確かさは、どの様な範囲に渡っているのか？

例えば、モデルとシナリオに依る不確かさを定量化する方法を探す必要がある。

議論4：不確かさ解析の結果をどの様にリスク評価や結果の信頼度提示に用いることができるか？

議論5：不確かさ解析に用いる主観的な確率密度関数の定義は、どの様になされるべきか？

例えば直接測定が不可能な時、専門家の判断を用いる場合に問題となる。

議論6：時間に依存するプロセスは、どの範囲に対して性能評価に取り入れるべきか？

議論7：異なるパラメータ間の相関をどの様に見積もることができるか？

議論8：リスクと線量率を評価する時、不確かさパラメータを扱う確率論的手法は規制側による決定を助成することに対してのみ適用されるのか？それは、処分に関する安全性のレベルを大衆に知らせるために用いるにはあまりにも複雑ではないか？

以上の議論から、地層処分ヴァーチャル・エンジニアリング・システムの開発においては、その対話システムの支援機能として、以下に示す要求にそった情報サービスを行える機能を開発する必要がある。

要求1：定量的な手法と非定量的手法を結び付けるべきである。

要求2：不確かさ解析の種類と型式は規制側の要求により決定されるべきである。

要求3：決定論的な計算と、確率論的不確かさ解析を結び付けることにより全体的なシステム性能評価を導く一般的なガイドラインとすべきである。

要求4：感度解析は、不確かさ解析に対し密度な関係がある。

要求5：異なる不確かさパラメータ間の相関を扱う方法の開発が必要である。

要求6：幾つかの二者択一的な統計的サンプリング法がメリットを持っており利用可能である。

要求7：主観的な確率密度関数の形において、パラメータの不確かさの表現がシステムのなアプローチに必要となり、特に主観的な専門家の意見が取り入れられる。

要求8：コードの質的な評価がモデルの不確かさを減らすのに役立つ。

要求9：ベンチマークによる検証と確認試験が不確かさを減少させることの助けとなり、モデルのユーザーの信頼度を向上させる。

要求10：システムのなアプローチは不確かさ解析を指導することに適用されるべきである。

### 3 シミュレーション技術研究

地層処分システムの性能評価のための、複雑なシミュレーション機能を開発する準備として、幅広く関連シミュレーション技術基盤の研究開発を行った。シミュレーションの動作は、FEP (Feature, Event, Process) に代表される評価シナリオが基盤となって、実行に必要な複数の計算モデルに対応したオブジェクトが組み立てられ、実行プロセスとして、並列にアクティブとなった状態で開始すると考えて研究した。

平成8年度は、ネットワークレベルで並列処理を行なう基盤技術である、MPI (Message Passing Interface) 等の応用技術の蓄積と開発環境の整備、及び並列にメッセージを交換するオブジェクトの組立技術の蓄積、シミュレーション結果の3次元動画化表示処理等、シミュレーション・システム開発における共通な技術について試作を行い検討した。

#### 3. 1 仮想世界における実験の概念

地層処分ヴァーチャル・エンジニアリング・システムのシミュレーション機能は、評価シナリオが基本となって、実行に必要な複数の計算モデルに対応した独立したオブジェクトとして動作する個別計算プログラム (エージェントと呼ぶ) が、複雑に組み合わせられ、並列・協調して動作するアーキテクチャを想定した。それはまた、仮想世界における数値実験環境として考えた。

従って、核種漏洩と移行評価に係わる広域的な問題を確率過程として扱う確率論的リスク解析と、個別の事象を有限要素法や有限差分法で扱う決定論的現象過程の解析という、2つの手法の中間に位置 (従来の決定論的モデルより自由度が高く、確立論的モデルほど確立的ではない中規模の複雑度を扱う決定論的モデル) する計算による手法と考えられる。

##### (1) 新たな実験方法としての期待

地層処分ヴァーチャル・エンジニアリング・システム開発の期待は、コンピュータにより構築された仮想世界において、複雑な現象をシミュレーションにより再現して観察することにより、新たな定式化を期待することである。このような戦略においては、コンピュータ利用に係わる様々な最新の技術や情報科学・理論が、その世界を構築する基盤要素となっている。

また、コンピュータを利用した仮想世界に構築された地層処分システムを対象に、シナリオに添って複雑にモデルを組み合わせ、それらを協調させた新しいシミュレーションにより複雑な現象の解明が期待できる。そのような手法をマルチ・エージェント・モデル協調計算手法と呼ぶことにする。その場合、シミュレーション・ソフトウェア開発手法は、従来の手法と大きく異なるであろうことも予想される。なぜならば、従来のコンピュータ・シミュレーションは、実験や理論といった活動を支援する道具としての位置づけで開発利用されてきたものであるが、ヴァーチャル・エンジニアリングにおいては、その実験や理論化 (定式化) の場を、コンピュータ内部に構築された仮想世界に移そうという思想が根幹にあるためである。このような問題は、意思決定に係わる仮定性 (hypotheticality) の問題であると考えられる。それは、対象に対する判断を下さねばならないとき、その判断をする以前に、状況を知るうえで必要な情報を、実験や検査は充分に得ることができない問題に対する類推問題ともいえる。それは、近年、複雑系と呼ばれるシステムの振る舞いについて、検証したいと思うあらゆることに對し同様に適用されている。

例えば、大気中に放出される二酸化炭素の量が、現在の2倍のレベルに達すると仮定しよう。すると、今から50年後の地球の平均気温にどのような影響が及ぶのだろうか。これは、地球温暖化の問題を研究する際に気候学者が用いる標準的なシナリオである。この問題に対する答えは数多くある。全く影響がないというものから、気温が摂氏5度上昇するというもの（いわゆる温室効果）まで、様々な研究グループがそれぞれの推測を提案している。実際には誰にもわからないが、数多くの人々がそれに関心を抱き予測しようと試みているが、全体の予測を今までの方法では正確に行うことはできない。何故ならば、二酸化炭素の量の増加に伴う気候の変動は、非常に広範な、地球のダイナミクスに係わる現象が全体として増加後の現象に向かうのであり、個別現象の観測や予測では気候全体の質的な動きが捉えられないからである。そのようなアプローチは、あたかも車を構成するボルト1つの引っ張り強度から、そのボルトが利用されている車の性能を予測するようなものであると考えられる。

地層処分性能評価のような環境問題は、仮定性に基づく問題の代表的なものであると考えられる。高レベル放射性廃棄物地層処分の性能評価は、環境中に建設する処分システムという人工物の環境に及ぼす問題（結果的には、文化・経済を含む社会システムへの影響問題）と考えられる。このような問題は、純粋な自然現象ではなく、人の活動、即ち社会システムの活動が及ぼす自然環境への影響として捉えるべきである。そのような意味では、例えば高レベル放射性廃棄物処分に係わる地下水の地球化学的特性問題は、一般の地球化学とは区別して（人工放射性物質をリソースとして扱うという意味で）社会地球化学という範疇を考えるべきである。いずれにしろ、そのような複雑な問題にアプローチする時、始めに考えなければならないことはモデルへのアプローチである。モデルへのアプローチは、研究をすすめて行く上で基本的な考え方となるのであるとともに、ソフトウェア開発に於ては具体的なアルゴリズムやアーキテクチャを左右する重要な課題である。

## （2）モデル表現

例えば、飛行機の風洞試験をしている人にとってのモデルとは、飛行機の胴体のように現実に存在する物体のミニチュア版のことである。いずれの場合にも、それらのモデルは、数学の記号やコンピュータ・プログラムとは何の関係もないこのようなモデルは、通常、実験モデルと呼ばれている。それは、現実的対象が有する特徴を表現したり現実を物質によって表現しているからである。この種のモデルは、直接実験によって答えを出すタイプの問題に使うものであり本研究の対象ではない。

本研究の対象は、物質ではなく純粋な情報から構成されたモデルである。それは、現実を物質ではなく記号で表現したものである。そのようなモデルの領域には、区別しなければならない様々なタイプがある。例えば数学にはモデル理論と呼ばれる研究分野があり、そこでは論理モデルというタイプが扱われる。図3. 1-1は地下水シナリオに関する論理モデルの領域について示している。

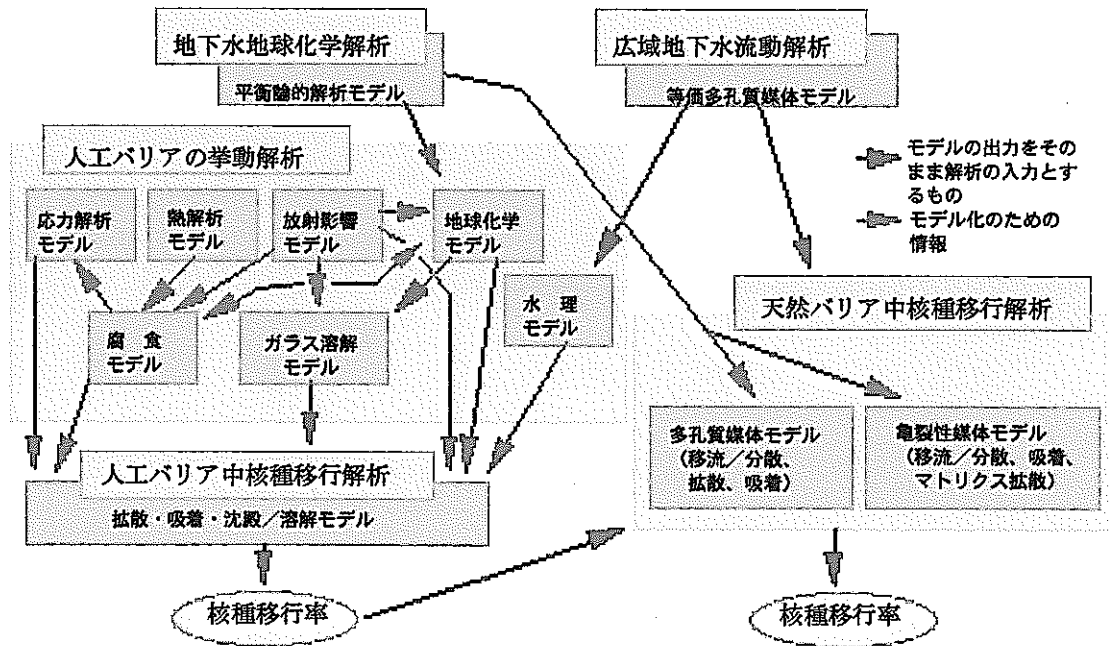


図 3. 1 - 1 地下水シナリオに対する解析モデルの体系

論理モデルは、まず公理と推論規則とを持っている。その公理から出発して推論規則を適用すると、その公理から新たに論理的に正しい記述を作り出すことができる。しかし、この段階で、公理と演繹規則から導かれる記述は、抽象的な記号の羅列、すなわち形式的な統語的体系（規則を使って機械的に作られた体系にすぎない。数学記号による表現はその例である。それは抽象的かつ一般的現象記述であり、特定の条件（シナリオ等）によって定まる系の動的状態を具体的に知りたい場合は、何らかの手法で条件付きの解を得なければならない。

### (3) 数理モデル

数理モデルの理論は、この抽象的な公理体系の性質を忠実に反映する操作規則を持った、具体的な数学体系を発見することである。言い換えれば、それは形式的な統語的体系に意味を付与することである。しかし本研究の課題は、厳密で抽象的な論理モデルとの中間にある数理モデル（詳しくは、計算可能な数学的かつ計算科学的モデル）と呼ばれるモデルである。数理モデルは、近似計算可能な幾つかの方程式により与えられるのであり、コンピュータを利用することにより、ある特定の条件についての振る舞いを計算結果で得ることができる。

### (4) 計算科学的モデルの主張

以上述べたことをまとめると、モデルは少なくとも3つのタイプに分類できる。すなわち実験モデル、論理モデル、数理モデルである。本研究で主に関心を寄せるのは、論理モデルと数理モデルであり、特にコンピュータを利用するモデルである。このようなモデルを利用して、多くの種類の放射性

核種をエージェントとして考えれば、それらの収着や沈殿、移行をエージェントの行動と見立てるモデル化も考えられる。恐らく、それは化学の説明に用いられる教材モデルと多くの共通点をもっているであろう。そのように、モデルのタイプには様々なものがある。それらは、それぞれ独自の特徴と利点や欠点がある。

以上の考察から、従来の実験と理論という2つの研究領域に加わる第3の領域として、コンピュータによる仮想世界の数理モデルを基盤とした複雑系の領域が上げられる。それは、地層処分ヴァーチャル・エンジニアリング・システム開発の最も基本となる動機と期待であるとともに、計算科学技術の根本思想に大きく影響する思想であると考えられる。それはまた、従来取られてきた要素還元論的戦略の限界に対する挑戦として期待することもできるであろう。そのためには、以下のような従来の科学思想に関する反省的な考え方に基づく主張（新しい方法に対する期待）が上げられる。

- 主張1：自然は複雑であり、人類はそのごく表層部分を明らかにしたにすぎない。
- 主張2：単純な還元論的思考の限界は、気象予測モデル開発に端を発するカオス理論の中で明確にされた。現象理解や、定式化と現象予測の間にも、単純な還元論的思考を働かせることはできない。
- 主張3：複雑現象に対して、単一の力学的予測は不可能である。従って、個別事象に対する力学的予測はあきらめなければならない。従来の観点では、力学的予測をあきらめることは力学的法則が存在しないことを意味した。しかし、新しい観点では、力学的法則がたとえ存在したとしても力学的予測はできないことが示される。同様のことが大気や地下水流れ等環境モデル全般を支配しているかも知れない。
- 主張4：観測者の視点を明示的に自然の中に導入することにより、純粹客観でも純粹主観でもない相対的な観点が入り込む。観測結果は、観測手段に相対的にしか意味をもたないので、単純に観測結果を「真理」と認めることは危険である。観測結果は、たしかに特定の観測手段にもとづく「事実」を与えるが、それは観測行為に対する相対的な真理である。このようにわれわれは、厳密には実験による客観性の獲得をあきらめざるを得ない
- 主張5：純粹客観の視点は、論理によってのみ与えられ、観測によっては与えられない。すなわち純粹客観の世界は論理的な世界である。論理を信じ、数学的構造を客観性の基準に与えることができる。コンピュータでの実験やシミュレーションを複雑現象の理解にとって非常に大切なものととらえるが、それは次のような理由による。コンピュータ実験ではコンピュータは素過程として論理的に動いている。しかし、一方で、コンピュータ内で起こる現象を理解したり、定式化したりするとき必要になることは、それら論理の筋をすべて追うことではなく、コンピュータ内の現象を通して、養われた新しい直観にもとづくことである。このとき、まず信じることは、素過程での論理の正当性である。これを感じた後、今までの物理学的直観とは異なるかも知れない、新しい直観を養わなければならない。たとえある事実が、物理的直観に反したとしても、このような意味で「数学的事実」であれば、それを真理とみなすことをためらってはならないであろう。このように、複雑現象に対する数学的定式化は、コンピュータの中で形式化するのが適当である。
- 主張6：コンピュータによって創られる「人工的」な世界と「自然」を、絶対的な基準で区別することは不可能である。コンピュータの中で世界を構築するという、工学的な手法は複雑現象の理解へと到達するおそらく唯一の方法を与えるであろう。複雑現象をコンピュータの中で形式化、モデル化することで新しい型の「予測」が可能になることをみる。それは、系の構造と歴史に対する可能な過程を提示することで、単一現象の「先」を予測するのではなく（それ自体は、すでに示したように不可能である）構造と歴史の集合を予測するのである。新たな方法を生み出す契機を与えたとみるべきだろう。カオス、複雑現象の理解をめざした研究から、今まったく新しい自然認識へ到達したと考える。それは人工世界と自然との相対化であり、観測行為の新しい定式化をめざすものである。



地層処分ヴァーチャル・エンジニアリングで目指すものは、コンピュータによる仮想世界の構築とそこで生じる複雑な現象の観察という新しい定式化である。ここでは、コンピュータ・シミュレーションとコンピュータ実験は区別して扱う。前者は、従来よりとられてきた自然界の観測から定式化したモデルを近似的に数値計算により解く行為であり、後者は、コンピュータ内部の仮想世界で生じた過程を観察して定式化する行為である。地層処分ヴァーチャル・エンジニアリングは、この後者をアプローチを可能にする具体的なシステムを開発することである。

#### (5) シミュレーション技術開発の方針

地層処分ヴァーチャル・エンジニアリング・システムのシミュレーションを取り巻く技術は、具体的にはコンピュータ・ハードウェアとアーキテクチャ、及び言語とコンパイラ、モデル化とアルゴリズム開発を包含していると考えられる。それは、システムに必要な諸プロセスの実行にかなりの計算能力が必要になると考えられる。とりわけ、化学的変化のモデル化、および処分システム構成要素の構造分析、気候・地殻変動による変化の予測と併せて考える。また、地下水流動のシミュレーションのようなアプリケーションでは、計算能力に対する要求（計算量）がきびしい。さらに、コンピュータによる精巧な可視化・仮想現実感に対するデータ処理要求が、これらのな計算能力に対する要請に加えられる。

コンピュータ・ハードウェアについては、従来型計算アーキテクチャの性能は、プロセッサスピードによって決まってくる。これまでは、単一プロセッサによるアーキテクチャの性能に進歩がみられた。ハードウェア開発は、今日のマイクロ電子工学技術の物理的限界に到達しようとしている。従って、より大きな計算能力を達成するために、並列計算が必要不可欠になると考えられる。プロセッサ・スピードの他にも、プロセッサとメインメモリおよびディスク間のデータ転送スピード等により、多くのアプリケーションに限度が設けられる。並列計算アーキテクチャを利用すると、計算能力が機械の規模に比例する可能性が生じる。この可能性は、とりわけデータベース・アプリケーションに、また大量なデータの可視化処理にも影響するであろう。

ソフトウェア開発においては、これまでのソフトウェア構成方法は、近年のオブジェクト指向による新しい手法との間にはかなりの開きがあると考えられる。経済性の面からは、既存のアプリケーションとデータベースを全てオブジェクト指向システムとして再設計することは、当面適当でないと考えられる。当面の間、システム開発は旧来の方式と新たな方式の混合パラダイムを使って作業を続け、諸問題に対処していかなければならないであろう。

例えば、既に定まったシナリオを評価する数理モデル（コード）としてできあがったものは、より高度な視覚化インターフェイスを接続して高度化しアルゴリズムは再利用すべきである。一方、始めから複雑な問題であるオーバパックの腐食と地下水・ベントナイト相互作用などについては、複雑系のモデル開発課題として考えるべきである。いずれの場合も、システム全体の設計はマルチ・エージェントなモデル協調計算としてすすめ、既存の数理モデルは、エージェントを特徴つけるメソッドとして組み込む方式等を研究開発すべきである。

このような状況を避けて通れない現状は、アプリケーションやデータベースへの既存の蓄積を再利用するオブジェクト指向開発手法を採用する戦略をよく検討することを基本的に要求する。採用すべき戦略は、オブジェクト指向方法論と超高速計算技術のような関連技術における進歩の良さを無理に

出したり、あるいは、排除したりするものであってはならない。代わりに、オブジェクト指向プログラミング言語や、ツールを使って精選したアプリケーションを少しずつ書き直していくより、オブジェクト指向アプリケーションと非オブジェクト指向アプリケーション、サービスとデータベースを統合するオブジェクト指向ソフトウェアを開発することに主眼をおいて、ソフトウェア開発戦略を立案すべきである。

地層処分ヴァーチャル・エンジニアリング・システムの動作に関する様々な点で新世代の高性能な並列計算を用いることの実現可能性も評価されている。少なくとも、原理上はソフトウェア階層全体にわたって並列アーキテクチャの適用が必要であるということである。これらを踏まえて、システム開発のためのフレキシブルで説得力のある方法論の立案が必要である。それは、これまでの研究から得られた経験に基づいてはいるが、利用可能な最も進んだ意思決定ソフトウェアを組み入れたものであるだろう。

例えば、「インテリジェントソフトウェア」（ニューラルネットワークに連結した遺伝的アルゴリズム等）を使用して操作の多くの部分の自動化や情報支援を図れば、対話システムの機能の最適化が容易になり、エラーを引き起こす潜在性が減少し、人的資源を最大限に活用することになる。この方法論の内容は以下が考えられる。

- ・各段階における不確実性の明示表現
- ・分析の進行にしたがってシミュレーション感度を調べるため、インタラクティブな可視化製品の使用
- ・システム展開中のプログレッシブ（漸進的）、サイクリック（巡回的）、エピソード（挿話的）変化を扱うため、動的シミュレーションの使用
- ・定常状態進化の考察から、主な分裂事象の知覚された重要性の認識への移行
- ・異なる世代のコードの結合使用を可能にするため、各技術領域における開発状態の違いを考え、ジェネリック・データベース・エンジンに連結した共通ソフトウェアインターフェイスの使用

それらは、地層処分ヴァーチャル・エンジニアリング・システムの利用者が数値モデルを選択するときの手引きし、数値計算実行に必要な入力データを検索し、対話型グラフィクスで計算を始め、シミュレーションの各時点の数値計算結果を評価すること等である。そのためには、一連の基準がコード選択過程で参照できるように設計されている必要がある。

システム開発においては、性能評価シミュレーションのために、システムの物理的および化学的、生物学的特質を表す典型的なデータを用いて、シナリオ全体をモデル化することが不可欠である。そのようなシミュレーションコードに必要なデータ選択と管理は、コードそのものの選択と同様に大切である。当面は、それらのコードや知識ベースシステムによって参照されるデータ全てに対する直接の操作を実現することは難しいであろう。あらゆるデータを蓄積し、管理するためには、個々のデータベースから開発統合化する必要がある。そのようなデータベースエンジンは、特定のタスクについて、関連データすべての位置付け、照合、抽出ができるものでなければならない。

また、将来の構想として地層処分プロジェクトの社会システムへのインパクトを評価するシミュレーションについても、研究課題としておくべきである。例えば、教育の第一歩、法律関係の調査を手

助けするための、また、社会受容性に関する業務を支援する一般情報の提供には、各々、異なるタイプのデータが異なる複雑なレベルで要求される。そのような問題には、複雑系のアプローチは有効なものと考えられる。

地層処分ヴァーチャル・エンジニアリング・システムのシミュレーションは、評価シナリオの決定、シミュレーションの実行、実行結果の整理等、シミュレーションに直接関連する操作に始まり、シミュレーション結果の妥当性等を検証・確認するための関連情報をデータベースより検索し、解析結果と比較評価する、等の様々な性能評価に利用されると予想している。

また、我が国の推進する高レベル放射性廃棄物処分システムの技術研究は、処分システム開発プロジェクトの各段階について行われると考えられる。例えば、現在実施されている基礎研究段階が終了するとサイト特定段階のプロジェクトが開始されるであろう。地層処分ヴァーチャル・エンジニアリング・システムは、それら地層処分プロジェクトのトータルなライフサイクルについて、目的に適合した結果を提供できなければならない。このため、その各開発段階に応じて評価結果の使用目的が異なると考えられる。そのような状況に柔軟に対応するシステム開発体制が求められる。

地層処分ヴァーチャル・エンジニアリング・システムの当面の目的は、第2次とりまとめ後である、第2段階で実施される処分サイトの特定、及び処分システムの概念設計のに係わる性能評価をするための情報を複雑なシナリオを組み合わせてシミュレーションし、提供することにあると考えられる。例えば、想定サイトでの処分環境影響の概要、処分システムの設計オプションに関する多方面からの性能評価等であると考えられる。

従って、地層処分ヴァーチャル・エンジニアリング・システムのシミュレーション機能は、地層処分した高レベル放射性廃棄物が超長期に亙り生物圏から隔離されること、言い換えれば人間生活圏の超長期に亙る放射線防護を目的とした、シミュレーションや数値実験を行うことを目的として開発すると考えるべきである。

### 3. 2 性能評価面から見たシミュレーションの目的

地層処分ヴァーチャル・エンジニアリング・システムの開発にあたっては、放射性廃棄物の処理処分の課題を解決するための、国の政策、安全基準、PA対策等の政治的なコンセンサスについて、常にそれを正しく反映した研究成果報告資料（CGやビデオ、CD等）等の表現技術開発が必要であり、かつ、固化体、容器、地層等の性質についての科学的な研究や、工学的な研究開発成果を総合的にとりまとめ地層処分情報として管理する機能を開発して行く必要がある。そのような観点から、システムを設計開発するためには、常にそれらの情報を把握し設計に反映することが大切である。ここでは、以上の観点から、高レベル放射性廃棄物地層処分の最も基本的な考え方である放射線防護基準に関する主要各国の考え方を整理し、シミュレーション機能研究の出発点とする。

#### 3. 2. 1 米国の放射線防護基準

米国は1982年の原子力政策法で処分サイトの選定から閉鎖までのスケジュールを詳細に設定した。許認可機関である原子力規制委員会は、地層処分基準（10CFR60）案を提案し、人工バリアによる閉じ込め期間、地下水が地表に達するまでの期間等、それぞれのバリアについて性能要件を詳細に決め

た基準を示した。図3. 2-1にそれを示す。

省庁/組織	文 書	基準/指針
NCRP	NCRP (放射線防護審議会) Report No.91、電離放射線被ばく制限に関する勧告 (1987)	p37-39 : 医療用を覗く人工放射線全線源による被ばく線量 <100mrem/y(1mSv/y)
NRC	10CFR Part 20 放射線防護基準	§ 20.1301 : 操業による公衆への全実効被ばく線量 <100mrem/y(1mSv/y)
NRC	基準案 10CFR Part 20 Subpart E (解体措置の放射線基準) 8/22/94	§ 20.1401 : サイトからの漏洩限度を決定グループの組織被ばく線量(TEDE)の平均値 15mrem/y(0.15mSv/y)とする。
EPA	40CFR Part 190 原子力発電所の環境放射線防護基準	§ 190.10 : 主な操業による公衆の被ばく線量 <25mrem/y(0.25mSv/y) (採鉱、輸送及び廃棄物処分場での操業を除く)
EPA	40CFR Part 191 使用済燃料、高レベル/TRU放射性廃棄物の処分の環境放射線防護基準	§ 191.13 : 核種ごとの集積漏出量 § 191.15 : 公衆の平均預託実効被ばく線量 <15mrem/y(0.15mSv/y)

図3. 2-1 米国の原子力施設放射線防護基準

処分場候補がユッカ・マウンテンに決められ計画が具体化されるに従って、柔軟性に欠けるスケジュールや基準が問題となり計画が難航してきた。そのため再検討が行われ、柔軟性をもって計画を進めるように1992年に原子力法が改定され、その結果ユッカ・マウンテンでの地層処分基準については、まず科学アカデミーが環境保護庁に勧告を提出し、それに基づいて環境保護庁が安全基準を示し、それに沿って原子力規制委員会が技術基準を策定することになった。

米国機械学会の廃棄物環境修復国際会議 (The international conference on nuclear waste management and environmental remediation) の1995年会議 (Waste Management 95) では、処分の実施に責任を持つエネルギー省が、科学アカデミーの検討を助けるために1994年4月にまとめた勧告の概要を「米国高レベル放射性廃棄物の許認可要件の国際的展望」と題して公表した。その中で以下の6つの基本条項を解説し、図3. 2-1に示す国内の関連基準と図3. 2-2に示す諸外国の防護基準との整合性について述べている。

国/機関	放射線基準	処分の時間枠
ICRP(Pub 60)	個人被ばく線量 (100mrem/y(1mSv/y) (住居のRn及び医療による被ばくを除く)	規定なし
カナダ	個人リスク $<10^{-6}/y$	10,000y ; 長期時間枠での評価
フランス	個人被ばく線量 $<25mrem/y(0.25mSv/y)$	10,000y ; 長期時間枠での評価
ドイツ	個人被ばく線量 $<30mrem/y(0.30mSv/y)$	10,000y ; 長期時間枠での評価
北欧諸国	個人被ばく線量 $<10mrem/y(0.1mSv/y)$ 個人リスク $<0.1mSv/y$ 相当のリスク	合理的に予測可能な時間
スペイン	個人被ばく線量 $<10mrem/y(0.1mSv/y)$ 個人リスク $<10^{-6}/y$	規定なし
スイス	個人被ばく線量 $<10mrem/y(0.1mSv/y)$ 個人リスク $<10^{-6}/y$	規定なし
英国	個人リスク $<10^{-6}/y$	10,000y ; 長期時間枠での評価

図 3. 2-2 各国の放射線防護基準指針

また、基準作成に対する要求は凡そ以下のとおりである。

- 要求 1 : 基準は健康ベースであるべきである。  
将来世代の集団への健康への影響として示すのが、公衆の健康や安全と比較しやすく判りやすい。
- 要求 2 : 基準は、処分候補地域の住人の保護におくべきである。  
環境境界に近づく人間、処分候補地域の住人、世界の人類、地下水、動植物等を対象にする方法があるが、処分候補地域の住人を対象にすることによりこれらを保護する条件を包含することになる。住人の保護という一つに基準を絞れば、許認可手続きで、その処分システムに従って、各バリアの基準を示すことになる。従来NRCが提案していたような個々のバリアの性能目標は、許認可基準に含まれるべきではない。
- 要求 3 : 基準は10,000年以上の期間についての性能を定量的に示すことを要求すべきではない。  
処分後10,000年以後にも、健康影響がある可能性があることはわかっているが、超長期の性能評価計算は不確実性が増しサイトや、設計を相対的に比較するだけの価値がなくなる。10,000年以上についての要件は定性的にすべきである。
- 要求 4 : 放射性物質の漏洩の起因による、将来世代の人間侵入の可能性を減らすことについての要件は定性的にすべきである。  
将来世代の人間侵入については、将来の技術や生活習慣を予測する必要があり、不確実性が大きすぎる。設計要件や制度的管理の定性的要件にすべきである。標識、土地利用制限、記録保管等詳しい要件を決めることは、処分場の閉鎖近くまで延期すべきである。それまでに得られる情報や技術をもとに適切な内容を決めることができる。
- 要求 5 : 基準の定量性は確定的な数値で示し、適応性の証明では確率解析が使われるべきである。  
10,000年以内での定量的評価の基準値そのものは、あくまでも一つの数値で示され、確率

的分布を持ったものであってはならない。性能評価解析では確率計算を行い、確率分布で結果が出る。その分布を基準値と比較することによって、低確率の高影響事象を考慮に入れることができる。

要求6：基準は、個人被曝線量で表されるのが適切である。ただし、確かな見解が規制や指針に明確に示されることを条件とする。

健康ベースの基準を表すには、許容リスク、個人被曝線量、核種漏出量、構成システムの性能、各人工・天然バリアへの要件等がある。いずれも許認可手続きで適応性を示すように表現されるなら採用し得る。

被曝線量基準を採用するには以下の3つの見解が明確にされる必要がある。

個人被曝線量限度は従来10CFRPart20にある公衆の個人被曝線量限度と同じにする。  
被曝線量限度はYucca Mountain地域に住む標準的住人に適応する。  
将来の生態圏に関する仮定（将来世代の人間の挙動）を特定する。

以上のエネルギー省勧告について同会議では、図3.2-2に示した海外の基準との整合性を次の3点から検討している。それらは、基準の表現形式、一次放射線被曝線量限度、要件における時間である。結論としては、エネルギー省の勧告は現有の内外の基準指針の範囲にある。

### 3.2.2 ドイツにおける原子力施設に関する放射線防護基準

ドイツでは、岩塩鉱への処分を主に考えて計画を進めてきたが、地元の下承が得られず計画通りには進行していない。Waste Management95では、岩塩鉱と合わせて花崗岩を候補岩種として対比しながらサイト選定基準、安全評価、及び地質調査について「高レベル廃棄物処分の安全基準の解析」と題して報告している。米国とは対照的で、地層処分の設計はサイト特有の地層条件によるので、定量的基準としては、放射線防護目標のみが原子炉安全委員会（RSK）の勧告として示されている。

それは、「全ての合理的な漏洩シナリオによる個人への放射線被曝線量は、0.3mSv/年を超してはならない。」とされている。この値をもとに候補サイトの調査を行い、地質及び地殻構造についての基準を導くとの方針がとられている。

### 3.2.3 スウェーデンにおける原子力施設に関する放射線防護基準

花崗岩処分を考えているWaste Management95では、「人間及び自然の保護」と題して、基本的事項について以下の通りの考え方を紹介している。

被曝線量の限度については、廃棄物処理施設、及び輸送システムは個人の被曝線量が50mSv/y、100mSv/5y、700mSv/（生涯）を超さないよう設計しなければならない。この値は集団集積被曝線量2person-Sv/y-GWeの限度の内に含まれる。公衆については0.1mSv/y以下とする。集団集積被曝線量は5person-Sv/（操業期間・GWe）、500年について平均1mSv/yを超さないようにする。

自然環境の保護とは、一般に人間の健康、生物種、生物資源の生産能力及び農耕地のことである。地層処分による影響評価の主点は生物種の保護にある。生物種を保護するには、個体ではなく集団であるが基準はこの保護基準で表される。そして、生体組織レベルで考えるべきである定量的基準としては、組織や成長過程で違いますが、実験データによると1から5Gy/yの範囲であるならば、多くの組織

について観測できる影響は生じていない。これらの実験研究は、通常低レベル長期被曝の場合には留意していないことを考えると100から1000の安全ファクターをとるべきである。1から50mGy/yを定量的基準とすべきである。処分場の実現性を評価する作業の場合は20mGy/yの値を使ってよいだろう。

超長期に互る観点として、処分場閉鎖後の放射性物質の涌洩については次の3つの期間に分けて検討する。

最初の1000年	: 最も潜在的危険性を考えなければならない期間である。
1000年から次の氷河期	: (10,000年のオーダ) Cs137、Sr90の寿命より短寿命の核種は減衰し、長寿命の核種が隔離の対象になり、期間の経過に従ってバリア、社会、自然を考える際の不確実性が増す。
次の氷河期以後	: 処分場を定量的に詳しく表すことは困難と思われる。 10,000年までの時期が対象であり、次の氷河期以後の時期については定性的判断になる。

将来社会は、その社会の意思での行動については責任を持たなければならない。将来世代の人間が意識的に行う人間侵入は防ぐことはできない。

以上の内容の報告の他に、「放射性廃棄物対策—国際的観点」と題して国際的動向も含めた紹介もしている。数年前に、OECD/NEAのRWMC（放射性廃棄物管理委員会）は、IAEAのINWAC（国際放射性廃棄物諮問委員会）と協力して、「長期の安全性は評価できるか」と題し、国際的に集約した見解を文書（Collective Opinion）としてまとめているが、これは世代間倫理等、地層処分的基本的な問題についての新しいCollective Opinionの作成作業が進行していることを紹介している。

### 3. 2. 4 サイト選定作業からの要求

サイト選定基準は、放射性廃棄物の地層処分に適したサイトを見つけ、選んで行くときの指針を示すことにある。サイト選定にあたっては、選定段階を次の4段階に分けてそれぞれの段階で行うべきである。それらは、概念設計段階、地域調査段階、サイト特性調査段階、サイト確認段階である。

サイト調査についての指針と調査して得なければならない科学的データとして、地質学的特性、自然変化の未来予測、地下水理、地球化学的特性、人間侵入に起因する事象、建設工事条件、廃棄物の輸送、土地使用及び社会的影響に関する指針とデータがあげられる。

それらのデータは、全体として科学技術的な面からの記述がほとんどを占めているが、サイト選定上しばしば問題になるのは、社会経済的な面、環境問題及び政治的な問題であることを重視しなければならない。サイト選定にあたり、あらかじめ制限がなければ、大きな領域から、特別のサイトを系統的に絞り込んで、それらのサイトが適切であることを確認できる。始めからサイトになる可能性が強いところがあることもある。これは、常識的に言われているような、広域調査で絞り込む方式のみでなく、最初に特定のサイトを決め、そこが適切であるかどうかを調べていく方式もある。

どのような方法で選ぶとしても、ベストのサイトを選ぶのではなく、安全上及び環境上の要求にあった場所から考えるべきである。サイトは、最初から必ずしもベストである必要はない。そして、最終的にサイトを決めるには、社会経済的な面、環境問題及び政治的な問題についての考慮に基づく判

断を含むであろう。従って、科学的判断のみで決まらないことを述べている。そのような、社会経済問題にまで影響する試行錯誤的な作業の効率化がヴァーチャル・エンジニアリングの開発思想に取り入れられる必要がある。

サイトを選定するときのガイドライン、及び必要なデータについて述べている箇所では次のようなことが上げられる。第一に放射性廃棄物の輸送について、処分場への廃棄物の輸送もまた処分場の場所についての公衆の合意を得るのに重要な要因である。場合によっては、新たなルートを造ったり、既存のものを改良する必要が生じる。サイトの選定では、輸送が地元との話し合いで重要な項目である。また、環境保護については以下があげられる。

坑道掘削やその他の建設工事での環境破壊
地域住民に重要な価値を持っている区域への影響
水源の破壊
植物・動物への影響、特に危険にさらされている種について

土地使用についての適切なサイトの選定に当たっては、将来の開発や、その地域での計画との関連で、土地利用や所有権について考慮する必要があると指針を示した上で、土地の管轄権や所有権は経済性や公衆の合意を得ることに関連して多くの国で重要な原因である。実施主体または国に土地の所有権があれば、サイト計画や労力を見積るのが簡単になる。さらに別の用途で土地を使用しているのを上げさせることに伴う問題を少なくする。国有地あるいは実施主体の所有する土地が、公衆の合意を得る上で有利であることを述べている。

最後に社会的効果については、サイトはどのような場合でも、その地域に処分場を決めることによる利益が増進されるべきであり、負の社会的効果は最小になるようにすべきである。一般論として、高人口密度の地域から離れたサイトに特典が与えられるべきである。しかし、社会基盤が変化しないようにし、労働力が得られる必要があると考え、地域社会での生活が破壊されないよう十分な配慮が必要である。

### 3. 3 性能評価シミュレーションへの要求

前項では、放射線被曝防護という最も基本的な観点から、放射性廃棄物処分について各国の基準とサイト選定に係わる基本的な議論を概観した。その結果、高レベルだけでなく、放射性廃棄物処分全般について言えることは、議論の対象が、人間社会活動の結果として生じる汚染物質が、超長期に亘り充分人間生活圏から隔離されることを確認する為の技術・経済面からの手法の開発であると言える。それは、最も費用対効果の高い放射性廃棄物の処分方法を策定する為の議論とも言える。このような議論は、極めて大規模で複雑な系の振る舞いの最適解を見いだすような問題であると考えられる。そのような問題は、個別に分解された問題について解明しても、それらを統合して全体を説明することは難しい場合が多いと考えられる。なぜならば、問題のシステムとしての質が個別に分解されると失われる為である。従って、問題に対するアプローチとして、コンピュータ内部に仮想環境を構築し個別に開発した現象過程のモデル等を試行錯誤的に議論とともに組み合わせ、シミュレーションを観察する問題解決戦略が考えられる。

地層処分ヴァーチャル・エンジニアリング・システムは、そのような観点から、従来シミュレーシ



オンや数値計算などと呼ばれてきた物理工学上の現象を、コンピュータにより数値的に計算模擬する技術をもとに、オブジェクト指向、並列計算法、認知科学等、革新するコンピュータ・サイエンスの成果を地層処分の安全性評価研究に応用し、処分の安全性に係わる技術論拠となる情報を多方面から総合的に評価することを目的として、専用に統合開発するコンピュータ・システムであるべきである。

それはまた、高レベル放射性廃棄物地層処分の性能評価にかかわる、実験の難しい長期（数千から数十万年）に亘る現象を考慮した各種のシナリオを複雑に組み合わせて、システムの長期的挙動の評価を行ない、地層処分の安全性に係わる技術を判断し、処分システムの設計の最適化をするための指針等を得るとともに、多分野にまたがる専門家が、自己の領域と他の領域との関係を容易に理解するための基盤を提供する場となることでもある。

例えば、地層処分システムの性能評価は、地震等の確率的に生じる事象による変化を除いて、地層中の環境が安定している場合、地層処分された廃棄体から放射性核種が放出され地下水により生物圏まで輸送されるシナリオを基本としている。そのような安定した地層中での環境に添って核種移行を考えた場合、それだけでも非常に複雑な環境評価モデル開発と評価の問題であることが解かる。

- |  |
|--|
| 領域1：ガラス固化体（崩壊熱モデル、放射性核種放出モデル、等）              |
| 領域2：オーバーパック（腐食モデル、腐食物ベントナイト化合物モデル、等）         |
| 領域3：ベントナイト（2層多孔質媒体地下水流れ、熱・オーバーパック腐食化合物モデル、等） |
| 領域4：人工バリア周辺（人工バリアと岩盤との熱応力モデル、コロイド、核種収着、沈殿、等） |
| 領域5：地層中（広域的地下水汚染物質移行モデル、亀裂水理、等）              |

実現象は上記のモデルが複雑に組み合わさったものと考えられる。評価シナリオは、例えば温度場の変化が力学系に作用し、地下水流動を変更することにより人口バリアの冠水や亀裂を生じ、結果として核種が滲出して輸送されるシナリオが考えられる。このようなシナリオにそったシミュレーションを対話方式で試行錯誤的に行なうには、下記の操作機能がアプリケーション・システムに要求される。

- |                                       |
|---------------------------------------|
| 操作機能1：シミュレーション・シナリオの一覧                |
| シナリオは、十分に評価され妥当なものとしてデータベース管理されていること。 |
| 操作機能2：シミュレーション条件の決定                   |
| シナリオの選択、個別評価モデルの選択、計算条件の設定等           |
| 操作機能3：シミュレーションの実行と制御                  |
| シミュレーションの実行、実時間視覚化、計算結果の解析等           |

以上の様に、個別モデルが複雑に協調したシミュレーション機能は、高レベル放射性廃棄物地層処分のような、大規模で複雑な系の問題に対するアプローチとして新たな結果が期待できる。

### 3. 3. 1 性能評価シミュレーションの基本的な考え方

性能評価の基準は、処分が公衆にもたらすリスクが社会的に十分許容されるレベルかどうかで判断することが基本である。従って、シミュレーションの最終段階の判断は、この社会的許容度となるであろう。実際、この社会的許容度は放射線医学に言う人体への影響という直接的（物理的）な尺度だ

けで判断することは困難であり、諸々の政治思想、社会経済条件等が考慮された社会システムとしての許容度となるであろう。

1990年11月にOECD/NEA主催で開催された「高レベル放射性廃棄物処分における放射線防護及び安全規準に関するワークショップ」並びに1991年10月に開催されたIAEAのINWAC（国際廃棄物管理諮問委員会）サブグループに提出された資料を参考に、性能評価の基本的な事項について示し、地層処分ヴァーチャル・エンジニアリング・システムの最も基本的な機能要件とすることが良いと考えられる。

### （1）性能評価に関わる時間（期間）に関する考え方

性能評価の前提として、人間の生活様式、生態圏の環境、及び地圏の環境の長期予測が必要となる。これらの条件を科学的に予測できる期間をどう考えるかが、性能評価シナリオを開発する上で重要である。何故ならば、高レベル放射性廃棄物中にはPu239、Np237等、10000年以上の長半減期の放射性核種が含まれており長期の問題は避けられないからである。

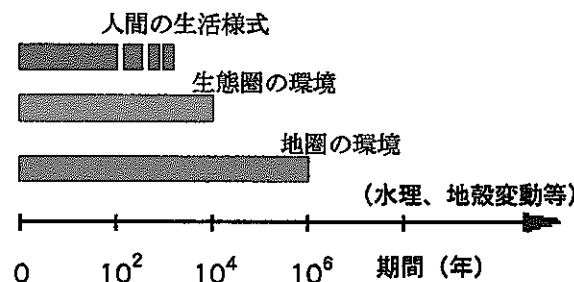


図3. 3-1 処分システムの性能評価に必要な前提条件が予測できると思われる期間

例えば、スウェーデン及びスイスの規制関係者がまとめた報告書（SKI - TR - 90 : 15）にある図3. 3-1に示されている期間についての大略のイメージが一つの参考になる。「生態圏の環境変化を考慮すれば、その予測可能な期間は10,000年程度と推定される。地圏の環境変化を予測できる期間は1,000,000年オーダー程度。サイトを慎重に選べばそれ以上予測できるかも知れない。」としている。

廃棄物中の放射性核種の量は時間が経過するに従って減衰し、大きな影響が生じる可能性も減るので、地圏での異常事象の発生確率の予測上高い精度は要求されなくなる。英国国立放射線防護局では期間の経過に伴って次のような尺度で規制することを提案している。

0-100年	: 制度的管理が残るので施設と同じ被曝線量限度を適用する。
100-10,000年	: 生態圏は変わる可能性があるので、最大被曝を受ける個人ではなく、クリティカルグループを想定し、リスク限度を適用する。
10,000 - 1,000,000年	: 地圏からの放射性核種の定常的漏出を制限することで人類の安全を保つとの考え方から、標準生態圏及び人間社会を想定し、リスク限度を適用する。
1,000,000年以上	: 合理的に可能な限り環境への影響が少ないシステムであることを定性的に示す。

この提案のように、処分場を閉鎖した後の制度的管理の有効な期間は数百年程度であり、定量的な安全評価ができる期間は数万年程度が限界であろうとの点では、各国の規制関係者の見解がほぼ一致している。また、数百万年以上の定量的評価は、生物の歴史や地殻変動等による物質移動を考えると、ほとんど意味がないとの考えが一般的であるように思われる。従って、シミュレーションは時間とともに確率的になって行くであろう。

### (2) 長期の安全性を示す方法に関する考え方

将来の人間の生活習慣や生態圏環境を正確に予測することはできない。従って、現在と同じとして評価することが一般に行われている。このことの意味は、将来の人間の被曝線量を評価するというより、廃棄物から地圏を通過して生物圏に漏出する放射性核種の量を現在の生物圏の環境条件での被曝線量に換算し、将来の放射線的環境が処分によって現在と変わらないことを示すことにあると理解した方が解りやすい。北欧5ヶ国の規制関係者がまとめた提案書では、10,000年程度以上の期間については、地表の侵食によって土壌や岩石中の自然のアルファ核種が生態圏に移行する量と廃棄物に起因する漏出量を比較することを提案している。アルファ核種のみを取り上げているのは、自然の放射性核種と廃棄物中に含まれる放射性核種の種類は必ずしも同じではなく、直接比較できないので、放射線学的及び化学的特性が似かよった自然のアルファ核種と比較することにしたものである。

### (3) 発生頻度の小さい事象の影響評価に関する考え方

慎重に処分サイトの選定を行っても、断層、火山等の急激な地殻変動を完全に否定することはできない。発生頻度が小さくても影響の大きい事象は評価する必要がある。しかし、それは一般的に起こり得る事象と同じ規準で規制する必要はないという考え方を取るべきである。被曝線量を、ガン発生確率でリスクに換算し、被曝の原因となる事象の発生確率を乗じてリスクで規制しようという考え方と、事象の起こり得る程度をランクづけして、それに応じた線量規準を設ける方法がある。先に示したSKI-TR-90:15では以下の通りとしている。

発生頻度	
1から10のマイナス3乗/年	: 通常事象として、被曝線量規準で規制
10のマイナス3乗から10のマイナス7乗/年	: リスク規準で規制
10のマイナス7乗/年以下	: 評価免除

この提案は、次のように解釈することができる。数千年以内でも普通に起きるような事象は、一般的に起きる事象として被曝線量規準で規制する。それより低い発生頻度と思われる事象については発生頻度の少なさに応じ、緩やかな線量規準にしようとの考え方である。線量の規準を1mSv/年とすると、数千年間では、ほとんど発生しないような事象に対しては1Sv/年の被曝線量でもよいことになる。数万年間経過すれば、ガラス固化体の放射能濃度は非常に低くなるので、リスク限度を超えるような事象はほとんど考えられない。地層の安定性についての予測も時間の経過に比例して詳しい予測でなくてよくなる。

高レベル放射性廃棄物処分に係る性能評価は、計画する多重バリアシステムによる放射性核種の長期間に亙る隔離能力を対象に行われなければならない。従って、システムの操作の基本はこの多重バ

リアの性能を数値的に評価し、今後の処分システム開発に参考となる情報を提供することを目的とした作業に適していることが望まれる。

また、前述のリスクに含まれないリスクとして、ボーリング調査等で廃棄物の一部を知らないで取り出してしまう危険性が問題となることがある。(SKI-TR-90:15)では、「地下資源のない場所を処分場にすることで、ボーリング調査等によるリスクを少なくする。」また、「制度的管理に関する全ての資料が失われた後のことを考えた場合でも、将来、ボーリング調査をするような人は、そのことにより、どのような影響があるかを判断する能力を持ち合わせているだろう。従って、現在の人間活動を前提にした評価は疑いなく安全側である。」との考え方をとっている。このような考え方は、将来の社会が文化的にエントロピーが減少し、より高度な文明を築いていることを前提とした考え方である。

### 3. 3. 2 性能評価シナリオに関する考え方

計画するヴァーチャル・エンジニアリング・システムは、開発されたシナリオに沿ってさまざまなモデルを複雑に組み合わせ、協調させてシミュレーションを行い、経過を観察する機能を提供することができるよう開発する。そこで、シミュレーション・シナリオの情報としての扱いが問題となるため、シナリオについて概観する。地層処分は、多重バリアの全体システムの働きによって廃棄物中の放射能が充分減衰するまでバリア中に閉じ込めておくことが前提となっている。廃棄物中の放射性核種が生物圏に漏出する経路は、地下水に溶けて移動するのが主と考えられ、火山の噴火、断層活動等による直接的な露出による被曝の可能性は極めて少ないと想定される。前者は通常起こり得るシナリオとして被曝線量を定量的に評価し、後者の直接的な露出については異常シナリオと考え、発生確率が極めて低いことに着目した評価方法（例えばリスク評価）が導入される。

例えば、通常シナリオである地下水シナリオは次のような考え方(PNCTN141096-071)により示される。地下水シナリオに沿って構築されるモデル体系による地層処分システムの性能評価では、まず地下水流動と化学的特性に関するモデルにより地質環境の条件を与える。次にこれを境界条件として、廃棄物から発生する熱の影響も含め、地下水の浸入に伴う人工バリア内での物理的・化学的変化を扱うモデルにより、処分後の過渡状態における人工バリアの挙動を把握する。これらによって予測される環境の下に、廃棄物、オーバーパック、緩衝材等、人工バリア自体の地下水との相互作用に関するモデルにより、その性状の変化を把握し、その長期的健全性を評価する。さらに物質移動現象を取り扱うモデルにより、ガラス固化体からの放射性核種の溶出及び溶出した放射性核種が人工バリアや地層を経て生物圏に到達する過程を解析する。最後に生物圏モデルにより線量を算出する。それらのモデル体系の中で、地質環境条件の設走に必要なモデルとして、地下水流動モデル、地下水水質モデル、物質移動経路モデル、岩盤力学・熱特性モデルが挙げられる。それらのモデルや最終的な評価対象である生物圏のモデルを組み合わせるシミュレーションを行うシナリオとして、例えば以下のようなシナリオが考えられる。

- 1：地下水が処分場に浸入し、緩衝材中を移行し廃棄物容器と接触する。
- 2：廃棄物容器が腐食する。
- 3：廃棄物容器の密封が破られる。
- 4：廃棄物から放射性核種が浸出する。
- 5：浸出した核種が緩衝材中に拡散する。

- 6：周辺の岩盤へ拡散・移流により移行する。
- 7：岩盤の断層、割れ目または水を通しやすい地層中を移流・拡散により移行する。5－7の過程で核種によっては沈殿・再溶解あるいは吸脱着を繰り返す。
- 8：表層に貫通する断層あるいは水を通しやすい層を通して表層の地下水流、川、湖等に漏出し、生態圏に広がる。
- 9：飲料水、食物等の摂取及び塵埃等の吸収により体内へ取り込む。

評価においては、この基本的なシナリオに影響を与える要因による変動を評価しなければならない。このようにして提案されたシナリオについて評価を行う場合、これらの考えられるシナリオに関連する個別の現象について、仮定とモデルを考え、具体的な評価シナリオとしてシミュレーションによる評価に持ち込むことができる。例えば、上記の条件から多重バリアシステムの性能について評価を行うためには、地下水の存在を前提とし、ガラス固化体から溶出した核種が地下水を介して人間の生活圏に至るという過程について解析することが重要である。このような過程は一種の想定であり、地下水シナリオと呼ばれる。解析を行うにあたっては、地下水シナリオに関連する現象をモデル化することが必要となる。

地層処分システムにおいては、地下に広く存在する地下水がガラス固化体中の放射性核種を溶出させ、これを人間の生活圏に運ぶという過程への対策として多重バリアシステムが考えられている。ガラス固化された高レベル放射性廃棄物は、オーバーバックに封入され地層中に設置される。この際、オーバーバックと地層との間には粘土質の緩衝材が充填される。これらの人工的に設けられる多層の安全防護系が人工バリアである。一方、核種の吸着、遅延といった安全防護機能を本来的にそなえている地層は天然バリアとよばれ、人工バリアと天然バリアのもつ種々の機能を多重に組み合わせたものが多重バリアシステムである。期待される機能は以下のようにまとめられる。それらの理由から、下記の地下水シナリオを想定しシミュレーションを準備することができる。

- 1：透水性の低い緩衝材は、周囲からの地下水の浸透を制限し、その中で地下水の動きを極めて遅くする効果がある。これによって、ガラス固化体と地下水との接触を抑制する。
- 2：オーバーバックは、透水性の低い緩衝材によって地下水の移動が制限されるため、腐食が生じにくく、ガラス質に封じ込まれた放射性核種を容器内に長期間閉じ込めておくことができる。
- 3：オーバーバックが破損し、地下水が廃棄物と接触したとしても、緩衝材中に浸入する地下水の量は限られており、また長半減期の放射性核種の大部分は、もともと深部の地下水に溶解しにくいと考えられていることから、ガラス固化体から溶出する放射性核種の量を制限する。また、地下水に対して比較的溶けやすい一部の核種については固化材であるガラスの保持機能が溶出を制限する。
- 4：緩衝材中ではほとんど水の動きがないため、地下水中に溶け込んだ放射性核種の移動を拡散支配にする。また、緩衝材は放射性核種を吸着しやすい性質を持っているので、それによっても核種の移動の程度を制限する。このため放射性核種の移動には長時間を要し、この間にその放射能を減衰させる。
- 5：緩衝材中から地層に移動した放射性核種は、地層中の地下水の動きが緩慢であるうえ、岩石へ吸着しながら移動するため、その動きは地下水の動きに比べてさらに遅くなる。この移動期間中に放射性核種の放射能を減衰させる。また、地層の複雑な空隙構造が、地下水中放射性核種をいろいろな方向に分散させ、次第に希釈する。

図3. 3-2に主要各国が検討している処分概念と、評価レポートで採用されている評価シナリオの概要を示す。評価結果は通常シナリオに関する限り、線量限度を何桁も下回っている。また、影響が最大となるのは10,000年以後の遠い将来であるとの報告が多い。評価の例を図3. 3-3に示す。

国名 評価報告書	米 国 NUREG/CR11667 (1988) <sup>注)</sup>	カナダ 2nd Post-Closure Assessment Report (1985)	スウェーデン KBS-3 (1983)	スイス Project Gewähr 85	ベルギー SAFIR (1989)
1.処分の対象	使用済燃料、ガラス 固化体	使用済燃料	使用済燃料	使用済燃料、ガラス 固化体	使用済燃料
2.処分概念	ネバダの凝灰岩の地層 (ユッカマウンテン) の不飽和層に処分	カナディアンシールド 花崗岩地帯の地下500 ~1,000mの地層に処 分。チタン製容器に封 入し、坑道の床に掘削 した処分坑道に定置し 周囲に緩衝材(ペント ナイト)を充填する。	結晶質岩(花崗岩)地 帯の地下約500mに処 分。鋼製キャニスター に居れ、坑道の床に掘 削した処分孔に1本ず つ定置し周囲に緩衝材 を充填する。	スイス北部の花崗岩層 地下1,200mに処分。 厚さ25cmの鋼製容器 に封入し、横坑道内に 定置し、坑道内空間に 緩衝材を充填する。	ベルギーの北東部モル の地下約230mの粘土 層(ブームクレイ)に 処分。 高レベル廃棄物を単独 で処分する方法と中レ ベル廃棄物を一緒に処 分する方法を検討中
1.通常シナリオ	地下水移行シナリオ	地下水移行シナリオ	地下水移行シナリオ	地下水移行シナリオ	地下水移行シナリオ
2.変動シナリオ	<ul style="list-style-type: none"> <li>気象変化に伴う水理 条件の変動</li> <li>新しい断層の形成地 震</li> <li>不飽和層における地 球化学的特性の変化 流入水の変化</li> <li>コロイドの形成</li> <li>ボーリング調査</li> </ul> 他	<ul style="list-style-type: none"> <li>基本シナリオのパラ メータの変動</li> <li>容器の早期破壊</li> <li>緩衝材の性能劣化</li> <li>グラウティングの性 能劣化</li> <li>氷河期来襲による気 候変化</li> <li>生活様式の変化</li> </ul> 他	<ul style="list-style-type: none"> <li>スカンジナビアの陸 起</li> <li>初期キャニスター破 損</li> <li>酸化雰囲気の変化</li> <li>コロイドの形成</li> </ul>	<ul style="list-style-type: none"> <li>基本シナリオのパラ メータ変動</li> <li>上部地層への井戸の 掘削</li> <li>気候変動</li> <li>定性的評価: 氷河、 火山、隕石、地震、 岩盤弱部変動、キャ ニスター移動、シー リング破損、その他</li> </ul>	<ul style="list-style-type: none"> <li>気象変化に伴う水理 条件の変動</li> <li>新しい断層の形成、 地震</li> <li>不飽和層における地 球化学的特性の変化 流入水の変化</li> <li>コロイドの形成</li> <li>ボーリング調査</li> </ul> 他

注) Sandia N.L.の研究レポートで評価報告書ではない。

図3. 3-2 主要各国が検討している処分概念

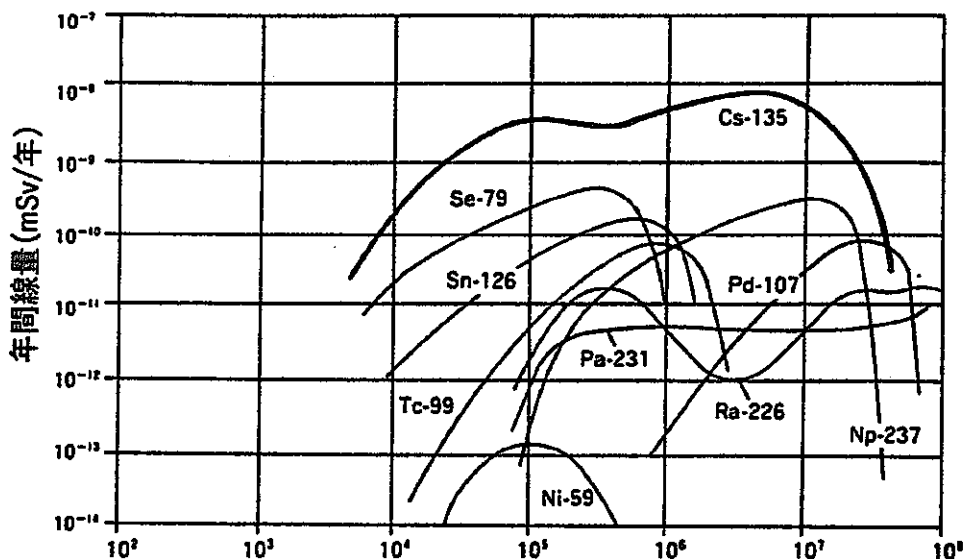


図3. 3-3 スイスに於ける被曝線量評価例

地層処分ヴァーチャル・エンジニアリングにおいては、評価シナリオは、別途開発しているシナリオ・データベースにより準備されたシナリオを利用することを前提としている。但し、シミュレーションにおいては、シナリオの詳細を適宜に参照することが必要である。

シナリオの詳細はFEP (Feature, Event, Process) 手法により記述されることが一般的と考えられる。従って、システムはこのFEP情報を適切に管理し、与えられた評価シナリオについて補足的な情報を提供できなければならない。FEPは、処分システムからの放射性核種の放出や輸送に直接的、及び間接的に影響を及ぼすと考えられる対象についての記述文書の集合である。それは想定されるリスクとして各界の専門家等により示される文章とその関連からなる情報の組である。FEPとみなされたものは、それぞれ原因、及び発生の可能性そして結果についてだけでなく、他のFEPとみなされたものとの最終的相互作用についても連係して分析されなければならない。このような相互作用は順序による影響も考慮する必要があり、場合によっては新たに生じたFEPが既存のFEPに影響を与えることがある。処分システムの健全性の評価に関する複雑な情報を適切に取り扱うためには、性能評価の方法論に関する戦略について深く研究する必要がある。この方法論の重要な構成要素としてシナリオ開発がある。

シナリオ開発の基本目標は、処分システムが超長期にわたり健全性を確保できるかどうかを評価することである。シナリオ開発の最も重要な戦略の一つは、重要な問題を確認しやすくすることである。シナリオをわかりやすく文書ファイル化することによって、関連シナリオを広範囲に再吟味し更新することができる。それをを用いて多方面の専門家により見直しが行なわれることは、公正で合理的なシナリオを開発する上でも、放射性廃棄物の安全な処理という重要な問題について一般的な合意を得る上でも、おそらく最良な方法といえる。FEPによるシナリオ開発経験から、プロセス・システムと呼ばれる新しい考え方が開発された。プロセス・システムが開発された動機は、主にスクリーニング作業が、予想以上に複雑で時間を要しすぎる手法であると判断されたためである。

プロセスシステムは「処分システムやその周囲の環境におけるバリア特性、例えば放射性核種の地下水との化学反応等を説明するのに必要な、あるいは、一定の外的な条件から少なくともある程度決定論的に予測できるような、すべての現象 (FEP) の組織立った集合体」と定義づけされる。

従って、それは全体シナリオといった広範な確率論を中心として扱うような領域と、特定地域の地下水の地球化学モデルといった、分解された過程である個別の領域との中間のサイズの複雑度のモデルを扱う問題に帰着する。また、個別の現象解明については、還元論に基づいて分解された現象過程と現実の実験により検証する決定論的手法がとられ、自由度を下げた方程式から数学モデルが開発される。決定論的手法の弱点はモデル単体の検証は可能であるが、それらを重ね合わせた場合に正しい解が得られるか正確な見通しを得ることが難しい事、また、理想化された実験環境から開発されたモデル式は、現象を普遍的に記述する力は一般にないことが上げられる。

プロセスシステムとは、一定の外的な条件によるシナリオが、FEPに影響を及ぼすような対象によって定義づけされる。現在の知験の範囲内で、より高いレベルに達するための手がかりとなる手法は、プロセスシステムを体系的に形成してコンピュータにより視覚化することである。このようにシナリオに添ったシミュレーションを実現するためには、シナリオの基礎となる情報 (FEPリストや調査デ

ータ、分析データ等)を総合的に管理しシミュレーションを支援する情報を提供する機能を開発することが要求される。以下に、FEPの例を示す。

-----コロイドの形成-----

(1) 内容

腐食生成物は、ベントナイト空隙水化学に応じてコロイドを形成する。これらのコロイドは、核種を吸着したり、取り込んだりしてベントナイトの微細空隙を移行する。コロイドの移行は、それら微細空隙幅や表面電位の様な物理化学特性に依存する。

(2) 評価 (想定期間: オーバーパック破損後)

腐食生成物やそのコロイドは核種を吸着するであろう。コロイドの移行は、ベントナイトのフィルトレーション効果により重要ではないと考えられるが、ベントナイトの実効空隙径はいまだ明らかにされておらず、今後の研究が必要である。

(3) モデル化

(4) 文献リスト

・ McKinley, J.G. (1985): The Geochemistry of the Near-field, Nagra NTB 84-48.

(5) 諸外国での取扱い

[Kristallin-I]

放射性的コロイドは、主としてガラスの溶解や沈澱過程で廃棄物近傍に形成されるであろう。ベントナイトの非常に重要な機能は、コロイドの廃棄物からの移行を妨げることで、それはベントナイトの微細空隙構造によって達成できる。ベントナイトの可塑性と膨潤性は、コロイドが移行できるような空隙や割れ目がある中に存在し得ないことを保証するものであることが期待されている。

3. 4 性能評価シミュレーション問題

シミュレーションに於ては、基本的な考え方としてその評価期間を短期、長期、超長期に分けて扱うべきであると考えられる。

3. 4. 1 シミュレーション期間

(1) 短期 (約1000年まで。処分後の地下水浸潤等比較的過渡的な期間)

廃棄物中の全放射エネルギーの時間変化を見ると、初期の1000年で約1/1000になることが知られている。それ以降は、清練に使ったウラン鉱石全体中に存在する量と大差ないと考えられる。廃棄物中の全放射エネルギーは放射能濃度で見るとウラン鉱石の約10倍程度になると予想されている。従って、設計対応や定性的評価手法が該当する事象が起きる可能性のあるのは1000年までであり、この期間は『影響大』の事象が発生しないように設計することが望ましい。従って、処分システムは、この間は複雑な地球化学的過程を中心とした現象に関心が集まることを重視して設計されるべきであると考えられる。それは、シミュレーションが決定論的モデルを複数連成させた複雑な問題となることを意味している。ヴァーチャル・エンジニアリングにおけるシミュレーション・アーキテクチャ開発の当面の目標は、この期間では複雑な現象問題をシミュレートする技術を開発することである。

また、シミュレーション結果の検証等にナチュラル・アナログ研究の一環として、歴史的資料や考古学的建造物等の情報をデータベース化し、工学的手法参考になる情報を提供することによりシミュ



レーション結果の検証・確証を支援できる。現代の経験で確かなことが言えるのは100年程度であり、科学的予測の精度を上げるためには更に基礎研究が必要であると思われる。そのためには、この期間の現象解明に係わるより包括的な評価を支援する複雑系シミュレーションと高度な対話機能が必要であると考えられる。

### (2) 長期 (約1000 - 10,000年)

放射能濃度は1万年経過するとウラン鉱石と同じオーダーになると考えられている。従って、長期の評価に於ては影響は少ないと考えられるが、発生の確率が高いと思われる事象シナリオについて評価の関心が高まると思われるため、対話システムの機能は、より多くの気候変動や地質等のデータベースについて充実したものが求められると考えられる。

例えば、氷河期の周期は数万年オーダーであり、約1万年程度であれば、自然条件、生物活動等が現在と同じと考えても不自然ではないと思われる。生物圏に漏出する放射エネルギーによる影響を現在の生活条件を根拠に、定量的に被曝線量として評価しても不当ではないであろう。また、第四期(約200万年)の地殻変動データを使うことにより、地日中における約1万年以内の事象は比較的定量的に評価できる。

### (3) 超長期 (1年以上)

1万年を越える超長期の評価においては、生物活動について人類の存続の疑問も含めて、不確定要素が多くなり、被曝線量を評価する意義が疑わしくなるかも知れない。ウラン鉱床中の放射能濃度と処分システム中の平均放射能濃度は大差なくなると思われる。超長期のシミュレーションにおいては、そのような理由から地質学的年代に互るナチュラル・アナログや古気象学、古生物学等の情報も参考となるため、それらを適宜に管理し提供できる機能が必要となる。ナチュラル・アナログ研究においては、地層処分された廃棄物から放射性核種が生物圏に漏出するのを防止、あるいは抑制するバリアシステムの機能時間と放射能毒性レベルとの関係が、図3. 4-1のように簡単に示されている。

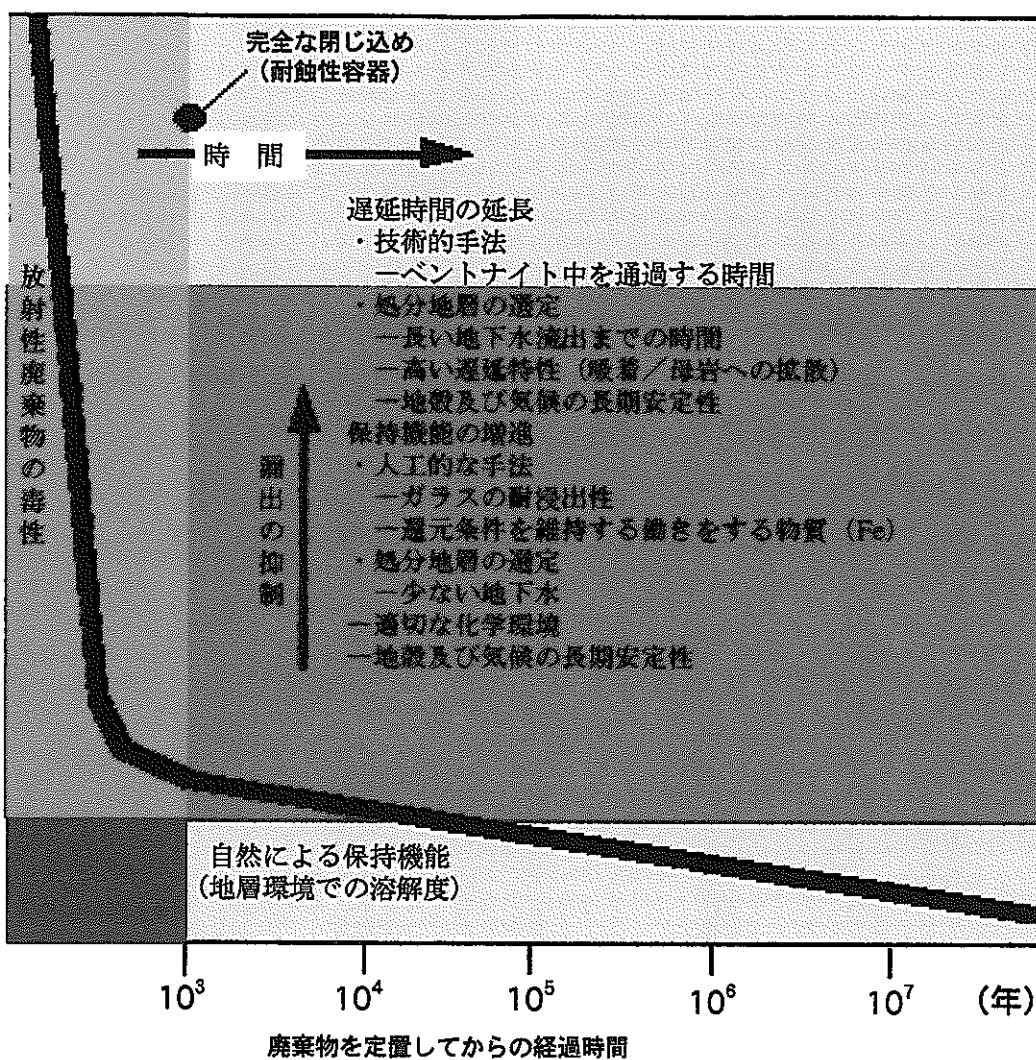


図3. 4-1 バリアシステムの機能時間と放射能毒性レベルとの関係

また、放射性核種がニアフィールドからファーフィールドを経由して生物圏に至る過程を図3. 4-2のよう示す。



図 3. 4 - 2 放射性核種が生物圏に至る過程

容器の閉じ込め性能が確保されている間に、短半減期の放射性核種が減衰し、放射性廃棄物の全体の放射能は何桁も下がる。孔が開いても残った金属容器の残骸は周囲の地下水を通元性に保ち、金属酸化物は放射性核種を吸着する。近年、この他に図 3. 4 - 3 に示す処分場で発生するガスの影響が話題となっている。

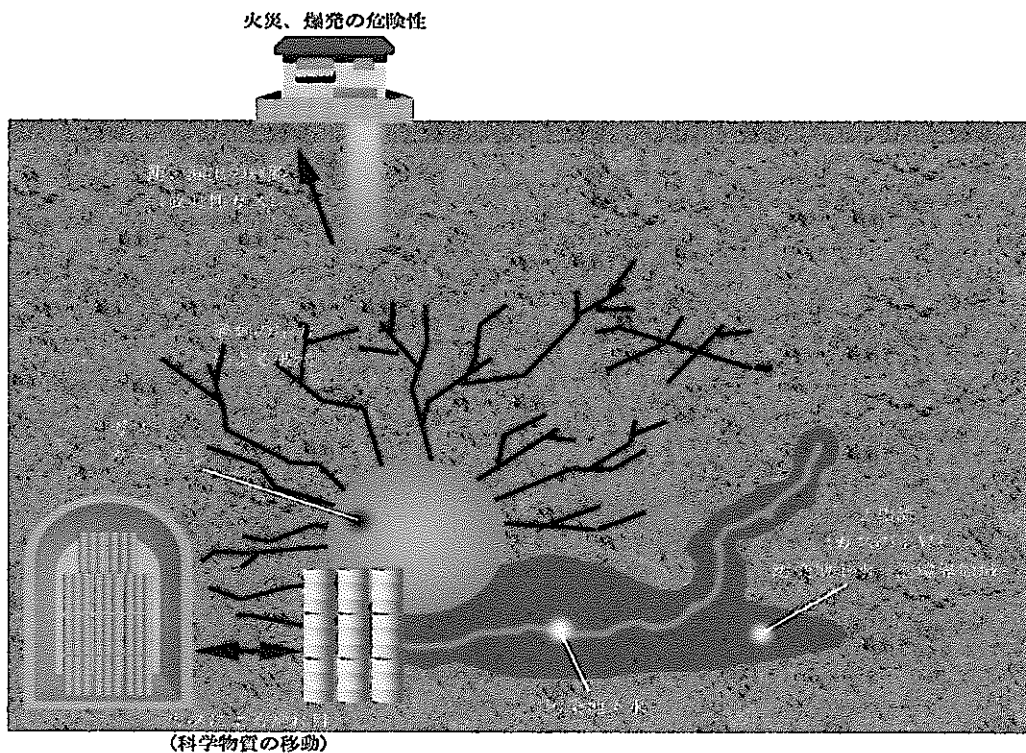


図 3. 4 - 3 ガスの影響

廃棄物が腐食して発生するガスは、地下水の動きを変える。 $\alpha$ 放射線による水の放射線分解による局所的な酸化性環境下で廃棄物は溶解する。岩、ベントナイト、廃棄物及び容器の劣化生成物への吸着や酸化還元フロントでの沈殿で放射性核種の移行が遅れる。

### 3. 4. 2 ナチュラル・アナログ

ナチュラル・アナログ研究は、このような地層処分の概念と長期の核種移行過程を念頭に進める。ガスの影響や微生物の影響については、石油工学等での研究成果などを参考にするとよい。また、廃棄物に含まれる放射性核種のなかには天然に存在しない元素がある。ナチュラル・アナログ研究では化学的性質の類似する元素の挙動で近似する。しかし、類似していた元素同志も原子価が変われば性質が著しく変わってしまう。どのような元素の挙動で近似するかは、ケースバイケースで考える必要がある。

例えば、Npについては、Np (V) はPa、Np (IV) はTh、酸化還元の変化を伴う現象の研究ではUで近似するのがよいと考えられている。TcはRe、AmはEu、Pu (IV) はTh、Pu (VI) はUである程度近似される。ナチュラルアナログ研究は、研究対象により様々な自然環境が使われる。最も多く使われているのはウラン鉱床である。使用済燃料の腐食溶解現象、酸化還元フロントや割れ目での放射性核種の挙動、コロイド形成等の核種移行助長現象、吸着現象や岩盤への拡散現象、半減期の違う核種を使った時間依存現象等多くの項目の研究が行われている。緩衝材中での核種移行現象やベントナ

イトの変質など緩衝材の性質の研究には粘土が専ら使われている。

このように長期の評価について当面は、天然に存在する放射性鉱物の放射線影響との比較において、定性的な検討にとどめるのが適当と思われる。ナチュラル・アナログ研究情報は、シミュレーションを試行錯誤的に進める上で重要な参考情報を与えることができる。ナチュラル・アナログ情報を整備しデータベース化することにより、シミュレーションを進める上で参考となる各種の重要な情報を随時参照することができ、より効率的な性能評価シミュレーションを行うことができると考えられる。地層処分ヴァーチャル・エンジニアリングにおけるシミュレーション機能には、凡そ以下の分類によるナチュラル・アナログ情報サポート機能が提供されることべきである。

### (1) ガラス

ガラス質の安定性については、失透して結晶化するため長く安定には存在しないといわれているが、アポロが月で採取した天然のガラスは $3.7 \times 10^9$ 乗年前にできたものであり、北アメリカで採取した天然ガラスの半数以上は $2 \times 10^6$ 乗年以上、 $4 \times 10^7$ 乗年前のものもあつたとの調査結果がある。そのほか、 $1.1 \times 10^9$ 乗年前のものも発見されており、条件によってはガラス質も非常に長い間安定に存在することがわかっている。水による溶解速度と変質については、生成物天然ガラスと $0^\circ\text{C}$ の氷解水、海底火山の天然ガラスと約 $4^\circ\text{C}$ の海水、火山灰中のガラス質と常温の雨水、玄武岩質ガラスと温泉水などいろいろな種類の天然ガラスを使い、温度もいろいろと異なる水と長い間接触した場合の溶解量や表面の変質の様子が調べられている。海水による浸出についての調査の例では、溶解速度は $3 \sim 20$ マイクロm/1000年程度であり、海底堆積物で埋まっている場合は約0.1マイクロm/1000年との値が報告されている。変質生成物は、状造を形成しており、ガラス質から可溶性成分が抜けた薄い層、非結晶性の二次生成物層、及びその上に粘土鉱物などの結晶性の鉱物の層の凡そ3層が存在する。この厚さは $0 \sim 1000$ マイクロmである。これが代表的な変質層であり、廃棄物を固めているホウケイ酸ガラスの実験室試験で認められる変質層の構成とほとんど同じである。

今後のナチュラル・アナログ研究は、二次鉱物への廃棄物成分の取り込みについての研究に重点が移されるべきであるとされている。考古学上のガラス製遺物は、紀元前BC500年頃のエジプトのガラス製品の保存状態が調べられている。人工のガラスでも水に接触している場合の変質層の様子はよく似ているが、湿気のある空気に接触していた人工のガラスでは、表面に天然のガラスでは認められない微細なひび割れがしばしば認められる。

### (2) 鉄、及び鋼

天然の鉄金属が見出だされることは稀であるが、隕石、グリーンランドのDisko島及びドイツのBUhlの天然の鉄についてのナチュラル・アナログ研究の例がある。考古学上の遺物としては、スコットランドのInchtuhil遺跡で約1,900年間、約5mの深さに埋まっていた約百万本の釘があげられる。釘の長さは $6 \sim 40$ cmである。鉄の腐食は埋まっている場所が酸性化環境か還元性環境かで大きく異なる。10の6乗年間以上残っているBUhlの天然の鉄は、透水率の低い玄武岩に埋まっている。鉄から玄武岩の中への酸化還元フロントの移動は106年間で $1 \sim 2$ cmである。Johnsonらは、これまでの天然の鉄や考古学上の鉄製遺物の還元性環境での腐食の調査結果を整理して、 $0.1 \sim 10$ マイクロm/yという値を出している。

### (3) ベントナイト、セメント、コンクリート

ベントナイトは、イライト化して膨潤性等の性質が変わることが心配されているが、100°Cに保たれたとしてもイライト化するには10の7乗年以上かかることがわかっている。温度が上がるとベントナイトが固まり割れが生じやすくなることもあるが、100度C以下では重要な変化は生じない。コンクリート及びセメントについては、1,700年前のローマのコンクリートの調査等により、コンクリートの主要成分であるケイ酸カルシウム化合物は長期間安定に存在するとの結果が出ている。古いコンクリートの間隙水の研究は、ニアフィールドの放射性核種の濃度評価にとって大切であり今後の課題である。

### (4) 溶解度と化学種

放射性廃棄物の処分は、地下水の動きができるだけ少ない場所が選ばれるのが原則であり、ニアフィールド、ファーフィールド共に平衡状態での核種濃度の評価が重要であり、地球化学モデルが多用されている。その検証のためにナチュラル・アナログ研究が行われている。地球化学モデルは、一般に地層鉱物と地下水の特性から考えられる化学平衡式を抽出し、系統的に解く方式がとられる。この方式で求めた濃度とフィールド調査での値とはかなり違う場合が多い。そこで、高pH鉱泉のMaqarinサイトでの研究などでは、従来の方法に加えてフィールドに存在する微量鉱物まで詳しく調査し、見つかった微量鉱物のうち、どれが水中濃度を決めているかを探するという方法が取られている。地球化学モデルでの評価値は、一般に計算値より実測値の方が濃度が薄いとの結果になるが、Maqarinサイトの研究例では、CaUO<sub>4</sub>の値を使ったためウランについて逆に実測値の方が濃い結果になった。フィールドでは、CaUO<sub>4</sub>の純粋のものは発見されてなく非結晶性の物質が発見され、この物質が水中のウラン濃度を決めているであろうとされている。一般に化学平衡が成り立っているとして計算されるが、自然界での結晶化現象など非常に遅い反応があり、その速度がわかっていない場合が多い。また、固溶体や共沈現象は地球化学モデルには組み込み難い宿命を持っている。これらの点が実測値の方が低くなる主な原因である。

### (5) 移行遅延現象

結晶岩の割れ目については、割れ目の内側を覆っている鉄の水酸化物、有機物及び粘土鉱物への吸着、及びそれらの物質との沈殿作用が主であることが多くのナチュラル・アナログ研究で明らかにされている。すなわち、安全性評価では吸着は可逆的であるとして評価されるが、不可逆の沈殿成分が多い。また、鉄の水酸化物は特に重要である。吸着による遅延作用のみでなく、吸着したコロイド粒子となって放射性核種を逆に動きやすくする場合も観測されている。堆積層中での物質の拡散についてのナチュラル・アナログ研究で興味あるものとして、スコットランドのLoch Lomondでの研究がある。

淡水の湖の底に厚さ2m程度の薄い海洋堆積物がある。この堆積物は6900—5400年前のものであることがわかっている。この堆積層から淡水の堆積物への臭素の拡散分布が調査されている。その調査結果では拡散係数は $8 \times 10^{-11}$  m<sup>2</sup>/sであった。堆積層での微量元素の移行調査で良く知られているものとしては、Okloの天然原子炉に隣接する粘土層への核分裂生成物の移行状況の調査があげられる。生態圏と地圏との境界領域の移行調査については、原爆実験による放射性核種の地中分布の調査があげられる。それによると100m深さの井戸でも9ヶ月後にはRu106が検出されており、5年間の連続

モニタリングで岩石の割れ目を約300m移行したことが確認されたとの報告もある。

これらの結果は生態圏での放射性物質の移行評価モデルを検証するにはよいデータである。

#### (6) 岩整中の拡散

ウランの崩壊系列核種を測定した例では、10の5乗年程度の間には酸化性の地下水が流れた場所の岩盤でウランが減少している所の割れ目からの距離は30-40mmであったとの報告がある。また、海水に浸潤されていた岩盤が上昇して淡水に浸潤された岩盤を使った調査がある。

割れ目に沿った間隙水の塩分濃度の分布を測定したところ、拡散係数は $2 \times 10^{-9}$  m<sup>2</sup>/sであったとの報告がある。岩盤中の拡散は、酸化還元フロント移動との組み合わせの現象として研究されている例もある。岩盤の中に還元スポットが生ずる現象が見つまっているが、花崗岩での還元スポットでの調査では、還元性領域は約10の6乗年の間に半径約10mmしか広がらないとの調査結果が出ている。以上のような調査から、一被に放射性廃棄物の処分の評価に関係する岩整の拡散領域は100-200 mm程度を考えればよいとされている。

#### (7) 地下水の放射線分解

Okloの天然原子炉現象があった場所から、Moが80%、Tcが35%、Ruが25%失われている。そして、まわりの酸化還元フロントで止まっている。これらの元素は、酸化状態の高い化学形でなければ移動しないので、水の放射線分解で生じた酸素、または酸化性物質で一旦は酸化されたであろうと考えられている。そして、放射線分解で生じた水素が粘土層に拡散して形成した還元領域で還元され沈殿したと考えられている。Cigar Lakeでの研究でも、ウラン鉱石(UO<sub>2</sub>)の表面に少量ではあるが、酸化状態の高いウラン酸化物が存在することが確かめられている。

#### (8) 酸化還元フロント

結晶質岩については、PoCos de Caldasのウラン鉱床の岩盤の風化側の茶色の領域と青灰色との境界を使って詳しく研究されている。境界は非常にシャープである。ウランは還元側境界で濃度が増している。Mo、Ce等酸化還元敏感な元素はウランと同じ挙動をとる。その他の酸化還元条件で化学形が変化しない微量元素も、鉄の水酸化物との共沈、固溶現象、あるいは吸着現象で多かれ少なかれ酸化還元フロントに濃縮している。そして、10の8乗年間以上移動しないで固定されている。

#### (9) コロイド

例えば、Po - gos de Caldasでは浅いところと深いところで調べ、浅いところでは、コロイドは主に粘土で0.8mg/l程度存在しており、深いところでは、主に非結晶性の鉄の水酸化物で0.5mg/l程度存在している。また、コロイド中のTh230/U234比から、ウランはコロイド中からは少なくなっていることがわかっている。高pHの鉱泉であるMaqarinでの調査でもコロイドの存在量は1mg/l程度と少なかった。

### 3. 5 地下水シナリオの例

多重バリアシステムを採用した場合は、どの部分を各バリアの境界と考えるかが重要となる。例えば、米国NRCの基準では廃棄物パッケージと地層の2つのバリアに分けている。設計ベースで考えると、オーバーパックと緩衝材を2つのバリアと考え、更に地層バリアが加わり3重のバリアシステムと考えることができる。この場合、容器は密封機能、緩衝材は移行遅延機能、地層は隔離機能（接近できなく移行距離が長い）と、それぞれ異なった機能で核種移行を阻止する働きをする。従って、この異なった機能を期待する評価シミュレーションにおいては、これらのバリア構成要素を明確に3次元オブジェクト等で視覚化して区別することにより、整理されたシステムを構築する必要があると考えられる。また、長期の評価においては、高放射線被曝の事象の発生はほとんど考えられないので、極めて低い発生確率であることを定量的に評価することは必ずしも要求されないと思われる。単独で閉じ込め性の高いバリアを重ねる必要はない。しかし、我が国の地層は必ずしも安定ではないので、設計目標としては、地層の不安定性に起因する不確定要素を排除することを目的とした多重バリアを採用した方が、評価が容易となることが多重バリア方式の採用論拠である。この多重バリア方式の処分システムの機能を周辺母岩と合わせて、機能別に分類評価できる対話画面構成が重要である。

地層データベースについても詳細な調査結果から構成でき、確実な性能評価が可能な処分システム周辺の岩盤と距離による隔離機能を期待する、広い範囲の地層の多重バリア・システム設計の支援機能を考えることができる。処分システムに及ぼす影響事象をサイト特定、及び設計により、かなりを除くことができると考えられている。更に、除くことのできない事象のうち、放射能の漏洩に結びつくシナリオを抽出する方法として、それぞれの事象に定性的発生確率を与え、相対的発生確率から重要事象を求める方法がとられる場合が考えられる。また、定性的な数値は科学的根拠に基づいて与えることができるとは限らない。それらの数値の見積りに当たって根拠となる科学的データが得られない場合は、まず専門家の見識に頼ることになる。また、同時に公衆の感覚的判断も大切になる。従って、いわゆるパブリック・クライテリアの必要性が生じる。合理的なパブリック・クライテリアを形成するためには、まず、関連する科学的事実を公衆に伝達することが大切である。そして、長い時間をかけ多くの人々が理解できるよう整理する必要がある。

以上のことから、シナリオに添ったシミュレーションを操作する機能には、確率論的评价面から重要事象が示されることや、将来のパブリック・クライテリアとの関係、社会システムへの影響といった包括的な機能が設計当初より含まれていることが望ましいと考えられる。

#### 3. 5. 1 地質環境と地下水シナリオ

当面の評価シナリオとしては、処分システムの建設時における掘削影響等からの地下水環境変動シナリオ、及びそれぞれが原因となった核種の放出シナリオを開発する必要がある。地層処分ヴァーチャル・エンジニアリング・システムの開発においては、シナリオは別途推進されている地層処分シナリオ・データベースの開発プロジェクトにより得られたものを応用する。従って、シナリオに添ってシミュレーションを進めるための機能設計上要求される事項について基本的事項をとりまとめる。

ここでは、現在提案されている（PNCTN 1410 92-081）地下水シナリオに添って、設計開発するシミュレーション・システムの基礎的な要求機能を検討する。

地下水シナリオの記述にあたっては、地下水の流動や化学的性質に影響を与える地層処分システム



の場としての特徴やそこで生ずると考えられる種々の事象、プロセスを検討する必要がある。そのため、地層処分システムを人工バリアが設置される環境である、地質環境と人工バリアの設置によって影響を受けるニアフィールドという2つの空間領域に大別し、場としての特徴を記述し、次いでそれぞれの領域において生ずると考えられる事象、プロセスを考慮して地下水シナリオを作成することとした。

地質環境は、地下水の流動状況と化学的性質、さらに核種移行等に関連する化学反応や物質移動を規定する場として捉えられる必要がある。地下水の流動状況については、広域流動系と局所的流動系とによって、また地下水の化学的性質については、地下水供給域と深部岩体における地球化学的特徴によって考えることができる。地球化学反応や物質移動に対しては、鉱物組成及び微視的空隙構造が重要である。地質環境は、シミュレーションを行う場合の仮想環境を構成する情報と考える。我が国の地質特性等から、ヴァーチャル・エンジニアリング・シミュレーションを行うための仮想地質環境を構築するには、海岸平野、内陸盆地、山岳地域、火山地域の4つに分けて構築することが適当であると考えられる。また、別の考え方として、地殻の力学的安定性について地質データ等を基に分類し、モデルとなる仮想地質を構築することも有効かと思われる。いずれにしても地層処分ヴァーチャル・エンジニアリングにおいては、仮想地質環境をデータ等から開発することが基礎的な要件であると思われる。

#### (1) 仮想環境の構成

仮想の地質環境を構築するには次の方法が考えられる。第一の方法として、実際の調査データ（衛星リモートセンシング・データ）から地表面に関する3次元の情報を取りだし、それをもとにポリゴナル・データ（ポリゴン）を作成する方法である。第二の方法は、特定した地域の情報が利用できない場合であり、この場合は、前述の調査データやフラクタル・シミュレーション等を利用して、理想的な仮想地形モデルを作成しポリゴン情報とする。地質情報は、実際には地表データから得られるポリゴンから作成する一枚の形状オブジェクトが基本となる。従って、前述の地表形状オブジェクトは、構築する仮想地層モデルの最上位のクラスに位置する情報と考えるべきである。また、側面情報は、そのサブクラスとなるべきである。

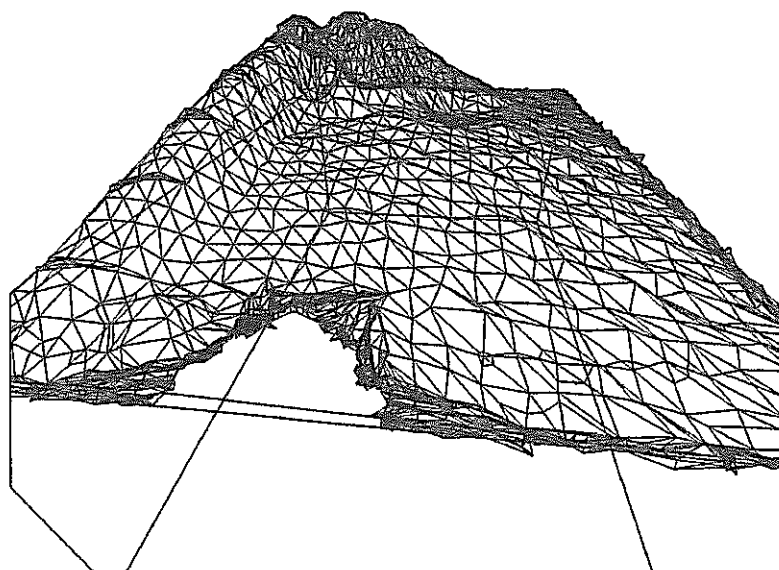


図 3. 5 - 1 地表形状オブジェクトの例

### (2) エージェントの扱い

仮想の地層環境を構成するエージェントは環境要素すべてである。従って、それらエージェントを詳細に調べることは現在は難しい。ここでは、将来において最低必要と考えられるエージェントについて示す。第一に地上処分施設が上げられる。



図 3. 5 - 2 廃棄体受け入れシーン

恐らく、地上施設は処分システムと生活圏を接続する最も可能性の高い接点であろう。地上施設のエージェント要素は処分設備である。これらは、作業時には搬入された廃棄体を地下の処分ピットまで搬送し埋設する。そのモデルは、丁度、交通のシミュレーションと類似しており、地震、電源喪失等で大きな影響を受けるシステムである。また、地下水や想定できる岩盤等、地層環境を構成するおおよかな環境要素を接続することにより、よりリアルなシミュレーションが可能となるであろう。

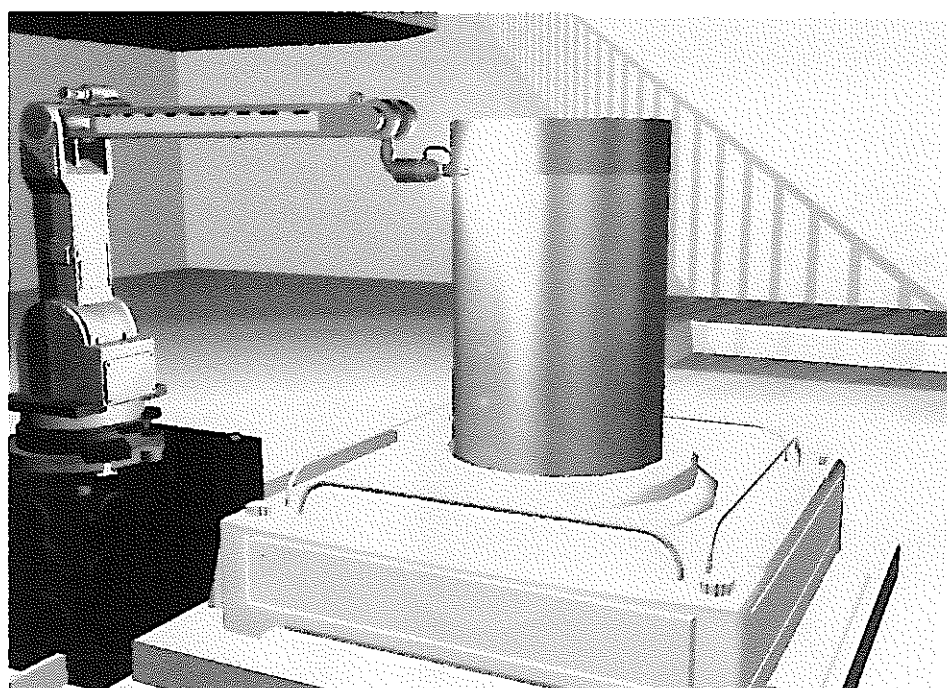


図3. 5-3 オーバーバックの溶接シーン

図3. 5-2、図3. 5-3は、地上施設のエージェントとして考えられる要素の試作データから、動作シミュレーションの1シーンを取り出したものである。

### 3. 5. 2 広域地下水の流動

広域の地下水流動系は、地下水供給域から地下水排出域までの循環系全体の地下水流動状況を記述するものとして重要である。広域の地下水流動系をモデル化するためには、日本で観察される地形、地質に基づき、系の境界条件と系内での流動経路を規定する水理地質構造を設定することが必要である。そこで、系の上部境界条件は、生物圏の下層部にあたる土壌下面のポテンシャルによって規定し、下部境界としては、一般的に透水性が深部に行くに従い減少することから、系全体の流動状態に対して流束が相対的に無視しうるような深度を考えるものとする。他方、側方境界は、地形、地質的要因によって、地下水系が分断される場合と他種の水系である海との境界として設定される場合とが考えられる。また、割れ目の発達した系では、断層が側方境界の一つとして考慮される必要がある。

従って、鉛直断面に着目すると、側方境界のバリエーションとしては、以下の組み合わせが考えられる。処分システムの候補となる仮想世界の地形をコンピュータ内部に構築する場合、側面の境界は

およそ次の6通りとなるであろう。それらは、断層—断層、断層—海、断層—山、山—山、海—海、山—海である。ここで、山—山を側方境界とするパターンの中には、峡谷と盆地とが含まれる。これらは、三次元的な流束の集散状況が異なることから評価対象としては区分して考える。結晶質岩、及び割れ目の発達した堆積岩においては、地下水流動系が断層により分断されていることが一つの特徴となるのに対して、新第三紀系などの割れ目の少ない堆積岩系は、平野あるいは堆積盆として形成されている。ここでは、透水性の異なる層（例えば、砂岩と泥岩等）が発達し、巨視的流動状況は、この層序に支配されることが特徴である。

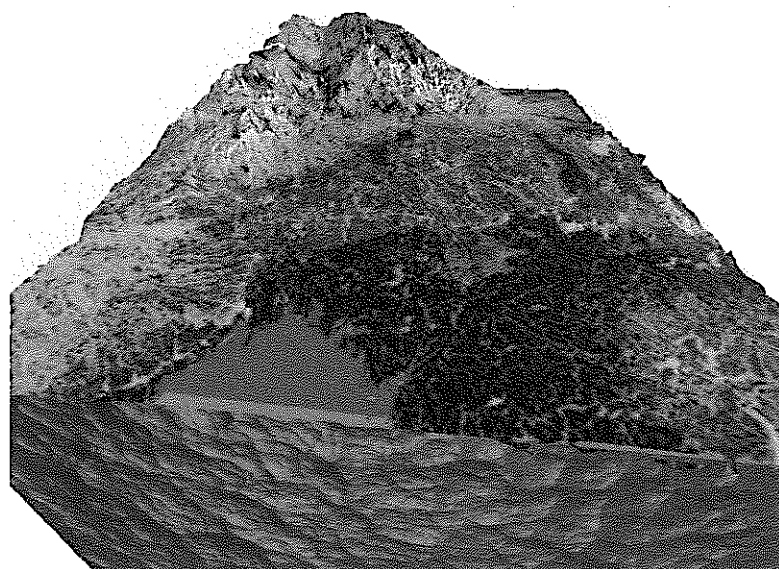


図 3. 5 - 4 側方境界を含む仮想地層モデル

また、難透水性層の下位に高透水性層が位置する様な場合、被圧帯水層が形成されることとなる。さらに、これら2種類の系が共存するパターンとして、割れ目の少ない堆積岩に覆われた結晶質岩等がある。この系では、上記の特徴に加えて、新第三紀堆積岩と結晶質岩との境界において、新第三紀堆積岩層での流動状況に対応したポテンシャル分布が生じ、これが下位の結晶質岩中の地下水流動に関する上部境界条件を規定することとなる。

#### (1) シミュレーションのための仮想環境の構成

広域地下水の仮想環境は、前述の仮想地層クラスのサブクラスとして接続すべきである。仮想環境構成要素は、岩盤と亀裂等広域的な浸透流の流路に影響する物理形状である。また、前述の砂岩と泥岩等、透水性の異なる層ごとにクラスを編成することも有効かと思われる。

クラス編成上問題となると考えられる対象に亀裂がある。亀裂は、それ自身では空隙であるため体積をもつが、従来の亀裂水理コードなどでは、面やジャンクションとして扱っている物が殆どである。また、亀裂は非常に小さなものまで多数分布しているため、モデルを考えた場合、ある程度の亀裂より小さなものは確率的に分布しているとして記述することが望ましいとも思われる。

## (2) エージェントの扱い

地下水を与えられた環境で独立して動くエージェントとして考えた場合、次のアルゴリズムが考えられる。第一に従来の手法の発展系である。それは、従来とられてきた離散近似的手法（有限要素法や有限差分法）を基礎としたモデルの組み合わせである。例えば、広域亀裂水理を計算するコードでは、与えられた条件に基づく地下水の流れ場を時間とともに得ることができるため、例えば核種移行コードを組み合わせる利用することが考えられる。この方法は、シミュレーションを組み合わせる方法であるため、コード間のメッセージ交換をプログラムすることは可能であるが、組み合わせられたコード間の数理モデルの整合性を完全に調整することは難しいと思われる。

一方、マルチ・エージェントによる協調計算を想定した広域地層の仮想環境を構成することを考えた場合、地下水流れは地下水の微小体積要素をそのままオブジェクトとしてモデル化した粒子群になるであろう。その方法によれば、個々のオブジェクトは水分子としての属性を与えることも期待できる。問題となるのは、計算量が桁違いに異なることであるが、近い将来開発されるであろうテラフロップ級（1テラは1秒間に10の12乗回の浮動小数点演算を行う計算速度）のコンピュータを利用すれば、かなりの精度の計算が期待できる。図3. 5-5は、20,000個のオブジェクト粒子を微小な流速で動かした例である。

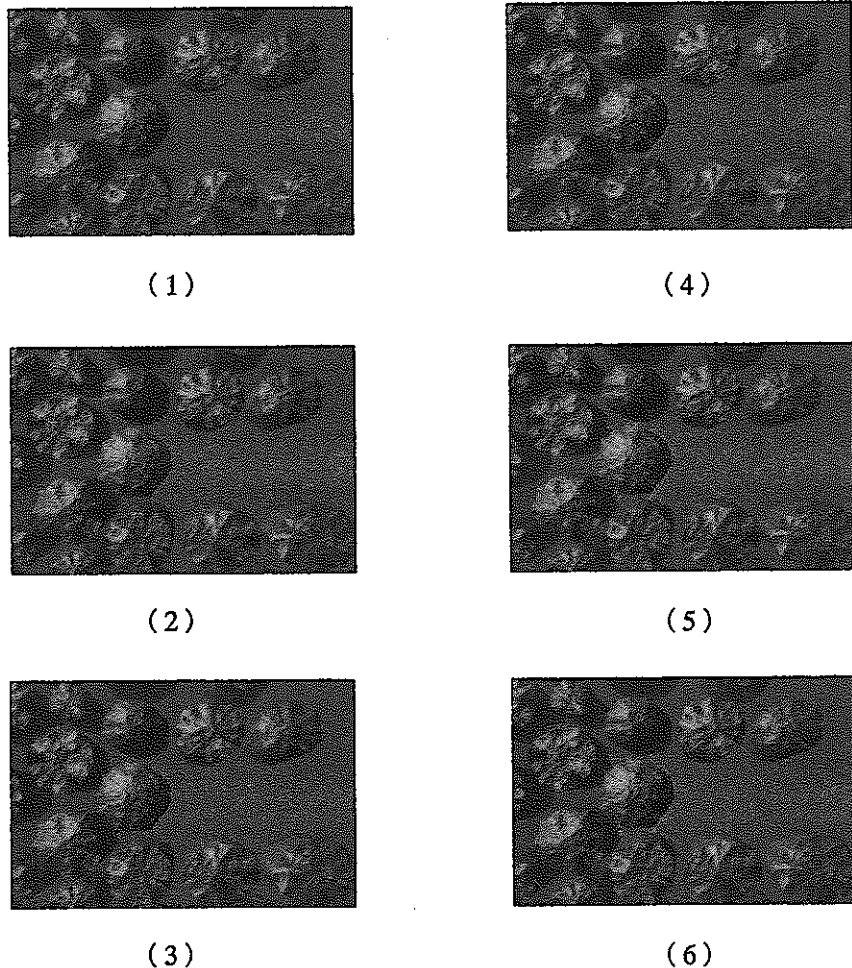


図3. 5-5 簡単な吸着オブジェクト・シミュレーションの視覚化概念

### 3. 5. 3 局所的地下水の流動

結晶質岩や固結の進んだ堆積岩は一般に比較的小さい空隙率を示すが、岩体の生成時より存在している一次的空隙に加えて、その後の応力履歴等を反映した二次的な空隙、即ち割れ目、節理等を有することが特徴である。二次空隙は、岩体中の粒界等の不均質境界に応力が加わってクラックを生ずることによって形成される。形成された二次空隙は、地下水の卓越した流路となり、選択的に風化作用を受けることとなる。可塑性を有する堆積岩では、岩石形成時以降に加えらるる応力に従って変形することにより、一般に大きい開口割れ目等は生じにくいと考えられる。

これらの岩石は、堆積時の粒子間空隙に対応した一次的空隙を有しており、空隙は堆積後の続成作用において、圧密による空隙の縮小と二次鉱物の沈殿過程を経る。従って、微視的構造は、碎屑粒子間を二次鉱物群がとりまき、かつ、一次空隙が一部残されたものとなっている。このような堆積岩においては、一般に顕著な層序や異方性が認められる。特に、泥岩中に砂岩層が存在する場合には、それが相対的に高透水性の地下水流路となる。

#### (1) シミュレーションのための仮想環境の構成

局所的地下水を取り巻く環境は、花崗岩や玄武岩のような結晶岩質である。それは処分システム全体を取り巻く環境である。従って、このような領域について仮想環境を構成するには、結晶岩質の岩盤、処分システム全体が主な要素となるであろう。評価にあたっては小さな空隙による、水理、岩盤力学、温度変化等に関心が集まると思われる。

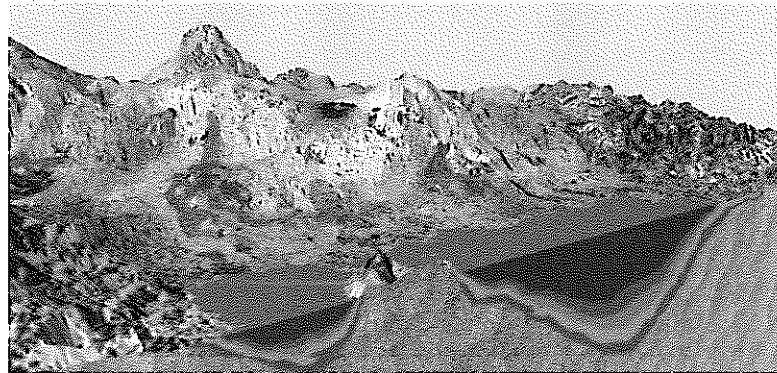


図 3. 5-6 岩盤と地層のモデル例

仮想岩盤のモデルは、ボーリング等の調査データや前述の生成データを利用して地層をつくることの良いと考えられる。図 3. 5-6 は、試作したオブジェクトから視覚化したものである。

#### (2) エージェントの扱い

花崗岩や玄武岩のような固い岩整中のほとんどの流れは、不連続な破碎帯（割れ目）を通して流れる。破碎体の流れのモデルは、従来より次のようなアプローチがなされている。例えば、十分大きなスケールにおいて、一連の小さな割れ目は等価な多孔質媒体としてモデル化できる。しかし、処分場近傍あるいは断層のような大規模な破碎帯に関しては、特に破碎帯専用のモデルを用いるべきである。

破碎帯内の流れをモデル化するアプローチには、不連続破碎帯（ディスクリートフラクチャ）法と二重有孔（デュアルポロシティ）法がある。二重有孔法は、岩層モデルと破碎帯モデルを重ね合わせたモデルである。それに伴う移動関数は、岩層と破碎帯間の流体の移動の関係を表している。従来の不連続破碎帯モデルは、一般的に割れ目中の流れがほぼ平行板間の流れで近似できることを仮定する。また、岩層中の流れは多孔質媒体中の流れとして解析する。破碎体をオブジェクトとしてモデル化することは、概念としては可能と考えられる。図3. 5-7に示すような調査データ等をもとに、破碎体をランダムな形状の体積をもった、ポリゴンの組み合わせで生じた隙とする手法として考えることができる。

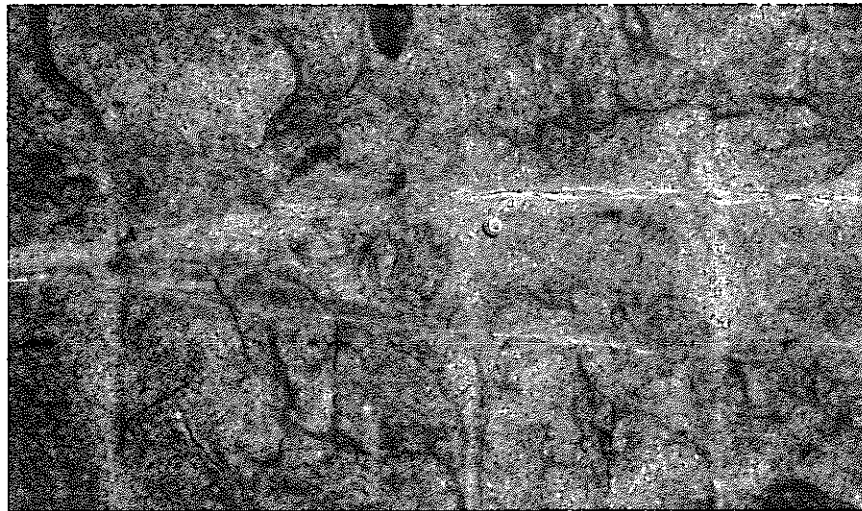


図3. 5-7 亀裂の例

問題は、ポリゴンを生成するにあたり、必要なランダムな局面生成のアルゴリズムを決定することと考えられる。いずれにしろ、この方法はマイクロからマクロまでの一環した破碎帯モデルの記述を可能にすることが期待できる。そのような手法で生成された3次元空間中に詰め込まれた不規則な形状のポリゴンが接する隙は、前述の微小粒子モデルで示した水粒子の流路を形成する。そのとき、水粒子と岩質との収着や沈殿等の現象はクラス階層として組み込むことが期待できる。そのような手法は、地下水の流れと化学的性質を個のモデルとして組み込むだけで全体の流れ実験を数値的に行うことが可能であろう。但し、計算速度は現在の超高速コンピュータとは桁違いのものが要求されるであろう。

### 3. 5. 4 地下水の化学的性質

#### (1) 地下水供給域

地下水供給域は、地下水の水質形成に関する地球化学的プロセスの初期条件を考える上で重要な要素となる。この領域においては、広く土壌が分布しており、生物圏との接点として有機物を介した化学反応が活発に生じている。特に、有機物の腐敗は、地表水に含まれる溶存酸素を消費することで深部地下水が還元性を維持する一条件となる。また、花崗岩や石灰岩の風化によって生じた表層は植物の成育に必要な養分に乏しく、従って、植生もそれほど発達しないため、有機物を介した化学反応も活発とならない傾向がある。

#### (i) シミュレーションのための仮想環境の構成

地下水供給源の仮想環境の構築は、ボーリング・データ等より物理環境のクラス階層をポリゴンを形状モデルとして組み立てることが期待できる手法と考えられる。地下水と一般に呼ばれている対象では、地下にあるすべてがすべて地下水ではない。土壌の粒子を取りまく水がその量を増してくると、土粒子の間で保持されにくくなって、さらに下方に移動するようになる。このようにして地下に集まった水は、水を通しにくい層の上に集まって、さらに横の方向に移動するようになる。このような状態にある、地下の水を地下水と呼んでいる。地下水の発生は2層のモデルとして考えることができる。物理モデルは土粒子であり、これに付着する水粒子のモデルは、オブジェクト指向では、前述の破碎帯と同様な手法で土粒子モデルを構成し、水粒子の飽和条件を適当に与えることにより地下水を発生させることができると思われる。地下水は、飽和的に地下の空隙を満たして存在している。これを飽和帯という。このような水体の上面より上の方は水で飽和されているのではなく、むしろ空気の方が多い。このような部分を通気帯と呼び、これと飽和帯との境界が地下水面である。地下水は図3. 5-8のように分類される。



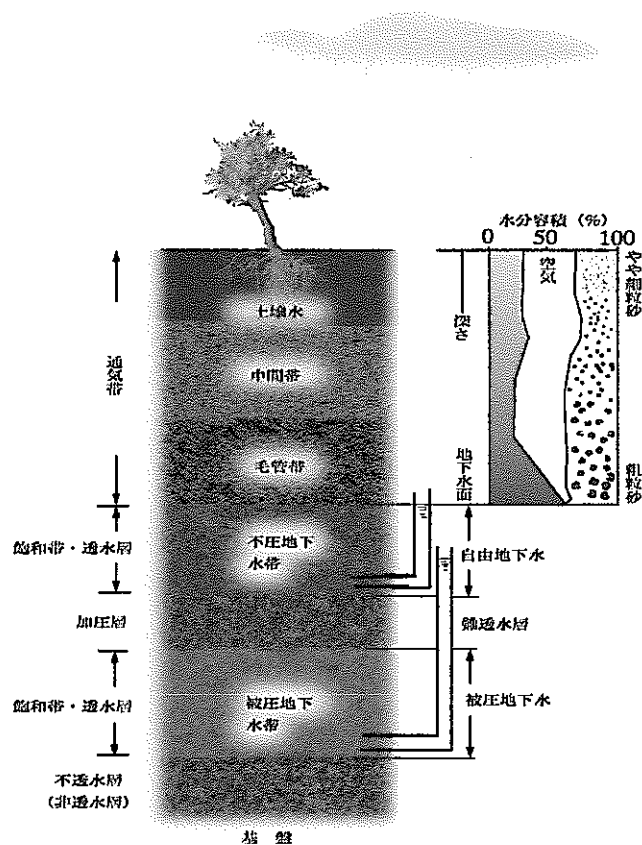


図 3. 5 - 8 地下水の分類

固結していない礫や、砂のように水をよく通す地下の地層を透水層という。シルトや粘土のように水を通しにくい地層を難透水層という。これに対して、固結した岩盤は非透水層であるが、難透水層と区別する必要のないときは、これらを不透水層とよんでいる。地下の状態は、透水層と難透水層とが重なり合って互層をなしていることが多い。このような帯水層中の地下水は、上下の難透水層に挟まれて被圧されている。これを被圧水という。被圧されていない地下水を自由水という。地下水が発生する限度は涵養と地質構造との相互関係によって決定され、これを水文地質構造と呼ぶ。地形・地質と地下水との一般的関係を表 3. 5 - 1 に示す。

表3. 5-1 地形・地質と地下水の関係

大地形	小地形	地質	賦存形式	地下水位
平地	沖積谷, 海岸平野, 三角州, 扇状地, 氾濫原, 砂丘	沖積層 (洪積層)	自由地下水 被圧地下水 一部水脈	浅い
台地	隆起海岸平野 隆起扇状地 段丘	洪積層 (第三紀層)	自由地下水 被圧地下水	比較的深い
丘陵	丘陵(主として単斜・ 向斜の構造を有する)	第三紀層	被圧(一部自由) 地下水, 一部水脈	深淺様々である
火山斜面	火山斜面 溶岩台地	溶岩・軽石 火山岩層	裂か水脈, 一部水層 あり	湧泉以外に地下水 乏しい
山地	山地・準平原	基盤台	裂か水脈 洞穴水路	湧泉のみ

(建設省河川砂防技術基準(案)より)

表3. 5-1から解かるように、地下水のモデルを仮想環境として構築する場合は、第一に地形・地質のモデルを表3. 5-1の分類を参考に作成するべきである。この作業は、ボーリング・データ等を参考に理想化したオブジェクトとしてデータベース化することが重要である。いずれにしろ、地下水シナリオを中心に評価する場合は、この地形・地質の条件が基本的な環境要件となるため、それらのデータベースはシステム開発の必須事項である。

#### (ii) エージェントの扱い

エージェントの扱いについては、地質構造上の開孔によって地下水の発生、運動や地下水面の形成が異なるので、地下水を次のように分類して考え、発生と流れのメカニズムを前述の地形・地質のデータ上に組み合わせる手法が必要である。

- ・間隙水 (pore water)  
堆積物、とくに地質時代の新しい沖積物の中の地下水で、開孔の大きさは毛細管の程度である。水はこの間隙の間を縫って流れる。
- ・割目水 (fissure water)  
レツコ水ともいう。固結した、あるいは結晶質の岩石の中の地下水で、開孔の大きさはもちろん超毛細管であるか、毛細管状の枝をもっている。水は間隙水のように流れたり、空洞水のように流れたりする。
- ・空洞水 (cavern water)  
洞穴水、またはカルスト水ともいう。開孔は大きな導管、洞穴などで石灰岩や溶岩などの中の空洞中の地下水がこれに属する。地下水は貯留されている量がその大部分で、地下水体に供給される量、またはこれから流出する量の何十倍、何百倍である。地下水体は、地下の岩石中で平衡を保つ貯水池を形成している。重力

または、揚水によって引き出されるのは、ほんの一部分で、これをその地層の産出率という。残りの大部分は地層によって保留され、利用できず、これを保水率という。

地下水体は主として地質構造によってその範囲が限定され、その地下水体の形態が不圧帯水層、被圧帯水層、漏水性被圧帯水層などのいずれを形成するかは、地質構造の差によって判別できる。

このように地下水体を規定する地質構造を水文地質または水理地質構造 (hydrogeological structure) と呼んでいる。水理地質を図3. 5-9に示す。

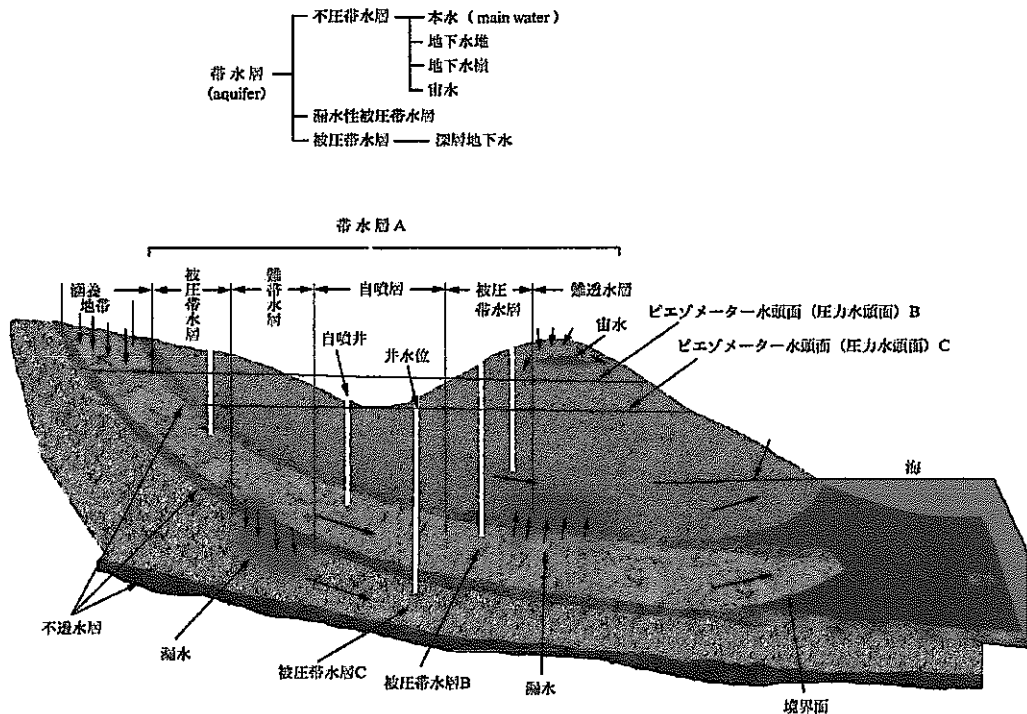


図3. 5-9 水文地質図

つぎに帯水層の水理地質構造であるが、地形図、地質図や現場調査からその発達状態を推定する。地形、地質から判断される一般的な帯水層の発達は次のように考えられる。

・河谷

現川の河床下、あるいはそれに沿って透水性の良い砂礫層からなる、旧河道堆積物が帯状に発達し、伏流水が流れていることがある。不圧地下水の場合が多いが、粘土質地層に被われていると被圧地下水の賦存することもある。

・堆積盆地

砂礫層、浮石層、粘土層が規則的に堆積し、地下盆地を形成している。盆地中央に掘られた井戸は自噴することが多い。

・堆積盆地

周辺部地下水の重要な滴養地域で、帯水層が河床下に直接露出する場合は、川の流量の減少が認められることもある。

以上のように地下水供給域と流れは、水分地質図によりとりまとめられた情報を参考にオブジェクトやモデルを組み立てることが適当であると考えられる。但し、これらは、一般に浅部地層領域のモデルであるので、深地層処分においては当面問題視されないであろう。従って、地下水の発生とその種類、それらの地球化学的性質がモデル化の課題と考えられる。

## (2) 深部岩体地下水の化学的性質

深部岩体中の割れ目を有する系においては、二次空隙の近傍において、粘土鉱物、鉄鉱物及び炭酸塩鉱物に富む場合がある。また、反応生成物である二次鉱物、特に粘土鉱物が滞留するような条件下においては、二次空隙はこれらの鉱物によって充填され、割れ目のもつ水理学的な不連続性を次第に失う。このような微視的構造に対応して原則的には、二次空隙中には循環水が、一次空隙中には岩石の生成時点の水（固結の進んだ堆積岩の場合、堆積時の海水が陰イオン排除により閉じ込められた化石水等）が満たされていることとなる。しかし、実際には、これらのそれぞれ異なる水は、溶質の拡散により混合されているのが普通である。他方、割れ目の少ない堆積岩中では、粘土鉱物に富む層で、かつ、陰イオン排除が生ずる程度に地圧の高い深部で形成されたものにおいては、化石水が空隙を満たしているものと考えられる。これらの2条件のうち、少なくとも一方を満足しない層においては、循環水が存在すると考えられる。この場合にも、これら2種類の水は溶質の拡散による混合作用を受けているものと推定される。

### (i) シミュレーションのための仮想環境の構成

深部岩体の仮想環境を造る基本点な手順は、前述してきた方法と基本的に同じである。図3.5-5は試作したオブジェクト・シミュレーションから視覚化したものである。これは、乱数より生成したポリゴン球を用いた空隙の構成例を示している。球体ポリゴンの表面特記を乱数生成して微小岩片を近似し、それを適当に立体空間に詰め込むことにより、恰も砕いた岩や砂を箱に詰めたようなモデルを造りあげることができる。但し、深部岩体地下水の環境の関心事は、核種の溶質移行にかかわる地下水の地球化学的性質にあるため、岩体自体は前述してきたモデルが利用できる。但し、空隙については、空隙率が影響するKdなどをモデルとして考えた場合に重要となる物理モデルである。従って、仮想環境を構築する上で注意すべき点は、空隙の再現であると考えられる。

### (ii) エージェントの扱い

深部岩体地下水の化学的性質に係わる関心事は、地下水と地下水に含まれる不純物、岩体中の無機物、及び廃棄物（人工バリア）との化学的相互作用を評価することである。それは、この相互作用が及ぼす核種輸送の遅延効果等を期待することを含んでいる。従って、この問題は、核種輸送媒体としての広義の地下水の浸透流解析に含まれると考えられる。浸透流解析については、鉱物組成及び微視的空隙構造にて考察する。

深部岩体地下水の化学的性質について、それが、個々の独立した水分子の振る舞い（タスク）として動作するオブジェクトを考えることができる。その時、水溶液中の分子は化学平衡、酸塩基反応、

電離平衡等の反応や、沈殿・溶解、酸化還元反応等、化学的振る舞いをする。分子と分子の間には、いろいろな引力があり、これが分子の熱運動とつり合って、気体、液体、固体など物質の巨視的な状態を作り、また溶解性、吸着性などの性質を成分分子に与えている。

従って、独立した水分子のオブジェクトを考えた場合、それが前述のような反応を起こすための性質として、オブジェクト指向モデルの考え方を与えることができる。例えば、水分子というクラスを考えた場合、それは質量を持った微小粒子（インスタンス）に物理的性質を与える変数（引力等の物理関数）を定義すると考える。このように物質自身をインスタンスとして考え、そこに働く力を関数とすると、一つの分子のオブジェクトをモデルとして考えることができる。このオブジェクトは、相互にメッセージパッシングすることにより、互いの状態を変化させることができると考えられる。そのようなオブジェクトを十分な量だけ準備して、初期条件を与えて計算を行うモデルを考えることができる。その時、個々のオブジェクトが相互にメッセージを交換し、物理関数として与えられる状態によって性質を決めることが出来ると思われる。勿論、原子レベルに落とすことも考えられるが、計算量等から当面（超高速コンピュータの出現まで）は現実的でないと思われる。

- ・ファンデルワールス力（関数：Van der waals power）  
ファンデルワールス力は分子間に働く弱い引力であり、分子が近づいたとき現われる一種の静電的な力である。
- ・水素結合（関数：hydrogen bond）  
ファンデルワールス力より強い、分子間引力である（2~8kcal/mol）。これはO、N、Fなどの強陰性元素の原子が水素原子を介して行う結合である。
- ・配位結合（関数：Coordinate bond）  
配位結合は電荷の移動を伴って形成される共有結合であり、イオン結合と共有結合の中間の性質を持っている。
- ・分子（イオン）の重さ、密度、大きさ（インスタンス）  
分子（イオン）間に働く力のほかに、分子の重さや密度、大きさそのものも性質をあらわす。
- ・溶解（関数：dissolution）、分配（関数：partition）、沈殿（関数：precipitation）  
液体の分子間引力は、完全に混じり合う場合と、一部しか溶け合わず2相に分かれる場合とが生じる。溶解、分配、沈殿はそのような性質を表す。
- ・吸着（関数：adsorption）  
気体または溶液中の分子が、固体表面に吸着される現象も以上と同様の親和力で説明される。

### 3. 5. 5 鉱物組成及び微視的空隙構造

#### (1) 結晶粒の等方的分布

砂岩、深成岩及び片麻岩では、基本的に結晶粒が等方的に分布し粒間に空隙が存在する。このような空隙中の物質移動の支配的メカニズムとしては、基本的には拡散を考えることができる。ただ砂岩については、水理学的条件によって、移流が物質移動の支配的メカニズムとなる場合もあると考えられる。

花崗岩類、及び同系の片麻岩を構成する主要な鉱物は斜長石（Naに富む）、アルカリ長石、石英及び黒雲母であり、特に斜長石、アルカリ長石については、変質に伴う空隙が見られる例が多い。ハンレイ岩類及び同系の変成岩においては、これに代わって斜長石（Caに富む）と角閃石、輝石、カンラ

ン石が主要鉱物となる。

また、砂岩については、石英を多く含むが、同時に、長石、雲母、粘土鉱物、炭酸塩鉱物、及び硫化鉄等の鉱物を含む。花崗岩類についても、変質層の近傍においてはこれと同様の鉱物組成が考えられる。

#### (i) シミュレーションのための仮想環境の構成

深部岩体地下水の化学的性質の場合と基本的に同じ考え方やクラスを利用できる。基本的な考え方は次のとおりである。鉱物組成の及ぼす影響は岩盤力学と中心とした物であるため、結晶岩質等の鉱物質のクラスを考える。例えば、それは立方体で密に詰めるものとする。また微視的空隙構造を再現するため、微視的な亀裂を持つクラスとしてポリゴンによる不均一な多面体を考える。次に、対象とする微視的空隙構造をもつ岩質を空間として考え、先ほどの2種類のクラスを適当な配分（乱数等）で詰める。そのようにして、微視的空隙構造を含む結晶岩質の仮想環境を造ることができると考えられる。

#### (ii) エージェントの扱い

考えられるエージェントは、微小鉱物組織と水分子のオブジェクトである。これらの協調は前述の水分子の性質等の物理量を関数とした連系のモデルを考えることができる。

### (2) 粘土粒子等の異方的分布

泥岩、頁岩、結晶片岩においては、シート状の粘土粒子等が方向性を持って重なり合い、そこに、石英や長石の粒子及び有機物片や黄鉄鉱が入り込むという配置を示す。曲ガラス質マトリクス火山岩は、生成時に比較的速やかに冷却されたことにより、非晶質のガラスをマトリクスとして有する。この中に、黒雲母、石英、斜長石（Naに富む）あるいはカンラン石、揮石、斜長石（Caに富む）が点在する。また、ガラス質部分の変質に伴い、ある領域がスメクタイト等の粘土鉱物によって占められる場合もある。

#### (i) 仮想環境の構成

深部岩体地下水の化学的性質の場合と基本的に同じ考え方やクラスを利用できる。基本的な考え方は結晶粒の等方的分布と同じである。

#### (ii) エージェントの扱い

考えられるエージェントは、微小鉱物組織と水分子のオブジェクトである。これらの協調は前述の水分子の性質等の物理量を関数とした連系のモデルを考えることができる。

### (3) 浸透流モデル

鉱物組成、及び微視的空隙構造の間を流れる地下水のモデル化手法は、一般に地質媒体の特性により2つに大別した方法が取られている。第一に水の飽和状態になっている岩盤中の流れであり、第二に、不飽和状態にある岩盤中の流れである。これらの流れのモデル化にあたって、熱の影響を入れる

か入れないかで更に分割して考えることができる。

#### (i) 飽和流れモデル

飽和流れモデルは、等価多孔質媒体に置き換え得る多孔質媒体や破碎媒体中の地下水流れ分布のモデル化である。それは、等温での流れ状態や、流れが圧力と熱により決定される一般的な非等温での流れ（熱一流れ連成モデル）を記述する。等温での飽和流れモデルは、処分システムの近傍では温度上昇域が流れ場に影響を及ぼすような現象に限って役に立つ。熱流れ連成モデル（非等温）は、熱により引き起こされた流れ（対流）と、伝導や移流による熱移動の相互作用を記述する。この種のモデルは、ニアフィールドの流れを解析するために利用する。飽和流れには、等温飽和流れと非等温飽和流れがある。

#### (ii) 不飽和流れモデル

不飽和流れモデルは有孔性媒体のモデルである。不飽和流れ状態は、処分場建設施行段階と初期貯蔵段階で岩盤が排水する処分場近傍で生じる。このモデルには、基本的に以下の3種類の不飽和流れモデルが考えられる。それらは、空気相を無視した不飽和等温流れモデル、空気と水の二相不通和等温流れモデル、蒸発と水の非等温流れモデルである。等温不飽和流れモデルは、ファーフィールド評価等に用いられる。ニアフィールドや処分場設計にそれらのコードが適用できるのは建設施行時の水流入量から透水性を逆算して求めるような場合に限られる。

#### (iii) 破碎媒体中の流れ

花崗岩や玄武岩のような固い岩盤中の処分場では、ほとんどの地下水の流れは不連続な破碎帯（割れ目）を通して流れると考えられる。スケールを充分大きくとれば、一連の割れ目は等価な多孔質媒体としてモデル化できる。しかし処分場近傍や断層のような大規模な破碎帯に関しては、特に破碎帯専用のモデルを用いるべきである。破碎帯内の流れをモデル化するアプローチには、不連続破碎帯（ディスクリフトフラクチャ）法と二有右孔（デュアルポロシティ）法がある。二有右孔法は2つの媒体を重ね合わせたモデルである。一つは岩層であり、もう一つは破碎帯である。移動関数は、岩層と破碎帯間の地下水の移動を記述している。不連続破碎帯モデルは、一般的に割れ目中の流れがほぼ平行板間の流れで近似できることを仮定して造られる。岩層中の流れは、多孔質媒体中の流れとして解析する。

#### (iv) 破碎体中の溶質移行モデル

ほとんどの破碎媒体中の溶質移行コードは飽和流れ状態を仮定している。これらのモデルは媒体が十分に飽和に達した後、処分場近傍で廃棄物からの放射性核種移行を評価することを目指している。破碎媒体中の溶質移行を解析する際に特に重要点は岩層への吸着である。

### 3. 5. 6 ニアフィールド問題

ニアフィールドは、処分施設と人工バリアの影響が大きいという意味で、地下水流動の観点からは、局所的流動系の特異な領域と考えられる。ニアフィールドは埋め戻された坑道群と人工バリア、及び

これらが建設されたことによる影響を受けた周辺母岩とから成る。

### (1) 坑道群

廃棄物を定置する処分坑道は、複数のパネルに分割され、各パネルの外周は主要坑道と呼ばれる資材や人員運搬用の坑道で囲まれる。また、地上施設と地下施設との間はアクセス坑道で連絡される。建設、操業及び閉鎖は、パネル単位で行い、処分場全体として見れば、これらの工程が並行して進むことがあり得る。

#### (i) シミュレーションのための仮想環境の構成

仮想の坑道群は設計情報等から3次元のデータとして作成する。この時、独立したエージェントとして扱うべきと考えられる要素（例えば、アクセス坑道、処分坑道等）は、個別のクラスとして定義する。また、それらの接続は力学系の関数を利用して行うべきである。それにより、仮想の坑道群のは、地震や断層等、近くの力学的条件に応答することができるであろう。図3.5-10は、約100万ポリゴンで生成した、縦置きの処分坑道全景の配置である。図が示すように、仮想坑道は、処分坑道、アクセス坑道、緩衝材、オーバーパック、固化体の独立したオブジェクトを接続して構築している。従って、接続関数を利用することにより外部の力に応答して変形する手法が考えられる。

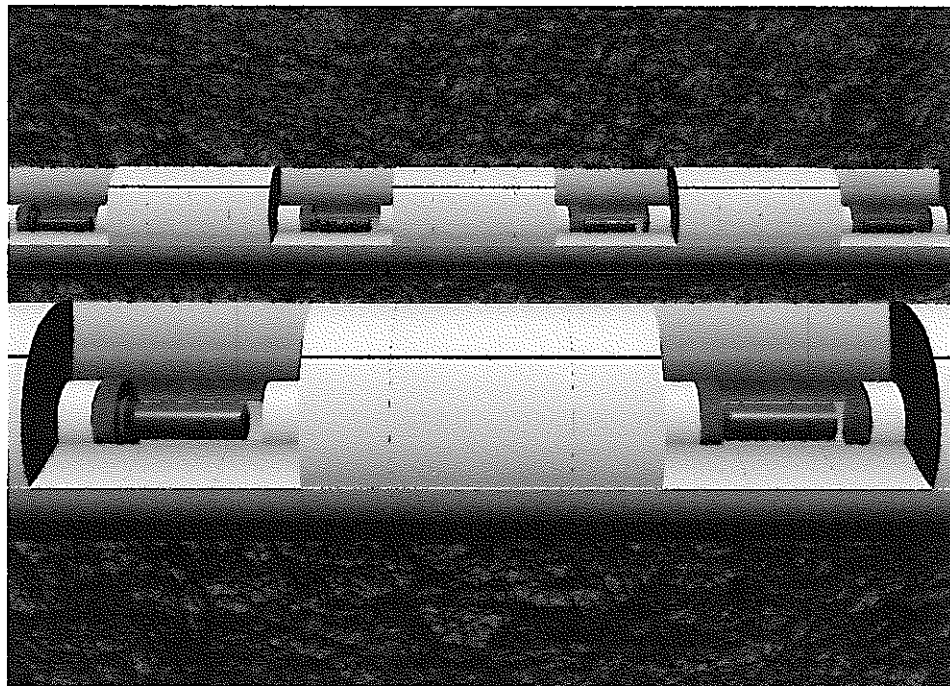


図3.5-10 仮想横置き埋設方式坑道全景

この程度のポリゴン数（100万）からなる仮想世界を用いて、視覚的に有る程度の満足感が得られる量の水分子エージェントモデルを用いたシミュレーションは、恐らく数十テラフロップスの計算速度（現在再高速の超並列計算機の約1000倍の計算速度）のコンピュータを利用することができるよう



になってから可能であろう。図3. 5-11に図3. 5-10に示す仮想坑道の内部を示す。

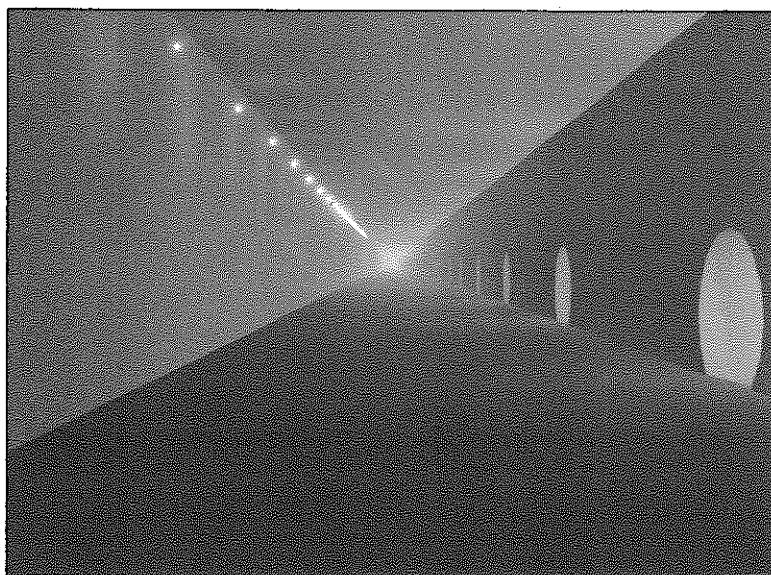


図3. 5-11 図3. 5-10に示す仮想処分システムの内部

## (2) 人工バリア及び周辺母岩

人工バリアは、ガラス固化体、炭素鋼オーバーパック、及びベントナイトの緩衝材によって構成され、処分坑道内に水平に設置（坑道横置方式）あるいは処分坑道に垂直に掘削された処分孔に設置（坑道縦置方式）される。周辺母岩は、処分坑道の掘削により力学的、水理的に影響を受け、ゆるみ領域を形成する。人工バリア周辺の地下水の流動については、既に述べた母岩の局所的流動系における異方性、不連続性の向きと処分坑道の向きの組み合わせによって種々のパターンが考えられる。

### (i) 仮想環境の構成

人工バリア及び周辺岩盤の仮想環境は、前述の仮想坑道を構築するクラスのサブクラスとして次の順序のクラス階層を構成する。周辺母岩、緩衝材、オーバーパック、ガラス（キャニスタ付き）である。なお、埋設方式に縦置きと横置きの二通があるのでそれらは別個のオブジェクトとして分類して構成すべきである。オブジェクト間の接続（例えば、オーバーパックとベントナイト）は、人工バリア内の複雑な現象を記述する複数の関数によるものとして考えることができる。

例えば、母岩の応力、オーバーパックの腐食等、場の状態や時間によって変化する物理量を接続関数として考えることができる。以下に試作したオブジェクトより仮想処分システムを視覚化した景観を示す。

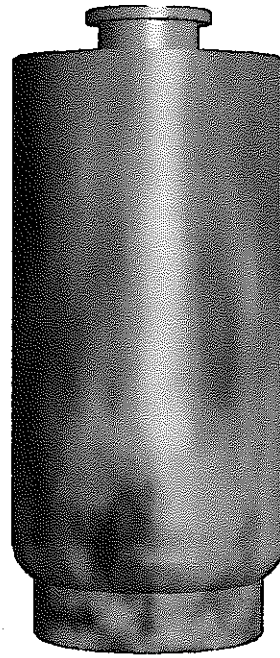


図 3. 5 - 1 2 縦置方式 固化体キャニスタ全景 (TVF)

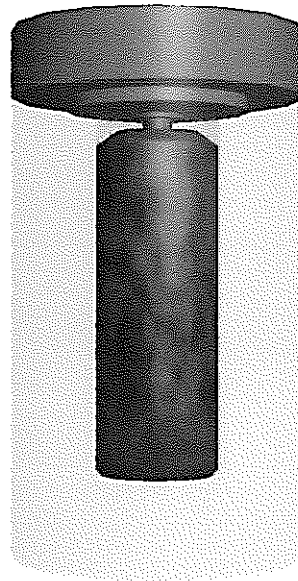


図 3. 5 - 1 3 縦置方式 固化体キャニスタ+オーバーパック

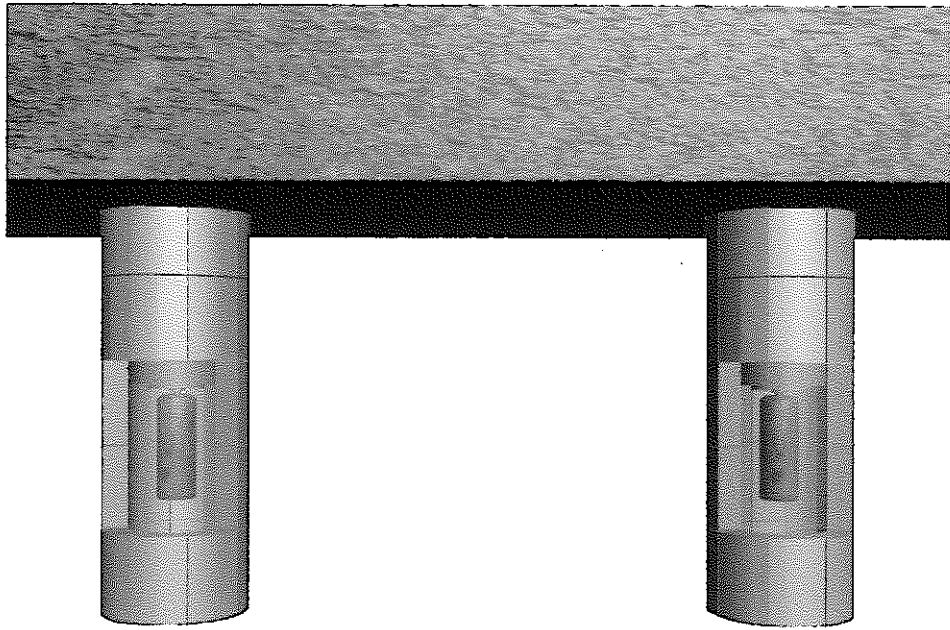


図 3. 5 - 1 4 豎置方式 固化体キャニスタ+オーバーパック+ベントナイト

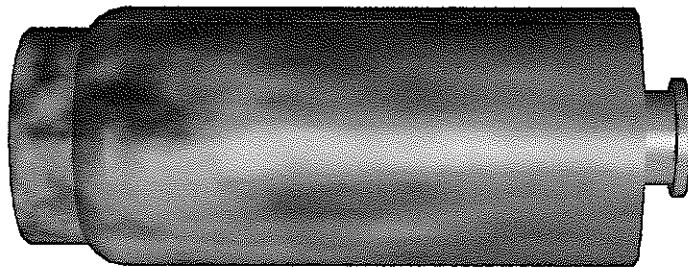


図 3. 5 - 1 5 横置方式 固化体キャニスタ全景 (TVF)

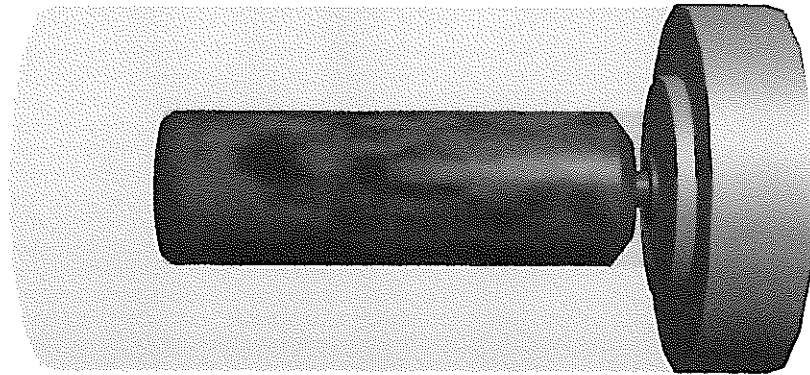


図 3. 5 - 1 6 横置方式 固化体キャニスタ+オーバーパック

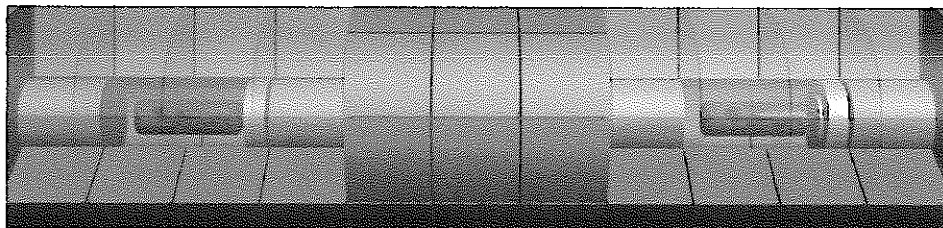


図 3. 5 - 1 7 横置方式 固化体キャニスタ+オーバーパック+ベントナイト

### 3. 5. 7 連性問題へのアプローチ

性能評価では、地下水の流れとともに移行する核種の異なるプロセスが連成した現象にかかわる知見が必要である。このような連成モデルの研究が、熱伝達、地下水挙動、岩盤力学、及び地球化学を含む、処分システムの長期的性能に影響する重要なプロセスに関する理解を深めるため多面的にすすめられる必要がある。この研究の焦点は、まず、異なる地質媒体ごとに、どの連成相互作用が重要であるかを系統的に明らかにする事である。連成モデルを開発するためには、その準備として以下の事項を事前に整理することが必要である。

- ・連成プロセスの分類とリストアップ
- ・現在の知識でとらえ得る限りの処分システムの長期性能に影響する各プロセスの重要度評価
- ・各プロセスの不確定性のランク付け
- ・不確定度の低減化を計れるとキーとなる課題の解決の向けての研究法の検討

例えば、流体・熱システムや、処分システム周辺と同様の条件であると想定される、天然岩石・水システムに関する情報の整理が必要である。そのようにして、上記の研究課題が適正に位置付けられ、

どのような条件の場合に連成相互作用が重要となるかをより良く理解できるようになる。処分システムの性能予想には、現実には発生し得る主要なプロセスに対する合理的な見通しが必要である。

主要な物理化学的プロセスは、熱 (T)、水理 (H)、力学 (M)、及び化学 (C) である。これらの4つの中から、考えられる連成の組み合わせは11種 ( $4C_2+4C_3+4C_4$ ) である。なお、連成のモデルには、各プロセスが1方向的に時系列で連成する形式と、相方向的に連成する形式がある。

これらの各連成プロセスを、全て包括するような評価システムを決定論的に導くことは難しいであろう。いくつかは、解析できるようになり始めているものもあるが、全く手が付けられていないものもある。オブジェクト・シミュレーションはそのような限界を越える一つのアプローチである。

並列に動作する独立したタスクとしてのオブジェクトをエージェントとして定義して、それらを協調させてシミュレーションをすすめるマルチ・エージェント・モデル協調計算の手法で連成モデルを開発することを考えた場合、従来の複数本の支配方程式を連立させて差分法や、有限要素法で離散近似するシミュレーションに比べ、遥かに複雑のモデルを造ることが期待できる。何故ならば、個々のプロセスは実験等により定式化されて得られるため、それを独立した関数として受け付けるクラス階層は自然な連成記述を可能にするためである。

そのような、クラス階層を考える場合は以下のような分類を事前に行うことがよいと考えられる。

#### 00 : 熱と化学の連成

熱拡散、相変化 (固体の平衡状態)、固体の分解、準安定固体、等

#### 01 : 熱と流れの連成

浮力流れを含む1相あるいは2相対流、相変化とInterference、熱的物性変化、熱的浸透、ガス拡散、毛管・吸着、等

#### 02 : 熱と岩力の連成

熱クラック、破壊変形、熱破碎、熱クリープ、熱膨張、等

#### 03 : 化学と流れの連成

分離、同位体変換、錯体、共沈、溶解、沈澱、吸着・脱離、酸化還元反応、加水分解、酸性反応、拡散、化学浸透と限外濃過、等

#### 04 : 流れと岩力の連成

水力学的破壊、間隙水圧変化、破碎帯の水力学的エロージョン、粒子の沈澱、剥離を引き起こすせん断効果、亀裂開口の変化、粒子の濾過、等

#### 05 : 化学 (弱) 熱 (強) 流れ (強) 化学の連成

溶解、沈澱、時間依存の溶解沈澱、浸透による流体移動、気相での化学的移行、気体と固体の分離、粒子の移動 (コロイド)、平衡状態、熱拡散、熱的浸透、等

#### 06 : 熱 (強) 岩力 (弱) 化学 (弱) 熱の連成

鉱物相の相変化、脱水、クリープ、脱水とスウェリング、等

#### 07 : 熱 (強) 岩力 (強) 流れ (強) 熱の連成

水力学的破壊、地震の引きがね。(応力再分配、間隙水圧)、岩盤の割れ目の開閉挙動 (応力再分配、熱的接合、間隙水圧)、破碎、強度変化 (応力腐食、加水分解)、水和作用、等

#### 08 : 岩力 (弱) 化学 (強) 流れ (強) 岩力の連成

このプロセスは考慮しない。何故ならば、熱的な効果が無ければ、物質移動は、低温での沈澱、応力腐食、及び、イオン交換性スウェリング (50°C程度以下) を除いて、形状変化につながるようなものではないからである。

## 09：熱、岩力、化学、流れの複雑な連成モデルの連成

貫孔作用（選択性溶解とトンネル腐食）、非均一浸透、沈澱（破碎帯、母岩）、岩の水熱的変性、溶出と沈澱と伴うヒートパイプ効果、オーバーバック廻りの鉛直方向蒸発（擬縮サイクル、圧力溶解）、等

### 3.6 数値実験オブジェクトとして考えた評価モデル

処分システムの安全性評価のためのモデルは、放出核種の移行過程等を表現する複数のモデル（サブモデル）から構成されている（評価モデル・システム）。これらのモデルは、対話システムから見ればバックグラウンドで独立したタスクとして動作するオブジェクトとして扱うことが好ましいと考えられる。また、それらは、対話システムのウィジェットや他のタスクからのメッセージで起動され、並列的に協調して動作する必要がある。サブモデルは、基本的には核種の移行過程をできる限り正確に表現するものであるべきであるが、一方、評価モデル・システムは総合的な解析を行える機能が重要なため、サブモデルは簡単なものが望ましい。

詳細モデルに於ては、我が国では火山による飛散、地殻変動による露出等の直接被曝シナリオも他国より慎重に評価する必要があるが、基本的にはサイト特定でそれらの要因を削除できると考えられる。例えば、地殻のゆっくりした上下動等、必ず起きる地殻変動として評価しなければならない事象もあるが、その場合は、境界条件の時間的变化による地下水の動きへの影響を考慮することになる。これらのことから、我が国においても、地下水による核種移行モデルが評価の基本となると考えられる。従って、モデルの選定とシミュレーションの実行においては、サイト特定で削除できる要因等を明確に表示し、評価シナリオを簡素化する操作を視覚的に表現することが好ましいと考えられる。

#### 3.6.1 各種のモデル

##### (1) 容器の耐久性評価

人工バリアは固化体（キャニスタ入り）、容器、緩衝材からなる。容器の健全性は、核種の移行の開始時期を決めることになる。容器は完全封入機能を持ち、腐食等の破損による開口部が生じた時点でバリア機能を消失するとして評価する。小さな開口部であっても、一旦開口部ができるとバリア機能が他のバリアの機能に比べて極度に低下する。容器は、キャニスタとオーバーバックの組み合わせを考え、代表的材料としてキャニスタはステンレス鋼、オーバーバックは炭素鋼が提案されている（図3.6-1、図3.6-2参照）。

容器の耐久性の評価は、腐食速度の評価と地震等の瞬間的な破壊につながる要因の有無の評価にわけて考える。また、全面腐食、及び局部腐食の速度に影響する要因として、温度、酸性度、地下水の組成、放射線線量率、酸化還元電位等について考える。シミュレーションに於ては、これらの環境条件の時間変化を求め、その環境条件での腐食速度の積分値として腐食厚さを計算する手法等が用いられている。オーバーバックの腐食現象に係わる評価は、化学過程を中心としたシミュレーションになるものと思われ、シミュレーション結果の視覚化表現には新たな手法の開発が必要である。

図3.6-1は、試作したオブジェクトから腐食を視覚化したものである。また、瞬間的破壊としては応力腐食割れと機械的破壊が考えられる。これらの破壊についてはあらかじめ割れの生じる環境条件を実験的に明らかにし、その環境条件が処分環境にないことを確認する必要がある。

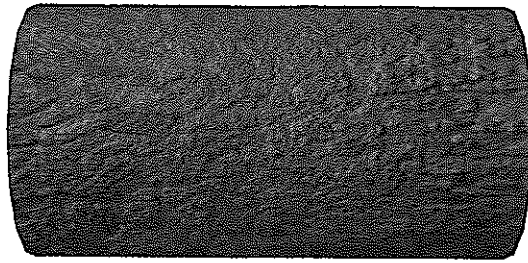


図 3. 6 - 1 オーバーパックの腐食概念

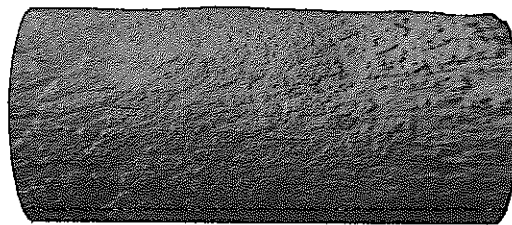


図 3. 6 - 2 オーバーパックの変形概念

機械的破壊については、破壊応力以上の外力が加わらないことを評価する。これらの要因が生じた時点で容器は破壊しバリアとしての能力はなくなるものとする。これらの破壊についても対話システムは優れた視覚化機能が要求されると考えられる。

## (2) 核種浸出評価

貯蔵時、輸送時の評価においては、時間に比例した浸出が生じるとして浸出率により評価する。処分システム操業後のソースターム評価においては、処分環境の地下水中的での平衡反応から決まる溶解度モデルと拡散、及び地下水の動きに伴う物質移動により評価する手法が取られている。

溶解度モデルによる評価のパラメータとしては、処分環境の温度、地下水の組成・pH-Eh、共存固相（ガラス、容器、緩衝材等）の組成が挙げられる。放射線は水の放射線分解をもたらし、Eh-pH等を決める2次要因として重要である。これらの平衡反応による評価要因に加えて、より溶解度の低い結晶性化合物にゆっくりと変換し、実質上、核種移行評価系から除かれる、いわゆる鉱物化も重要な評価シナリオである。

このような、地球化学シミュレーションについて対話システムは、熱力学データベースと連結したパラメータ・サーベイや感度解析機能を提供し、評価作業を支援する機能が求められる。また、化学反応を視覚化する手法の開発も大切であると考えられる。

地下水の流れについては、人工バリア内での地下水の動きは極めて遅いことから、主な物質移行機構は拡散になることから、ガラスの表面での核種の溶解度、及び緩衝材中の分散係数が物質移行評価

の重要な評価パラメータである。ガラス固化体、及び人工構造物の特性は人工的に決まるため、この部分には不確定要素は少ないが、浸出評価における不確定要素は主に自然環境の不確定性による。

従って、長期に亘る地下水流れを視覚化し、前述した化学過程を重ね合わせる等の表現技術の開発が対話システムに必要である。それにより、非常に緩慢な拡散中心とした流れに添って反応する過程の視覚化表現が期待できる。それは、核種の放出と輸送の過程をシミュレーションすることを目標としている、地層処分ヴァーチャル・エンジニアリング・システムの対話機能開発の重要な課題と考えられる。また、その手法は次項で触れる地層中での核種移行評価についても同様である。

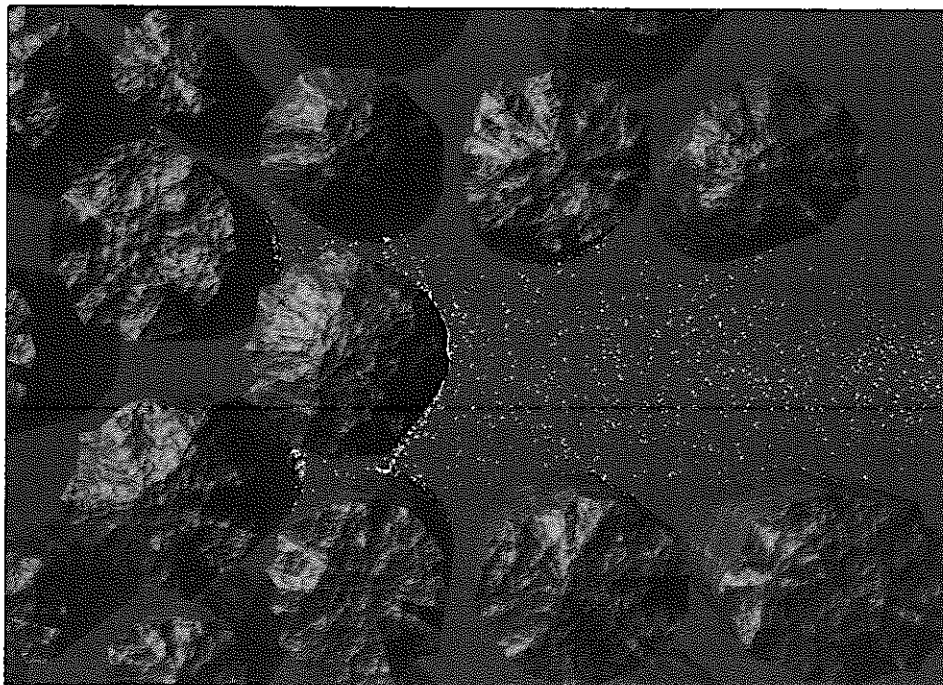


図 3. 6 - 3 物質移行の例

### (3) 地層中での核種移行評価

地層中での核種移行評価も基本的には核種浸出評価と同じ構成である。しかし、地層中においては評価上最も重要な部分は地下水の動きである。特に、破碎帯等の母岩に比べ極端に水が通りやすい、いわゆる水みち（水理）の分布である。また、母岩の間隙水は一種の貯留タンクであり、母岩の中への分散速度も重要な評価要因である。従って、地形、地層構造、母岩及び水みちの透水性、母岩中のイオンの分散係数が主なパラメータであると考えられる。地層中での化学反応モデルでは、可逆反応モデルと鉱物中に取り込まれ、実質上、溶液系から不可逆的に取り除かれる鉱物化モデルの組み合わせで考えることができる。原子価によって著しく溶解度の異なるNp、U、Trにとって還元性物質の存在は評価上重要である。即ち、溶解度、鉱物化の反応速度、還元物質の含有量とその還元特性が重要な評価要因である。これらの要因にまで立ち入った情報を自然環境で得るには、更に反応の種類、基礎データ（熱力学データ、反応速度等）に関する研究が必要であり、評価環境を想定して必要な吸着分配係数、溶解度を実測し、より汎用性のある基礎データとしてまとめていくことが必要である。



ソフト面では、PHREEQE等で代表される多成分系での核種の安定な存在状態を求める計算コードの開発を進め、更に反応速度、濃度依存性等を考慮した吸着移行評価コードを充実し、組み合わせる必要がある。データベースについては、熱力学データや吸着係数の収集が考えられる。

#### (4) 生物圏中の核種移行評価

生物圏に到達した放出核種が人体に至るまでの過程は、原子炉、再処理施設等の地上施設から放出される核種の挙動と類似であり、そのモデル化及び関連パラメータについては多くの研究が行われている。そのほとんどが適用可能と考えられる。ただし、TRU等の長半減期核種に関する情報、データは、まだ十分ではないので、この分野に関する一層の研究が望まれる。従って、生物圏への評価に備えて核種の崩壊等のデータベースや移行率、量等の作図機能等が機能に求められる。

### 3. 6. 2 パラメータの不確定性

地殻変動、地震等の接近シナリオと呼ばれる範疇に含まれる自然現象は確率的事実であり、いつどこで発生するかを決定論的に予想することはできない。しかし、過去の発生事象の統計とプレートテクトニクス仮説から、ある程度推定することができる。その手法として、第四紀における地殻変動の統計的データを使い、長期評価に適合するデータを得る。これらの現象による大きな影響はサイト特定で避けられるが、地震や雑続的隆起沈降の影響は考慮しなければならない。これらは、地下水による核種移行シナリオに影響する不確定性である。従って、地層処分ヴァーチャル・エンジニアリング・システム設計には、将来これらの地殻変動や地震等の事象を与えられる機能が求められる。

地層の透水性、地下水の組成、岩石の組成等、地層の特性情報は、原理的には処分サイトが決定されれば測定によって求めることができるが、サイトを破損しないで行う測定精度には限界がある（掘削影響）。そこで一般的に、類似地層での観測結果の統計処理により、それらの特性値を求めざるを得ない。具体的候補サイトが決まっていない現時点での評価では、地層の大分類の特性の統計的扱いをすることになると考えられる。従って、地層処分ヴァーチャル・エンジニアリング・システムは、仮想のサイトにて評価シミュレーションを進めるための情報として、前述の地下水や岩石の組成等、幅広く情報をデータベース化して参照を可能とするとともに、地形のオブジェクトや設計する処分システムを仮想環境としてコンピュータ内に、それらデータベースと関連させて構築できるものであることが望ましい。それらを構築する上での不確定な事項として下記が上げられる。これらの不確定さの原因については当面の間、事実情報を詳細に記述するデータベースを構築して、優れた対話機能により随時評価することが望まれる。

#### (1) 人工バリア関連パラメータの不確定性

材料及び施工等の品質管理上の不確定性がある。地層処分システムでは、地層特性の不確定性がそれに比較して大きいので、その不確定性は無視できる。更に重要なことは、人工バリアの健全性の不確定性は環境条件の不確定性に起因すると言うことである。特に経時変化が重要である。例えその経時変化が予測できても、長い時間の経時変化であれば、その条件での実証試験ができなく、モデルを使った予測に頼らざるをえないと思われる。

## (2) 生活圏関連パラメータの不確定性

遠い将来の人類の居住状況を予測することは非常に困難である。そこで現在の社会が変わらないものと仮定する。従って、サイト固有のデータを使うことは難しいであろう。陸地か海洋か、砂漠か農耕地か、井戸の利用か川の利用かと言う程度の区分で被曝経路シナリオを作り、評価することになると思われる。

## (3) 測定精度関連パラメータの不確定性

以上のようにパラメータそのものの特性として幅を持っている。また、測定技術が不十分である場合、技術があってもサイトの損傷を避けるために使えない場合がある。後者の場合は類似サイトを詳細に調査することでその精度を高めることができる。

## (4) 長期の不確定性に対する保証関連パラメータの不確定性

長期の事実を限られた期間で実証することはできない。それを補う一つの方法としてナチュラル・アナログ研究がある。評価事象の機構のモデルを作り、その事象のナチュラル・アナログを探し出し検討することにより実証的データを得る。この方法の難しさは、それが置かれた環境条件の経時変化を把握することが困難である点である。

## (5) 不確かさへのコンピュータによる演繹的推論

以上のような潜在的なモデルの不確かさに対する戦略として、知識工学分野の演繹推論等の応用がシステムのシミュレーション支援機能として期待できる。演繹とは、確かとされる前提から出発して、厳密な推論を重ねて新しい知識を導くことである。前提となる知識は、公理や命題、定理であり、推論には、三段論法、数学的帰納法、帰謬法などが使われる。そこから導かれるのは、再び命題や定理となる。この数学で使われる演繹的な推論の形式を整え体系化したのが、数理（形式）論理学であり、それをさらに発展させたのが数学基礎論である。現代の科学は、数学の表現法を借りている。

そこで、科学における記述も、前提命題、公理、証明法というように数学基礎論のような記述スタイルに整理できるのではないかという考えがある。このような公理的な記述法は、物理学で部分的に試みられてきたが、あまり成果が上がらなかった。つまり、そのような体系立った厳密な論理展開が実験に基礎を置く現代科学の進歩に必須というわけではなかった。また、一見厳密に思える自然科学の論理体系も、形式論理学で記述が可能なほどの抽象性と厳密さを持ち合わせていないことも明らかとなった。よく知られているように、コンピュータ内部の演算はすべて2進数の代数演算で行われている。この演算はブール代数と呼ばれる形式である。ブール代数は、数理論理学でいう古典論理と同等である。古典論理というのは、われわれの日常出会う自然な論理であり、古くから使われていた論理である。ついでに言えば、よく知られた論理としては、この他に直観論理、量子論理、ファジィ（あいまい）論理などがある。

コンピュータのブール代数的な演算機能を利用すれば、古典論理に基づく推論を機械的に実行させることができる。もちろん、そこでは1と0の列からなる入力データと出力データをどのような内容に対応させて解釈するかという作業が残っている。この部分に効率的な対話機能を開発すると、コンピュータに演繹的な推論を実行させることができる。演繹的推論システムの方法論は凡そ、以下のとお

りである。

- ・人間、特にある領域における経験者の知識をコンピュータで機械的に操作できるような推論規則として表現する。
- ・この推論規則を、あたかもデータのように考えて他のプログラムから独立させ、知識ベースとしてコンピュータに格納しておく。
- ・知識ベースを構築、管理したり、それを用いて演繹的な推論を効果的に展開することに適したプログラミング言語、およびそれを用いたシステム環境。

一般のソフトウェア・システムでは、人間の書き下す操作手順がプログラムの本体を構成する。ところがエキスパートシステムでは、経験知識の表現だけを他のプログラムからも、データからも独立させて定義して利用する。いずれにしろ、演繹推論等の知識工学はモデルの不確かさを補う情報を提供する可能性が期待できる。

### 3. 6. 3 システム操作面から見た評価シミュレーションの考え方

前述のように、処分システムの安全性評価の判断基準は、処分に起因するリスクが社会的に許容されるリスクレベル以下であるかどうかにおかれるものとする。そして、この判断のために総合評価が考えられる。総合評価は2種類あると考えられる。

第一に、代表的な(適切な)処分シナリオについて、適切な移行シナリオを選定し決定論的評価を行う手法である。決定論的評価に於てはシナリオが適切かどうか、評価結果の信頼性が十分かどうかについて議論がなされるであろう。それは、長期にわたる諸条件の変化やモデル・パラメータの持つ不確かさを考えねばならないからである。そのためには、多方面に渡る情報を評価とともに参照しながら作業をすすめる必要がある。このような決定論的評価を検証するための情報を効率的に提供することは対話システムの重要な機能要件である。

もう一つの方法に確率論的評価がある。それにより、処分システムの安全性の規制的観点からの評価は、決定論的評価手法によって行われ、確率論的評価手法は、決定論的評価を補完するものとして位置付けられると考える。確率論的評価手法は、放出核種の移行過程の変化確率及び移行過程を支配するパラメータの自然的、社会的条件の変化に伴う変動、または情報の不十分さによる不確かさによる最終のリスク評価結果の不確かさを定量的に解析するものである。

更に、評価結果の不確かさに大きく寄与するパラメータを明確に示す機能、いわゆる感度解析機能も持つものであると考えられる。また、確率論的手法は、シナリオを簡単なネットワークで画像表現し、操作を視覚ベースで行うシステム化の手法を提供するものとしても有効と思われる。

例えば、評価モデルとしては、廃棄物からの核種の浸出は一定割合で一定期間続き、緩衝材中の移行は拡散のみに支配され、地層中は一次元吸着平衡を保ちながら移行し、岩盤中の間隙水へ拡散し、飲料水として人間に到達して放射線被爆をもたらすという簡単なシナリオである。確率論的評価手法の開発では、手法そのものの開発とともに、入力データベースの開発も重要な問題である。そして、入力データの確率分布をできるだけ正確に知ることが必要である。

## (2) ヴァーチャル・エンジニアリング・シミュレーションの議論

シミュレーションは、およそ以下に示す議論について支援することが求められる。それにより、性能評価シミュレーション全体の不確かさをスクリーニングする必要がある。

- 議論 1：決定論的、確率論的、性能評価法の相対的な役割は何であるか。  
決定論的な最良の評価と、パラメータの不確かさについての確率論的な処理法を結び付けて考察することが必要。
- 議論 2：処分システムの性能評価を実行する時、確率論的不確かさと決定論的不確かさを区別して扱う必要があるか？それとも合わせて扱うべきか。
- 議論 3：定量化する必要のある全ての源に含まれる不確かさはどの様な範囲に渡っているのか。  
例えばモデルとシナリオに依る不確かさを定量化する方法を探さなければならない。
- 議論 4：不確かさ解析の結果を、どの様にリスク評価や結果の信頼度提示に用いることができるか。
- 議論 5：不確かさ解析に用いる、主観的な確率密度関数の定義はどの様になされるべきか。  
例えば直接測定が不可能な時、専門家の判断を用いる場合に問題となる。
- 議論 6：時間に依存するプロセスは、どの範囲に対して性能評価に取り入れるべきか。
- 議論 7：異なるパラメータ間の相関をどの様に見積ることができるか。
- 議論 8：リスクと総量率を評価する時、不確かさパラメータを扱う確率論的手法は規制側による決定を助成することに対してのみ適用されるのか？それは、処分に関する安全性のレベルを大衆に知らせるために用いるには余りにも複雑ではないか。

以上の議論から、地層処分ヴァーチャル・エンジニアリング・システムの開発においては、その対話システムの支援機能として、以下に示す要求にそった情報サービスを行える機能を開発する必要がある。

- 要求 1：定量的な手法と非定量的手法を結び付けるべきである。
- 要求 2：不確かさ解析の種類と型式は規制側の要求により決定されるべきである。
- 要求 3：決定論的な計算と、確率論的不確かさ解析を結び付けることにより全体的なシステム性能評価を導く一般的なガイドラインとすべきである。
- 要求 4：感度解析は、不確かさ解析に対し密接な関係がある。
- 要求 5：異なる不確かさパラメータ間の相関を扱う方法の開発が必要である。
- 要求 6：幾つかの二者択一的な、統計的サンプリング法がメリットを持っており利用可能である。
- 要求 7：主観的な確率密度関数の形において、パラメータの不確かさの表現がシステム的なアプローチに必要となり、特に主観的な専門家の意見が取り入れられる。
- 要求 8：コードの質的な評価がモデルの不確かさを減らすのに役立つ。
- 要求 9：ベンチマークによる検証と確認試験が、不確かさを減少させることの助けとなり、モデルのユーザーの信頼度を向上させる。
- 要求 10：システム的なアプローチは不確かさ解析を指導することに適用されるべきである。

### 3. 6. 4 ヴァーチャル・エンジニアリングの目指す科学技術

地層処分ヴァーチャル・エンジニアリングは、従来からの実験に基礎をおいた理論を構築する科学の手法では、アプローチの難しい（方法自身に限界がある）問題に対する、第3のアプローチとして今後研究開発する科学技術の領域である。そのアプローチの考え方の中心は、複雑系やカオスと呼ば

れる数学の分野である。また、その分野はコンピュータによる仮想世界のシミュレーション技術と密接に関係した世界である。それは、コンピュータの中に構築された世界で生じる現象を観察することにより、新たな定式化を期待しているためである。

ヴァーチャル・エンジニアリングに期待している問題は、現実の世界における解答が非常困難な問題である。ある問題が論理的に解答不能であることを証明する手段は数学でのみ見いだせる（不完全性定理）。従って、科学的知識とコンピュータ・プログラム（数理モデル）は、同等であると考えた場合、数学モデルも解答不能の決定に関与してくると考えられる。現実には、時間とか位置といった直接観測できる量と、それから導かれるエネルギーなどの物理量からなる。このような量は一般に離散集合を構成し、その測定値は有限集合から選ばれる。さらに、こうした選定は一般に不確実性を含んでいるのである。一方、数学では、現実の世界で観測されるものを記号で表現する。そこでは、空間も時間も連続体であると仮定される。さらに位置や速度などを表す数学記号は、整数とか実数や複素数などの値を取る。すなわち、どの集合にも無数の要素があることになる。さらに数学には確率論があって、それは偶然に影響されるランダムな事象を扱う。

最後に、コンピュータによる計算がある。これは、ハードウェアという物理的な条件と、ソフトウェアという抽象的な数学の条件をかね備えている。つまり、計算を一組の規則（アルゴリズム）の実行と考えるならば、計算は、記号と記号の関係を扱うことになり、純粋に数学的な対象となる。だが、計算をコンピュータのメモリー・スイッチをオン・オフに変えるプロセスと見なすならば、それは物理的に観察できる世界にしっかり根付いていることになる。科学的知識について、有効な理論を作れる可能性があるならば、これらの関係（現実、数学、計算）を最優先に留意しなければならないと考えられる。この三つが何か特定の問題（たとえば地下水による物質移行問題）を表現する限り、なんとかしてその三つを一致させる必要がある。

そのような問題の存在（または非存在）を立証するためには、二つの選択肢がある。すなわち、すべての議論を自然界のみに制限するか、自然界と数学および計算の世界をかみ合わせるかである。第一の方法を選び、もっぱら自然界の範囲内にとどまる。すなわち問題を数学的な表現に変えることを禁じるのである。また答えを出すために、論理的な証明メカニズムを使うことも同じく禁止してしまう。ここで難しいのは、科学的に答えを出せない（あるいは出せる）ことの証明が、私たちの目標となっている点である。しかし証明の概念そのものは、数学にしか存在しない。

したがって、私たちは現実の世界において、証明に代わる概念を見つける問題に直面することになる。自然界で証明に代わる第一の候補は、因果関係の概念である（例えば、FEP シナリオである）。これを採用してみよう。すると次のような主張が導かれる。もし因果関係の連鎖から得られた最終結果が問題の答えとなるならば、その問題は科学的に解答可能となる。しかしながら、複雑な状況下で確実な因果関係の連鎖を作り上げることは、たぶん大変な仕事になるだろう。特に人間が関与している状況下ではなおさらであるといえる。従って、コンピュータによる、複雑系としての総合的なシミュレーションがそれを立証する方法として上げられる。

地層処分ヴァーチャル・エンジニアリング・システムの開発では、ソフトウェア複雑化の要因を把握し管理することは重要である。一般にアプリケーション・システムは明確な開発目標を持っている。性能評価は、地震等の確率的に生じる事象による変化を除いて、地層中の環境が安定している場合、地層処分された廃棄体から放射性核種が放出され地下水により生物圏まで輸送されるシナリオを基本

としている。そのような安定した地層中での環境に添って核種移行を考えた場合、それだけでも非常に複雑な環境評価モデル開発と評価の問題であることが解かる。評価シナリオは、例えば温度場の変化が力学系に作用し、地下水流動を変更することにより人口バリアの冠水や亀裂を生じ、結果として各種が滲出して輸送されるシナリオが考えられる。このような、シナリオにそったシミュレーションを、対話方式で試行錯誤的に行なうには、下記の操作機能がアプリケーション・システムに要求される。

**操作機能 1 : シミュレーション・シナリオの一覧**

シナリオは、十分に評価され妥当なものとしてデータベース管理されていること。

**操作機能 2 : シミュレーション条件の決定**

シナリオの選択、個別評価モデルの選択、計算条件の設定等

**操作機能 3 : シミュレーションの実行と制御**

シミュレーションの実行、実時間視覚化、計算結果の解析等

また、可視化機能は、今後の性能評価シミュレーションの技術研究に於て重要な課題である“個別の計算モデルを複雑に組み合わせ協調して計算させる”技術の研究環境である。例えば、MIT X/11に代表される大規模で複雑なシステムプログラムの多くは、オブジェクト指向を開発戦略に取り入れている。それは、オブジェクト指向がソフトウェアの大規模化からくる複雑度の増大に伴う信頼性低下や、複数の計算モデルを組み合わせると言った要求に対する現在考えられる具体的な手法であるためと考えられる。従って、大規模で複雑な統合化シミュレーション・システムを開発するためには、オブジェクト指向の考え方を、物理工学シミュレーションで取られてきた要素還元論と合わせて整理することにより、計算科学の領域である大規模複雑系シミュレーションの技術開発戦略として期待できる。

オブジェクト指向は、プログラム開発に関する新たな方法論である。この方法論によれば、メモリー上に展開される従来のプロセス、タスクといった従来の実行状態の手続はオブジェクトと言う、より発展した概念に統合して考える。このオブジェクトの考え方に添って、物理・工学上の問題等をプログラミングすることにより、より複雑な問題を効率良くプログラムすることが期待できる。また、並列計算システムを利用することにより、より説得力のあるシミュレーションが期待できる。オブジェクト指向による、シミュレーション・プログラム開発には新しい考え方が必要であり、その影響はモデル開発にまで及ぶものと思われる。それは、シミュレーションの考え方を革新する思考方法と捉えることもできるであろう。

本研究では、以下の新しい概念とシステム開発技術の成果が得られた。従来研究開発されてきた、科学技術のトップダウン・アプローチ（現象全体を確率論的モデルで記述したもの等）とボトムアップ・アプローチ（現象全体を個別の要素に還元して個別の決定論的モデルで記述していく手法）とは別な、第3の手法としてモデル協調計算という手法を考え導入した。

## モデル協調計算

コンピュータ内に組み立てられた仮想世界の中で、複数の数値計算モデルを協調させて並列的に実行することにより、従来のトップダウン・アプローチで押さえきれなかった、より精密な現象過程や、ボトムアップ・アプローチで記述が難しかった、ある程度の自由度をもった現象のシミュレーションを計算で実現する方法とした。この、モデル協調計算は、理論、実験といった従来の科学技術の方法とは別な、第3の科学技術の方法である計算科学技術の位置づけを明確にする科学思想とした。

モデル協調計算を実現するための基礎技術は、オブジェクト指向によるシミュレーション、協調計算結果を実時間で3次元可視化する技術、異機種複数の計算機に機能分散を行ない1システムとして統合する技術を確立することにより基本的に可能であると考えられる。また、システムの設計開発にあたり、現象評価に係わるシナリオ、個別現象過程、そして、ソフトウェア・エンジニアリングについて広範な知識が必要であると判断された。個別の専門領域に限定してきた従来の技術者教育の今後の課題として考えられる。

### 3.7 オブジェクト指向のソフトウェア開発

地層処分ヴァーチャル・エンジニアリングは、放射性核種が生物圏に至らないよう隔離防護するための工業製品である人工バリアと天然の地層によるバリア効果をもって、2重に安全性を確保するしようとする設計思想について確証を得るための手段として、コンピュータ（計算）を利用する科学技術の手法であると考えられる。そのため、計画する処分システム（人工バリアと天然バリアから構成される、放射性核種の漏出・輸送防護機能を提供する系。従って、処分システムは処分場とは区別して考えるべきである）について、超長期的な、時間を含む地層環境中での振る舞いを予測評価することが性能評価の基本的な課題となる。性能評価においては、計画する処分システムに関する数10万年までの期間を含む超長期的な地層環境変動と多重バリアの振る舞いに関わる仮説を立て、それについて試験やシミュレーションを用いて検証を行い、総合的な予測の確証を得る手法が一般に取られてきている。

そのような性能評価作業では、各種の現象解明に関わる数理モデルやシミュレーション・コードの開発がなされている。開発された数理モデルやシミュレーション・コードは、一般に試験や観測でえられたデータから定式化されたものである。それは、設計論拠の決定や安全性評価等の、ある意味で汎用な意味合いをもつモデルであると考えられる。一方、研究作業においては、設計（安全解析を含む）と評価は基本的に異なると考えられるべきである。

設計作業においては、例えば処分場建設工事における水抜きや発破の地下水の水理への影響、吹き付けコンクリートが地下水に与える化学的特性、ロックボルトが岩盤に与える力学的特性等は許容できる範囲であるかといった、個別の設計課題に関わる安全性評価や論拠造りに、数理モデルやシミュレーション・コードが利用される。このような設計作業における仮説の検証や論拠造りの特徴は、評価（解析）しなければならない問題が比較的簡素化されていること、もしくは、簡素化できることがあげられる。なぜならば、前述のように、議論の対象となる工学的問題は、始めから有る程度限定された問題自身が明確な問題を対象として扱うことができるためである。

一方、性能評価は設計作業ではなく、環境アセスメントの領域であると考えられる。作業の目的は、人間が開発した処分システム（自然に対する開発行為と考える）が自然環境を変更し、最終的には生物の生存を脅かす可能性があるのかを評価することである。問題設定においては、処分システム全体としての地層環境中における振る舞いが問題視され、個別の現象過程からは予測しきれないシステム全体の振る舞いについてのアプローチが重要になる。従って、一般に性能評価は環境アセスメントと同等のアプローチとして、その初期の段階では枠組み（フレーム）とシナリオを開発することから始める。これは、また、環境開発として生じた地層中の処分システム全体が、環境に与える影響を超長期的に評価し、将来的に地球環境問題に発展しないことを確認する作業とも考えられる。

環境問題の基本は、人類の自然に対する働きかけ（開発行為）が、その量・質および時間等のすべての面において、自然のもつ許容量（もしくは社会システムの許容量）を超えてしまったことにより発生するものと考えられることにある。すなわち、自然のシステムが保有している環境維持許容量以上に、人類システムからの自然システムへの要求量が過大であるため、その未処処分が蓄積され、顕在化して行くことにより生じる問題と考えられる。

それは、本来、産業や地域開発の真の目的が、健康と安全の維持や確保等、人類生活の質を改善することであるはずのものが、実際にはその目的とはまったく逆の結果をもたらす場合といえる。このような地球環境問題を突起する可能性のある開発行為には、自然体系の構造と機能・多様性を考慮しつつ、かつ人間活動をも含めた相互依存システムの維持管理を、十分に検討・評価する活動が必要である。

こうした地球環境問題は、地球が長い年月をかけて作り上げてきた全体のバランスが壊れ、その結果として国境を越えた地球的なスケールで、その影響が現象化したものであり、それらのほとんどが不可逆的である。そのため、この問題の解決にはグローバルな国際協力を要し、さらに科学技術的対応以外に、経済社会活動や人間のライフスタイルのあり方といった根源的課題も含んでいる。

以上のように、性能評価を環境アセスメントの観点から捉えたとアプローチについて以下のように整理することができる。

- ・人類システムと自然システムの相互作用問題であるとして扱うこと。
- ・地層中に建設された処分場と多重バリア、そして自然体系という段階的プロセスに対する、総合的な枠組みが必要である。
- ・設定された枠組みに添った、多方面からの評価シナリオの開発が必要である。
- ・開発された評価シナリオを検証するための手段が必要である。  
手段として、野外調査、室内試験、コンピュータ利用が上げられる。

上記のアプローチにおいて、人類システムと自然システムの相互作用問題であるとして扱うと言うことは、地層処分性能評価問題が本質的には社会・経済的なリスクを含んだ人類の活動であることを示している。それは、妥当な予算で環境・社会に影響を与えず、放射性核種が消滅するまで安全に隔離処分することを要求する。

地層中に建設された処分場から漏出した核種移行に関わる段階的な地層環境プロセス評価の枠組みの作成は、現在のところ、地質学、地球物理・化学等の知見や、野外調査等により得られたデータから、人的活動として組み立てるものと考えられるが、現代の地球化学や進化生物学では、歴史が無作



為な発展として理解されてきている（繰り返さない）ので、地質情報等から過去のデータを取りだし未来に向けて安易に外挿することで説得力は得られないと考えられる。そのような仮定のもとに、どのように未来プロセスの枠組みを決めれば良いのかは、今後の重要な研究課題となるであろう。このようなアプローチにヴァーチャル・エンジニアリングが期待できるとしたら、それは、恐らくシステムに創発（Emergent）を期待することと同等であるとも考えられる。

評価シナリオの開発は、地層環境アセスメントの枠組が定まった後、それに従って、処分場より漏出する放射性核種の地層中移行に関わる事象について、系統的に説明する仮説を準備する作業である。この作業の複雑さは、予想される評価シナリオが、設計段階に考慮しなければならない事象も含む核種移行に関わる想定事象について、フィードバックを含んだ連結事象のループとなり、そのままでは計算可能性が失われることが考えられること（決定不能問題）、また、FEP等から構築された場合、シナリオそのものに心理的・主観的側面が含まれると思われ、それを除去することは当面困難と予想されることが上げられる。これらの問題に関する考えられる戦略として、第一に、自己再帰的なフィードバック・ループを持つネットワークの問題については、段階的プロセスを世代と考え、遺伝子アルゴリズム等を利用して最適化を計り、計算可能な問題に帰着させることが考えられる。また、FEP等に含まれる主観や心理作用の問題は、信念や世界観を持つ文脈依存のエージェント・モデル等で用いられる可能様相公理系等、数論や論理学に出発点を置いた系統だった評価を文脈に対して行い判断する手法等が考えられる。それらは、今後の重要な研究課題となるであろう。

シナリオを検証する手段には、野外調査、室内試験、コンピュータの利用等があげられる。これらのうち、野外調査や屋内試験は、前述の理由から、未来に対する系統だった予測評価に利用するには、十分な情報を提供することは難しいと考えられる。従って、コンピュータによる試行錯誤的な数値実験が期待できる総合的な評価手段である。地層処分ヴァーチャル・エンジニアリング・システムはそのような期待に基づいて開発するものであると考える。

以上のように性能評価作業をとらえてくると、それは、将来起こるであろう地層中を中心とした環境の変化と放射性核種の移行に関わる説明の理論モデルを組み立てる活動と考えることができる。また、作業にあたっては、設計検討の様に個別の現象について解明を行っていく方法をとることは難しく、処分システム全体の挙動として捉えて行くべきである。そのような問題の領域では、数理モデルは、始めから有る程度の自由度をもった連成形式となるであろう。それら連成モデルを試験等から導くことは非常に難しいであろう。なぜなら、一般に、試験はモデルを簡素化し、最も支配的な法則を見いだすことに目標が設定されるためである。また、そのような試験条件から定式化された数理モデルは一般に単純であり、試験にて示されるような理想化された環境化において観測される現象を、ある特定の条件の範囲でよく記述（近似）するものであるが、現象そのものに対する普遍的な記述を始めから除外されていると考えるべきである。

従って、目的とする連成現象を記述する複雑な数理モデル開発することは、解析的（理論）にも、試験的（実験）にも難しいと考えられる。そのような要求においては、試験は、それ自身の有する還元性から、より複雑なモデル開発に利用することは矛盾している。また、複雑な連成条件を示す非線形連立偏微分方程式を組み立てることは可能と思われるが、恐らくそれは解析解を持たないであろうし、離散近似する場合も多く困難を有するであろう。さらに、例え有る程度の条件の基に離散近似を行って、計算可能性（アルゴリズム）を見いだしても、それを具体的な超並列コンピュータやマシ

ン・パラレルのメッセージ交換アルゴリズム等にプログラムをするための、多くの知識・技術が必要となるであろう。そのように考えると、これ迄に示してきた性能評価のアルゴリズム開発戦略は、従来のアプローチでは非現実的であると考えられる。そこで、それらの従来型アプローチとは根本から異なった戦略が、このような中規模の複雑度をもつ問題に必要となる。そのアプローチの最も基本的な考え方は、オブジェクト指向によるモデル開発とシミュレーション手法である。

### 3. 7. 1 新たな研究開発の戦略としてのオブジェクト指向

近年、オブジェクト指向プログラミング言語の研究が進み、いくつかの応用分野が考えられるようになった。たとえば、システムのプロトタイピングや知識ベース・システムなどは、今後の有望な応用分野として期待されている。これらの分野に共通な特徴は、考えている問題やその解決プロセスの表現が、オブジェクトとその間のメッセージパッシング技術によって自然に表現できる、ということである。ここでは、オブジェクト指向ソフトウェア開発戦略を物理化学等のシミュレーションに導入する場合について基礎的な技術を取りまとめ、地層処分ヴァーチャル・エンジニアリング・シミュレーションのプログラム開発手法として位置付ける。オブジェクト指向を物理化学等のシミュレーションに導入することにより、以下のような事項を期待できる。

- ・オブジェクト指向での表現が適していると思われるニュートン力学系は、物理学の最も基本的な分野であり、また、評価に於ては掘削や地震影響等比較的、非専門家も関心を示しやすい議論の対象となる。
- ・力学系内の個々の対象（坑道、廃棄体、亀裂、等）をオブジェクトで、また、対象間の力の及ぼし合いをメッセージパッシングによって表現することが、比較的自然に行なえる。
- ・熟練者が問題を解くときによく行なうような、いくつかの対象をまとめて1つの仮想的な対象とみなすという思考方法が、その仮想的な対象を新しいオブジェクトとして作り出すことによって、自然な形でシミュレートできる。この考え方は、モデル開発の領域では達成問題として議論され、計算機科学の領域からは、並列に協調して独立したタスクとして動作するオブジェクト（即ちエージェント）のプログラミング・アルゴリズムとして扱われる。

#### (1) オブジェクト指向モデル表現

例えば、オブジェクト指向型表現が、力学の問題の構造を自然に表現できるならば、学習者が自分で問題を作り、システムにその問題を解かせることによって問題を理解する（もしくは観察する）ことが比較的自然的に行なえるであろう。そのような機能は、地層処分ヴァーチャル・エンジニアリングのシミュレーション機能開発にとって重要な設計開発思想である。なぜなら、学習者の考える問題表現とシステム内部でのオブジェクト指向的表現が、ある程度似たものとみなせるからである。そこで、将来の地層処分ヴァーチャル・エンジニアリング・システムは、問題解決システムに問題作成のためのインタフェースを統合した、問題理解のためのシステムとなることが予想できる。

#### (2) オブジェクト指向方法論

オブジェクト指向は、一つの方法論である。それはソフトウェア開発における一つの方法論である

と考えることができると考えられる。従来型のFORTRAN言語に代表される手続き型プログラミングでは、計算はその順序が決められていた。オブジェクト指向では、シミュレーション対象モデルを変数と式の組み合わせで記述し、それが外界からのメッセージを受け取ると内部にある式に従って、変数を変化させることにより、モデルの状態を変化させるプログラミング手法を考えることができる。そのようなモデルとプログラミングの手法は、次の点が従来のメインフレームに代表される手続き型プログラミングと相違する特徴と考えられる。従来、一つのプログラムとしてまとめ上げられたプログラムは、オブジェクト指向の考え方では、個々に独立してメモリー上にロードされ動作すると考えた。それらがメッセージを交換しながらシミュレーションを実行していく処理は、プログラミング上自然に並列的になるので並列処理や分散処理には適していると考えられる。また、このようなシミュレーションの内部処理技術は、シナリオに沿って複雑な連成問題を計算していく地層処分ヴァーチヤル・エンジニアリング・システムの設計思想に沿ったものである。

### (3) オブジェクト・シミュレーション

オブジェクト・シミュレーションは、計算機科学を基盤として生じてきた新たなシミュレーションの考え方である。ソフトウェア開発にあたっては、従来型のシミュレーションプログラムの開発手法とは異なった技術、手法が基盤となるため、計算科学技術の基盤研究開発課題として重要である。オブジェクト・シミュレーションにおいては、実際に複数のプロセッサやマシン環境で実行する場合の実行プロセスの同期制御等が重要課題となると考えられる。それは、メモリー上にロードされた個々の手続きは、メッセージを交換しあって状態を変える。メッセージはテキスト形式のファイルとして受渡しされることを原則とする。状態とは、メッセージに応じて変化するメモリーの内容であると考えられる。一つのモデル（処理単位）は、状態量を格納する領域と、それを決定する操作からなるとする。ここに言う“操作”とは、従来の手続き型プログラミングにいう命令の集合を決定する手法として扱い、状態量を格納するメモリーと“組”にして考えるものとした。

開発するシステムのオブジェクト・シミュレーション機能においては、シミュレーション条件を組み立てる為に、物理現象を個々の単純な個別現象シミュレーション・モデルの組に分解し、単純化した個別モデルについて、その振る舞いを決定する計算条件を付加し、それらに関連付けて実行することが望ましいと考えられる。そのためには、オブジェクト指向モデルに関するより詳細な研究が必要である。以上のように、物理モデルを示す支配方程式からプログラミングの為に離散方程式を導き、それをもとに一連の計算をプログラム言語に書き下ろすと言う、従来用いられてきた手続き型プログラミングの方法と異なったモデルと解法のアプローチとして、オブジェクト指向をとらえることができる。また、このオブジェクト指向によるソフトウェア開発手法は、これからの計算科学技術の開発課題である“マルチエージェント協調計算手法を応用した科学技術シミュレーション”研究の基礎的な技術を提供する。オブジェクト指向によるシミュレーションと、従来型の有限要素法や差分法によるシミュレーションは基本的に異なった思想に基づくものである。誤解のないよう下記の通りに区別する。

#### ・シミュレーション（決定論）

シミュレーションは、実験等で観測された物理化学現象等を記述する支配方程式を導き、それを基に計算コードとしてプログラムしたものを利用する。これは、観測された事実を計算式（数理モデル）に置き換えたものを利用して数値的に再現することを目指している。数理モデルから出発する。

#### ・数値実験（オブジェクト指向論）

オブジェクト指向のシミュレーションは、保存則等の場の方程式から出発するものではない。それは、分解された対象、個々について外部からの情報の交換と、それにより状態の変化を記述する操作からなる。従って、シミュレーションはより実験的である。オブジェクト・シミュレーションを従来のシミュレーションと分離して数値実験と呼称すると良いかもしれない。いずれにしても、オブジェクト・シミュレーションでは、シミュレーションを行うことにより結果を観察し、定式化する為の参考とする戦略をとる。オブジェクト・シミュレーションが、一般のハードウェア試験に比べ有利な点は、シミュレーションを構築する系が、素過程であり最終的には0と1のデジタル信号に完全に分解できることである。一般の実験はそのように分割可能ではなく複雑さを残す。

#### （4）新しいパラダイムの必要性

不幸なことに、良く知られているナビエ・ストークス方程式は、解析的には（つまり微分積分学の公式を使ってその解を数式として求めても）めったに解かれ得ず、コンピュータの力を借りた数値的な解析方法（数値計算）によらなければならない。ここに、次のような伝統的な方法論ではとても解決できないように思われる問題が生じる。

コンピュータで二次元問題（それは空間の二次元でも、空間の一次元と時間の一次元でもよい）を扱うことは比較的普及している。計算量もPCやワークステーションで殆どの問題は充分扱えるであろう。しかし、三次元空間での定常流や平面内の時間依存性のある流れのような流れの解のひとつを得るためには、大規模な計算（超並列計算機等のスーパーコンピュータ上で数時間あるいは数日間かかる）が必要となるであろう。さらに、三次元空間内の時間依存性のある流れのような四次元流を解くのは、膨大な計算量（数週間、あるいは数ヵ月かかるかも知れない）となるであろう。このように、問題の次元とともに計算所用時間が急速に増大する理由は、数値解析と科学的なデータ表現の本質に根ざしている。

例えば、一次元空間に沿って、すなわち線上に分布したデータの計算のみに限って考察する。その直線を点で区切って有限の格子に分け、各々にひとつの数値を割り当てる。仮に10個の格子点があるとしよう。このような一次元問題は10個の数値を必要とする。二次元問題なら二次元格子を必要とするので、 $10 \times 10$ すなわち100個の数値を必要とする。三次元問題は $10 \times 10 \times 10$ で1000個の数値を必要とし、四次元問題では一万個となる。もし、1km立方の地層中を流れる地下水を1mメッシュで表現し、計算しようと思えば、空間的に $1000 \times 1000 \times 1000$ 、そして時間的には1年毎に1000年の計算をしようとしたら1000個の格子を必要とするだろう。この時、3つのパラメータ（x方向に地下水速さ、y方向に地下水速さ、z方向に地下水速さ）の各々が、10通りの値をとって変化するとしたら $10$ の $12$ 乗すなわち一兆個のデータ点を計算する必要がある。その場合、一次元問題を解くのにスーパーコンピュータで10秒かかるとすると二次元問題を解くには1000秒かかり、三次元問題を解くには16分と40秒、そして

四次元問題を解くには約28時間かかるであろう。

以上のようにして、3つの重要なパラメータに地下水の流れがどのように依存するかを調べるのに要求される1000個の四次元問題を、この方法で解くには約1年かかるであろう。その結果、膨大な量の数値が計算結果として得られるだろう。そしてそれは、「なぜ」については、いかなる理解も与えてくれないであろう。このように計算することのみに主眼を置いた方法論を、地層処分ヴァーチャル・エンジニアリング・システムの開発に持ち込むことは、費用対効果、現実性を考えると危険であり、評価という観点からも妥当でないと考えられる。

なぜならば、問題が、工業製品の設計を確認することであるならば、それが機能することがわかればよいので、「なぜ」に気を配る必要はない。同じことが、前述した処分場の設計にあてはまる。処分場設計では、機能することを確認するために安全性評価をおこなうが、環境アセスメントとしての性能評価では「なぜ」に答えることが要求される。このように機能確認の為の計算は、科学的洞察を得る上では役立たないであろう。それに答える何か新しいパラダイムが必要である。そのためには、地下水の現実的な物理とはまったく無関係な、非常に一般的な疑問を投げかけることから始めることが大切であると思われる。これは、流体力学等における伝統的な研究気質に対立するものと思われるが、数学的、科学的には意義があるアプローチと考えられる。

はじめに問題の一般的な本質を理解し、それから攻略計画を立てるために、それを理解することに努力すべきである。それには、他分野からの「技術移転」も重要である。物理についての仮定が特殊なものではないほど、その結果はより広範囲に応用可能なものになるだろう。オブジェクト指向は、そのような観点からのパラダイムである。同様に系の対称性は何か。その対称性をもつ系において可能なパターンカタログは何か。これら可能なパターンが実際に起こるかどうかを決定するためには何を計算しなければならないか。そしてその安定性はどうか。これら一般的な疑問を整理してから計算をすることにより、計算量を減らしシミュレーションをより現実的な作業に組み込むことができると思われる。そのように、シミュレーションに置いては、唯、超高速コンピュータを利用して大量の計算結果を吐き出す数理モデルを開発することより、むしろ、問題の本質的な部分を整理してモデル化にあたるべきである。

例えば、前述のモデル対称性の例では、流れのパターンについて、より多くの知識を整理してデータベース、知識ベース等の情報ベース化することにより、計算そのものを簡素化するだけでなく、この系以外の同じ対称性をもつ他の任意の系について、数学的な特徴の幅広い理解を得るであろう。例えば、多孔質媒体中の浸透流れ解析については、微小な領域から広域的領域まで共通した対称性流れの考え方を見い出して応用できるかも知れない。そのようにして、可能な限りモデルに独立な概念を使用し、最後にモデルの詳細な物理の研究を行うべきである。例えば、地下水流れの対称性の議論は、可能なパターンの一般的な範囲を選択することを示すであろう。そして物理は、どのパターンが実際に起こるかを教えてくれる。

本項で主張していることは、“知りたいものによっては別の有効な戦略が存在する”ということであり、かつ異なった戦略は、異なったタイプの問題に対して有効であるということである。オブジェクト指向がそのような戦略としてシステム開発に期待できるとともに、モデル化においても、対称性の整理が、より現実的な数理モデルを開発する一つの戦略であることも合わせて示した。このような科学に対するパラダイムは、地層処分性能評価シミュレーション・システムのような、統合化された複雑系

として振る舞うアルゴリズムを開発する場合には特に重要であると考えられる。

#### (5) オブジェクト指向ソフトウェア開発

前章までに、オブジェクト指向プログラミング (OOP: Object-Oriented Programming) とは、ソフトウェア・システムの設計、及び実現のための比較的新しい方法論であることをみてきた。その主要目標は、ソフトウェアの拡張可能性と再利用可能性を改善してプログラミングの効率を向上し、ソフトウェア開発の費用対効果比を向上させることである。また、オブジェクト指向プログラミングにより、ソフトウェア開発の設計面と開発面の結び付きを一層深めることができると考えられる。さらに、物理化学等の問題について新しいモデル化手法の開発の可能性があると考えられる。

ソフトウェアの設計および実現の方法は、実に様々であるが、このオブジェクト指向開発法は成功の可能性、将来性の面でかなり期待をしてよい開発法であると思われる。個々の応用分野において、オブジェクト指向プログラミングの採用により成功したとの報告はあるが、地層処分関連のソフトウェア設計開発分野では、オブジェクト指向方法論による成功例はまだ報告されていないようである。

#### 3. 7. 2 クラス階層のプログラム構造

オブジェクト指向ソフトウェアの構造は、そのソフトウェアの動作を特徴づけるクラスの集合から構成される。各クラスのオブジェクトは、クラスのメソッドを呼び出すことにより実行を行う。これは、オブジェクトへメッセージを送る操作で実現する。従って、メッセージがオブジェクトに対して行われる操作を表現することになる。オブジェクト指向プログラミングは、操作する手続きよりも操作されるデータに重点をおいている。データはソフトウェア分割の基本となる考え方である。これは、クラス概念が拡張されたデータ型であることが背景となっている為である。実際、オブジェクト指向ソフトウェア設計では、ソフトウェアを構造化データの型、つまり基本クラスとサブクラスに分解し、それら基本クラスとサブクラスそれぞれの動作を定義することがソフト開発の中心となる。また、オブジェクト (つまりクラス (サブクラス)) 変数は、対象としている問題の物理的要素、または、論理的条件を特徴付ける要素 (例えば物理変数) として扱う。従って、オブジェクト指向で物理工学上の問題を扱う場合、対称となるモデルを記述するクラス階層モデルを設計することが研究課題の中心となるであろう。

オブジェクト指向ソフトウェアの概念を定義し、そのシステム設計を行う段階においては、クラス集合 (サブクラスを含む)、及び、その定義とオブジェクトのみを定義することが一般的である。各クラスの動作はメソッド・インターフェイスで決定する。しかし、メソッドの実現上の詳細は、システム設計には含めない。通常のオブジェクト指向言語では、クラスで定義されるメソッドのインターフェイスと、その実現の詳細は独立に定義できる。このことにより、システム設計とその実現を分離できることになる。これは、オブジェクト指向方法論に於ける重要な概念であるが、前述の通り性能評価などに於ては、従来蓄積してきたFORTRANコード等とのインターフェイス設計技術としてメソッドの考え方を応用するには今後の研究が必要である。このように、概念 (メソッド・インターフェイスを伴うクラス定義) を、その実プログラムの生成 (クラスのデータ構造とアルゴリズムを実現するコードの生成) と切り離すことは、オブジェクト指向プログラミングの場合、また再利用可能性や保守コストの削減のために非常に重要である。オブジェクト指向プログラミングの概念は、メソッ

ド・インターフェイスの形で定義され、クラス内に閉じ込められるため、再利用可能性 (reusability) に優れている。ユーザーは、メソッド・インターフェイスで定義されたクラス・オブジェクトの動作さえ理解していれば十分であり、そのオブジェクトの実現の詳細について知る必要はない。メソッドの実現は、ユーザから見えない“ブラック・ボックス”に閉じ込められる。

さらに、データ構造やアルゴリズム (すなわちクラス実現のコード) の実現に変更があっても、そのクラス内のコードの部分だけ変更すればよいので、保守性 (maintainability) も優れている。その場合、クラス・インターフェイスに変更は生じないので、クラスの外は何ら影響を受けない。このインターフェイスは、クラスの外からクラス・オブジェクトを“利用する”ときの利用の形式を定めることになる。

### (1) ソフトウェア開発管理面からのオブジェクト指向ソフトウェア開発の狙い

基本クラスを用いて再利用を推進したり、サブクラスを用いる段階的开发法を活用して、ソフトウェア開発に要する時間とコストを削減することが可能と考えられる。従って、地層処分ヴァーチャル・エンジニアリング・システムのような、試行錯誤的技術開発を伴う大規模システム開発手法として有効と考えられる。何故ならば、ソフトウェアの部品化・標準化による再利用の推進にともなうシステム開発効率の向上が期待できるためである。

### (2) 性能評価シミュレーション・コード開発面からの課題

性能評価シミュレーション・コードの開発面から考えた場合、オブジェクト指向の方法論は多くの研究開発課題を含むものと考えられる。第一に、クラスという抽象化された情報のモデルに、従来のFORTRANに代表する数値モデル (手続きモデル) の開発手法が、どの程度適用可能であるかを評価しなければならない。従来の手続きモデルにおいては、式は変数と定数から表現されるものとして与えられ、プログラミングにおいては、変数は名称付きの参照可能なメモリーを示し、定数はコンパイラがプログラム内部の適当な番地に配置した数値データとして扱われた。従って、式は変数と定数を格納するメモリーに対する演算操作命令に展開され、操作結果は左辺の変数領域に格納されるという計算制御手法であった。これは、演算とデータ (変数、定数) が分離して考えられた手法である。

手続き的操作は、このようにプログラム内部に宣言されたデータ領域と、それを参照して計算を進めていく順序だった式の記述表現による算法である。これに対しオブジェクト指向の方法論では、操作とデータを組み合わせたクラスという概念に添って、プログラミング・アルゴリズムを考えなければならない。例えば、式のそのものをクラスとして考えることができる。関数  $F(x, y, z, t)$  は、ある物理量を与えるクラスとして考えることができる。ここに、 $x, y, z, t$  はインスタンス変数である。即ち外部からメッセージとして与えられる。この関数  $F()$  の内部処理は、プライベートなメソッドとして考えた場合、 $F()$  はメッセージ  $x, y, z, t$  により一意に定まる物理量として扱うことができる。このように、オブジェクト指向について、性能評価シミュレーション・コードを開発面から考えた場合、ソフトウェアの基盤から設計開発を行わなければならないことが言える。また、既存FORTRANコードとのインターフェイスを独自の技術開発課題となると予想される。

オブジェクト指向ソフトウェア開発に於ては、その初期における設計段階で全体構造を示すことができるため、開発プロジェクトの信頼性 (reliability) も優れている。その方法によれば、システムを

構成する主要部分がまず最初に形作られ、それから個々の部分が組み立てられていく。その主要部分は、抽象的な性質で定義される。従って、全体の統合テストを下位レベルの詳細仕様を定める前に行うことができる。そのような手法は、ソフトウェア開発手法の一つであるラピッド・プロトタイピングの一手法として好例である。システムを、主要なクラスとそのオブジェクトに分解してから、システムの動作を特徴づける重要なメソッドをコードとして組みこんでいくことにより、システムの各“部分”がどのように全体として動作するか判断することができる。従って、試作システムの開発にあたってこの方法を適用した。

### 3. 7. 3 オブジェクト指向による数理モデルのアプローチ

オブジェクト指向を用いて物理工学的モデルをシミュレートする場合、従来の解析的手法（支配方程式（基礎方程式）の導出、数値解析（離散方程式の導出）、プログラミングとは、かなり異なった手法（思考論理）が要求される。特に、重視しなければならない点は、オブジェクト指向によるモデル組立は、数理モデルの組立にまで影響しかねないと考えられることである。従来のシミュレーション・モデルは、自然界の現象を実験室等の理想化された環境で再現、観測し、観測結果から現象の振る舞いの方程式（以下、数理モデルと記す）を導き出す手法を一般的としていた。このようにして導かれた数理モデルは、数理モデル全体が対象とする現象を記述するものであり、基本的には分割可能ではない。つまり、数理モデルを、分割することにより数理モデルとしての意味を失う。

例えば、ある連立方程式が力学系モデルとして与えられた場合、モデル式を分割しても、個別に意味のあるアルゴリズムを引き出すことはできない。従来の実験と数値解析的手法は、解析的な数理モデル開発を目標としたため、得られた数理モデルは完成したモデルとして考えるべきである。このように、数理モデルを実験等から得られたデータをもとに解析的に導いて検証した場合、得られた数理モデルは、それを導いた現象過程の範囲で数値計算による初期値、境界値問題に解を与えることができる。それは、従来一般に取られてきた数理モデルの開発手法である。そのような手法の限界は、得られたモデルの記述する範囲が、理想化された系での観測（計測）を基準にした、いわば閉じた理想系でのモデルのため、自然界のように外部からの事象影響等を考慮した記述は一般になされないと言うことである。言い換えれば、得られた数理モデルが利用できる範囲は、当該モデル単体で計算して得られる数値の範囲であり、例え、複数の数理モデルが解析的に得られたとしても、それらは、個々に独立しており、相互の連成もしくは協調は記述されていないということである。

そのような数理モデルの開発は、設計要素の個々の確認には有効であって、システム全体の振る舞いを評価するために組み合わせて利用することは基本的にできないと考えられる。なぜならば、個々の数理モデルは、それ自身に連成条件が記述されていないためである。

オブジェクト指向を、将来の地層処分性能評価モデルの組立戦略とすることは、数理モデル開発に分割と再編成の可能性が期待できることが最大の魅力である。また、総合的な多方面からのシナリオに沿った性能評価シミュレーションと言う考え方は、恐らく分割と再編成、そしてメッセージ交換を伴う、並列的なオブジェクト指向の数理モデルを構築するめどが立たなければ、理想だけにとどまるであろう。幸い、本年度の研究では、破碎帯空壁を通過する核種に見立てた微小粒子のシミュレーションを試行することができた。このモデルは、非常に単純な微小粒子と、1本の運動力学を記述するメンバー関数からなっているが、動画化されたシミュレーションの見かけはリアルであ



る。恐らく、化学特性を示す各関数を与えることにより、破碎帯空壁中を浸透する、地下水の地球化学的性質の数値実験を行うことができるであろう。このような、素過程の上に組み立てられる数値実験を試行錯誤的に行うことにより、分配係数の利用に関わる議論等多くの性能評価に関わる問題研究を、より不確かさの無い条件の基で行うことができるであろう。

オブジェクト・シミュレーションを極端に単純化して考えた場合、例えば、移動や衝突を伴う対象をシミュレーションする手法は以下の通りとなる。

- 1：対象を構成する最も単純な最小要素を物理モデルとして考える。例えば、構造物はそれを構成する要素の組として考える。流体はそれを構成する分子等を代表的な粒子として考える。従って、流体は非常に単純な対象に分解できる。
- 2：対象とする物理モデルの運動について数理モデルを考える。対象が粒子であれば、移動や、摩擦、衝突等の扱いは単純なモデルとなる。このようにして、導いた数理モデルは外部から与えられたメッセージ（外力等）にตอบสนองして物理モデルの位置等決定する関数と考える。
- 3：対象となる物理モデルは、それを特徴付ける変量（インスタンス）と状態を決定する関数からなるプログラム・アルゴリズム（クラス）として記述する。
- 4：クラスに対して固有の現在座標等のメッセージを送り込むことにより、実行プロセス（オブジェクト）を必要な数だけ産み出す（C++ではインスタンス化と言う）
- 5：各オブジェクトは、互いにメッセージを交換し、時間と共に摩擦、衝突等の状態を確認し状態を変える。その状態を視覚化して観察することにより、リアルなシミュレーション結果が得られる。

ただし、その様にして得られたシミュレーションは、場の状態や釣り合い等を記述する方程式から出発していない。従って、例えば、流れの問題を扱うことを考えた場合、従来のナビエ・ストークス方程式等の場の方程式から出発する有限要素法や有限差分法とは本質的に異なった考え方である。従来の手法が、どちらかと言うと観測された事実を数値的に再現するための定式化を目指していたことに比べ、オブジェクト指向は数値的に実験を行う手法ととらえることがより適切と考えられる。

### 3. 8 並列計算技術

地層処分ヴァーチャル・エンジニアリング・システムのシミュレーション機能について、一部並列計算技術を応用することにより計算速度の向上等を図り、より大規模かつ詳細なシミュレーションを可能とすることを目的とした研究を行った。現在、並列計算技術については、各種のハードウェア・アーキテクチャやコンパイラ等が開発されているが、メーカー固有の技術に左右される課題も少なくない。例えば、データ・パラレルに代表される超並列計算技術については、ハードウェアメーカー各社で独自の技術開発が行われており、仕様の統一がなされていない事が現状である。また、ネットワークにて接続された複数の異機種コンピュータを利用して並列に計算処理をすすめる、ネットワーク・パラレル技術の開発も進められている。この方式によると並列計算環境を特別なハードウェアに

限定しないで構築することが可能となる。ネットワーク並列計算技術は、1990年代前半の初期の段階では各種の手法が提案されていたが、メーカー、大学、研究所が協力して仕様の標準化が開始されたところである。

### 3. 8. 1 計算量の推定

並列コンピュータについて理論的、及び知的な関心が寄せられていることは別として、この分野の研究を推し進める推進力となる以下の2つの要素が考えられる。

その1：高性能コンピュータへのとどまることを知らない需要の増大

その2：期待される高性能コンピュータの開発技術を制限してしまう物理的な障害

地層処分ヴァーチャル・エンジニアリング・システムで必要とされる、多くの処理操作には高い計算能力を必要とする。特に、地下水流れのシミュレーション、化学的反応のモデル化、処理場周辺の岩盤力学モデル、気候や地質学上の予測などのモデル開発には、高い計算能力が必要とされると考えられる。また、それらのアプリケーションの実行を支えるものとして、データベース・システムが考えられ、それは、入出力性能についても高い能力を必要とするであろう。更に、シミュレーションの視覚化や、仮想世界の構築の必要性からも、より高い性能のコンピュータが求められるであろう。以上のように、地層処分ヴァーチャル・エンジニアリング・システムを実現するための、コンピュータ・ハードウェアに求められることは、第一に非常に高速な計算と入出力処理能力であると考えられる。

ハードウェア面から見れば、従来のコンピュータ機器の性能は、処理速度がどれだけのものであるかによって判断されてきた。従来、単一のプロセッサに基づくアーキテクチャが、その性能を飛躍的に高めてきたが、ハードウェアの開発も現在の技術では物理的な限界にきていると言われている。例えば、最近のVLSIチップのライン幅は約0.5マイクロメートルで、それは写真食刻（エッチング）技術で使用する波長（数千オングストローム）に近い数値である。高性能化のためには、信号線を密接に接続することによって、相互誘導効果が生じ、回路の操作速度を早めるために現在広く使われている小型化の技術がもはや妥当なものではなくなってきている。小型化の技術が妥当なものではなくなるのがいつになるのかについては、いろいろな予測があるが、多くの技術者は、それは西暦2010年前後であろうと予測している。従って、地層処分ヴァーチャル・エンジニアリング・システム・プロジェクトでは、現在のハードウェア・アーキテクチャを基盤とした研究開発を継続していくことができると考えられる。

並列処理では、複数のプロセッサの計算能力を組み合わせることによって、全体の処理速度が高まることを狙ってシステム開発を行う。この処理能力によって、シミュレーションがより迅速に行われるだけでなく、より複雑な解析が可能になり、より高度なシミュレーションを行うことも出来るようになることが期待されている。また、プロセッサの単体速度に加えて、プロセッサ、主記憶装置、及びディスク間をデータが転送される速度に、より多くの処理が制約を受ける。それについても、並列入出力アーキテクチャを使用することによって、制約を受けずに、機器の大きさに応じて性能を段々高めていくことも可能になると考えられる。特にデータベース・システムや視覚化処理の性能向上では、効果が期待されている。

## (1) 計算量

地層処分ヴァーチャル・エンジニアリング・システムのシミュレーションに要求される演算速度を見積ることは、モデルの正確な構造に関する知見なくしては難しい考えられるが、その概要を把握することは大切である。ここでは、シミュレーション領域を3次元の格子として考え、計算量に関わる凡その情報を把握する。最も単純な計算として、3次元立方格子でモデル化された計算領域を差分法に代表する隣接格子計算で考える。各格子における汚染レベルを取り囲む格子の汚染レベルをもとに決めるといふものである。隣接格子計算においては、あるひとつの格子で定まる物理量は、それに最も近い周囲の格子における物理量によって決まると考える。対角上の格子も数えると、3次元空間ではひとつの格子に26の「隣接要素」が存在する。従って、各格子について、計算は27回（各隣接格子についての計算が26回、それに前回の反復のあとの当該格子の計算が1回）行う必要がある。シミュレーションが、反復のたびに10個の変数を更新し、それぞれの更新に10回の浮動小数点演算が必要であると仮定すると、下記の量の計算が行われることになる。

$$27 \times 10 \text{ (変数/反復} \cdot \text{格子)} \times 10 \text{ (FLOP/変数)} = 2700 \text{ (FLOP/反復} \cdot \text{格子)}$$

また、滑らかな視覚化シミュレーションのためには、毎秒24フレームの割合で100万個の格子すべてについてシミュレーションを実行する必要がある。その場合、以下の計算量が単位時間（秒）あたりで必要である。

$$2700 \text{ (FLOP/反復リフシュ)} \times 24 \text{ (反復/秒)} \times 1 \times 1,000,000 = 65 \text{ GFLOPS}$$

この65GFLOPSという計算速度は、1秒間に10の9乗回の浮動小数点演算を行う速度（ギガ）を単位としたものである。この65ギガフロップスは、現在各国で利用されている汎用の超並列コンピュータの最高速度に近いものである。この数字は、10個の未知数について（100, 100, 100）の格子点を持つ3次元シミュレーションを実時間で動画再現する為に必要な計算速度を表している。

## (2) メモリー容量

メモリー容量は、それぞれの変数が4バイト（単精度）のメモリーを占有すると仮定すると、シミュレーションの1回の反復に必要なメモリー容量は下記のとおりとなる。

$$1 \times 1,000,000 \times 10 \text{ (変数/格子)} \times 4 \text{ (バイト/変数)} = 40 \text{ Mバイト}$$

上記は変数の数が10と、比較的少なく見積っている。従って、実際のプログラム開発に於ては、その2倍から3倍見込むことが適当と思われる。従って、このようなプログラムは約100MB程度のメモリーを要求すると考えれば良いと思われる。また、各サイクルの計算が、たとえその場でできるとしても（連続して行われる個々の反復の結果は同じメモリー位置に保存される）、中間的な結果の記憶のための空間は、やはり割り当てておく必要がある。それに、20%のオーバーヘッドを見込んだデータ空間の規模は100~150Mバイトと見積られる。

さらに、計算がその場でできない場合は、データ領域は反復2回分を記憶するのに十分な容量を備

えていなければならない。この最悪のケースでは、データ空間は最高200～300Mバイトの容量を備えている必要がある。それでも、物理モデルが2km立方の場合、1格子の大きさは20m立方である。廃棄体は緩衝材直径で約3mなので精度の面で充分ではない。

### (3) データ転送速度

この項では、シミュレーション機能と視覚化機能の間のデータ転送速度の概算値を導き出す。シミュレーション域全体を見る場合、半透明立方体視覚化モデルでは、反復のたびにシミュレーション・データ空間の内容すべてが視覚化機能に送られる必要がある。シミュレーション・データ空間の大きさは、144～480Mバイトと見積られる。毎秒24回の反復数でデータ空間全体を転送するためには、毎秒3.5～11.5Gバイトのデータ転送速度が必要となる。シミュレーション空間の一部だけを表示する場合は、データ転送速度の見積り値もそれに応じて下がることになる。

### (4) 視覚化機能

例えば、半透明立方体視覚化モデル（ボリュームレンダリング）では、ディスプレイ用に選ばれた変数をデータ空間から抽出し、データを圧縮したうえで、ディスプレイ準備機能に、そのデータを転送する操作を視覚化機能によって行う。計算量の見積り用として、ここでは極めて単純な形式のデータ圧縮を想定する。すなわち、データ圧縮は、シミュレーションの最終回の反復において数値が変化した変数だけを転送することによって行うと考える。毎秒24回の反復数で実行されるシミュレーションで、3つの変数を同時に表示する場合には、視覚化機能はデータ領域から最大で以下の変数を抽出する必要がある。

$$3 \text{ (変数/格子・反復)} * 1 * 1,000,000 * 24 \text{ (反復/秒)} = 72,000,000.0 \text{ (変数/秒)}$$

抽出後、それぞれの変数の数値は、その前の回の数値と比較されディスプレイ準備機能に出力される。データの比較と出力に3回の浮動小数点演算を見込んだ場合、視覚化機能の所要演算速度は以下のように見積ることができる。

$$72,000,000.0 \text{ (変数/秒)} * 3 \text{ (FLOP/変数)} = 22 \text{ MFLOPS}$$

視覚化機能は、抽出データを圧縮した後、ディスプレイ準備機能へと転送しなければならない。この時、データ圧縮が90%の効果を持つとしたら、3つの変数のディスプレイに必要となるデータ転送速度は以下のようになる。

$$1,000,000.0 * 4 \text{ (バイト/変数)} * 3 \text{ (変数/格子・反復)} * 0.1 = 1.2 \text{ (格子・反復)}$$

毎秒24回の反復数で実行されるシミュレーションの場合、視覚化機能とディスプレイ準備機能の間の通信チャンネルの所要帯域幅は次のようになる。

$$1.200,000.0 \text{ (バイト/反復)} * 24 \text{ (反復/秒)} = 29 \text{ (メッシュ/秒)}$$

次に、ディスプレイ準備機能は、シミュレーション区域の3次元半透明画像の図形表現を行い、それによって得られた画像データを圧縮し、ディスプレイ機能へと転送する。

図形表現のためには、視線にある全ての格子の整数加重合計を、画像のそれぞれのピクセルについて計算する必要がある。この計算は、赤、緑、青の3原色のそれぞれについて繰り返されなければならない。この計算には、各格子毎に5回の整数演算が必要であるとすると、シミュレーションの反復1回毎に必要なとされる整数演算の回数は次のようになる。

$$\begin{aligned} & 800*1000*100\text{ピクセル}*3\text{ (変数/反復)} *5\text{ (IOP/変数・ピクセル)} \\ & =1.2*10^9\text{乗 (IOP/反復)} \end{aligned}$$

毎秒24回の反復数で実行されるシミュレーションの場合、最悪のケースを想定したディスプレイ準備機能の所要演算速度は次のように見積ることができる。

$$1.2*10^9\text{乗 (IOP/反復)} *24\text{ (反復/秒)} =28,800. \text{MIOPS}$$

この数は、シミュレーションの1回の反復の間に、画像中のすべてのピクセルを計算し直すのに必要な演算の数を示すものである。但し、画像全体の計算のやり直しが必要となるのは画像を回転させる場合だけである。殆どのシミュレーションのシナリオでは、表示されたピクセルのごく一部を計算し直すだけですむ。そこで、画像更新速度が画像の回転中に遅くなるのを認めることにすれば、通常のシミュレーションのシナリオにおける所要演算速度の見積りを行うことができる。例えば、1回の反復の間に表示されているピクセルのうちの5%が変わるものとする、シミュレーションの1回の反復毎に必要なとされる整数演算の回数は次のようになる。

$$\begin{aligned} & 40*1000*100\text{ピクセル}*3\text{ (変数/反復)} *5\text{ (IOP/変数・ピクセル)} \\ & =60*10^6\text{乗 (IOP/反復)} \end{aligned}$$

そこで、毎秒24回の反復数で実行されるシミュレーションで、ディスプレイ準備機能に要求される演算速度は次のように見積ることができる。

$$60*1,000,000.0\text{ (IOP/反復)} *24\text{ (反復/秒)} =1,440 \text{ MIOPS}$$

図形表現の完了後、ディスプレイ準備機能はディスプレイ機能にデータを転送しなければならない。表現された図形データは、表示ピクセルひとつが24ビットの整数データ要素ひとつによって構成されている。この24ビットの整数データは3つの8ビット・バイトに分かれ、それぞれのバイトが各ピクセルにおける、赤、緑、青のそれぞれの色の濃淡を表現している。画像が800×1000ピクセルを占めると仮定すると以下のデータの転送が必要となる。

$$800*1000\text{ピクセル}*3\text{ (バイト/ピクセル・反復)} =2.4*1,000,000. \text{(バイト/反復)}$$

毎秒24回の反復数で実行されるシミュレーションの場合、最大負荷のケース（データ圧縮が一切行

われない)を想定したデータ転送速度は次のようになる。

$$2.4 * 1,000,000. (\text{バイト/反復}) * 24 (\text{反復/秒}) = 58 (\text{Mバイト/秒})$$

データ圧縮アルゴリズムが90%の効果を持つ場合には、所要データ転送速度は毎秒5.8Mバイトに下がる。半透明立方体視覚化モデルには1280×1024ピクセルのラスターディスプレイが必要である。

#### (4) 基本データ処理量

以上のように、3次元にて100,100,100格子の分割された計算モデルを差分法等でシミュレーションする場合、ハードウェアに対する凡そのデータ処理速度の要求は以下のとおりである。これらの数値は、地層処分性能評価シミュレーション・システム・ハードウェア要求性能のある程度の目安である。

計算速度 : 70 GFLOPS以上  
メモリー容量 : 200MB以上  
入出力速度 : 28MB/sec  
画像処理速度 : 60MByte/sec

### 3. 8. 2 並列計算技術

大規模シミュレーション技術開発として良く知られたグランドチャレンジ (1992年合衆国高性能計算通信プログラム) は、その目標を達成するのに必要なアプリケーションの内容、処理速度、記憶容量、そして入出力を次のように明示している。近い将来、テラフロップの処理速度、1テラバイトの主記憶容量、及び1テラバイトの入出力性能のコンピュータが、流体力学、海洋の循環、気候のモデル化等の問題解決に利用されるであろう。例えば、72時間の天気予報では、およそ10Gフロップの処理速度が必要とされている。現行のスーパーコンピュータの処理速度は、グランドチャレンジのアプリケーションに必要な処理速度の100倍から10,000倍も遅いものとなっている。

コンピュータ・ハードウェアは常に革新されている。市場は主要メーカーによる接收、合併など最近いくつかの大変革を経験し、業務内容も大きく様変わりしている。ここ数年間、従来の伝統的な“メインフレーム”設備から、より小さい“事業レベル”のサーバーシステムに移り変わってきている。この傾向は、すぐに入手可能なCPU技術と迅速なネットワーク機能を有する生活必需品として、小型化した多機能分散処理システムへの変遷ともいえる。従って、コンピュータ・ハードウェアの技術革新について次のことがあげられる。これらは多年度に亘る、ハードウェアとしての地層処分ヴァーチャル・エンジニアリング・システムの構築プロジェクトを推進するうえで重要な考慮事項である。

考慮事項1 : モアの法則をマイクロプロセッサ技術に継続して適用可能であること。30年前にインテルの共同創設者である、ゴードンモアによって次のように述べられている。

- \* RISCとCISCのチップトランジスターの記録密度は18カ月ごとに倍増し続ける。
- \* モアの法則は沈静化の兆しがない。

これに対する、否定的な見解は以下のとおりである。

- \* RISCとCISCのユニットの容量が大きすぎるので、特注品の超高性能プロセッサは、価格及び性能の競争において敗北してしまった。これは、ベクトル系の“メインフ

レーム” 機器の衰退を示している。

**考慮事項 2 :** 超並列コンピュータ (MPP) が大規模に市販されるようにはなっていない。MPPシステムはプログラム化することが困難で、より進んだソフトウェア技術に遅れをとっていることは、超高性能プロセッサ全体の市場では、すでに失敗していることを意味している。

次に高性能コンピューターの多くの実用的なアプリケーションを特色づける2つの重要な特性がある。性能を問題にする際、これから述べる2つの特性を必ず考慮にいれるべきである。

**リアルタイム特性 :** 多くのアプリケーションにおいて、計算結果が有効である、限られた時間内に計算がおこなわれなければならない。

**精度に依存した特性 :** より高い精度を得るためには、できるだけたくさんの問題を処理しなければならない。

精度に依存した特性は、リアルタイム特性と組み合わせられて言及されることが多い。例えば、同じ計算量の問題があるとすれば、より高い精度を得るためには、より短い時間で処理されなければならないし、たくさん問題があるとすれば、それぞれ同じ位の時間、もしくはより短い時間で処理されなければならない。このことは、単位時間内での計算速度向上パラダイムとして知られており、その主な目的は、必ずしも時間を節約することではなく、一定の実行時間内に、より高い精度で問題を処理することである。このために計算能力を高める必要性が生じ、コンピュータ自体の性能を高める必要性も生じる。

多くのアプリケーションは、外部 (すなわち環境) の物理的プロセスをシミュレーションすることに基づいている。シミュレーションの質がいいものかどうかは、モデルが外部 (環境) を物理的にいかに詳細にわたって記述しているかにかかっている。これらの物理的プロセスは、3次元空間と時間で (つまり4次的に) おこなわれている。

### (1) 並列ハードウェア

並列コンピュータは、凡そ20年間出回ってきているが、学界や研究センターのような場所でのみ使われていることが現状と思われる。現在使われている並列コンピュータは、主に次の2つのタイプに分けられる。

**分配型メモリーシステム :** 超並列並列プロセッサ (MPP) やワークステーションクラスターを含む。

**共用メモリーシステム :** 対称型マルチプロセッサ (SMP) や並列ベクトルプロセッサ (PVP) を含む。

MPPシステムは、何百個ものあるいは何千個もの安価なRISC、もしくはCISCプロセッサからなり、迅速な特注品のインターコネクトと接続されている。MPPの代表的なシステムは、Cray T3D/T3Eシリーズ、日立SR2201、インテルパラゴンシリーズや富士通VPPシリーズなどである。富士通VPPシリーズは若干タイプが異なり、超大規模プロセッサというよりは、むしろベクトル系のMPPのシステムであるといえる。多くの場合、MPPシステムは、迅速なネットワークを有した、大規模な同機種ワークステーションクラスターとみなされる。それとは反対に、産業界や大学研究機関などで頻繁に使

われるようになってきているワークステーションやPCのネットワークは、対応の遅いタイプのMPPと考えることができる。実際、典型的なデスクトップシステムのコンピュータ機器の方が、典型的なMPPノードよりも高い処理能力を持っていることも事実である。

迅速なイサネット、FDDI、ISDN、ATMスイッチングなどのネットワーク技術のめざましい発展に伴い、ワークステーションクラスターにおいて、その高い適用性そして費用対効果からも様々な環境に適合する、並列プラットフォームが開発されていくと思われる。

PVPコンピュータは、多重ベクトルプロセッサに単一の共用メモリー域を有した初期の並列システムである。PVPコンピュータは、いくつかの強力な汎用システムを供給しているが、特注プロセッサ価格が高いことから、ますます高価になっている。例えば、Cray T90シリーズやNEC SX-4が上げられる。

SMPは、ある意味ではPVPをさらに開発してできたシステムで、一つの共用メモリースペースを有してはいるが、ベクトルチップのかわりにRISC、もしくはCISCプロセッサを使用している。このように安価なプロセッサを使用しているために、SMPは価格の面でも性能の面でもPVPより優れ、RISC技術が非常に進んでいるため、プロセッサの基本性能までも優れたものになっている。

## (2) 並列ソフトウェア

MPPが広範囲で使用されていない原因として、並列ソフトウェアツールが完備されていないことによる。仮に超高速計算システムの包括範囲が変動するとしたならば、広範囲の技術が必要になり、その結果、アプリケーションの軽量化が確実に必要になるが、今のところ軽量化はまだまだ充分とはいえない。欧州や米国では国家的なレベルで、多大な資金を投入し超高速計算システムの研究を行い、その結果、ここ2、3年の間でMPI (Message Passing Interface) のような軽量の通信技術やPVMのような標準化により、安定したアプリケーションを確実に開発、生産できるようになってきている。

現行の並列ソフトウェアツールは層を形成していると考えられる。最も重要で最も下に位置する層がプロセッサ間のコミュニケーションである。標準化のために多大な研究が行われるのがこの層で、今では明確なAPI (Application Programming Interface) がある。メッセージ交換インターフェイス (MPI) と並列仮想機器 (PVM) の主に2つのメッセージ交換の標準があり、そのうちMPIは標準化団体組織によって正式に認められている。さらに、共用記憶機器は、IEEE Posix (Posix: ポータブルOS仕様) スレッド標準に基づいて、ますますスレッドプログラムライブラリを提供している。これらの3つの標準インターフェイスによって、安定した軽量のプラットフォームを開発することができる。アプリケーションプログラムからプロセッサ間の詳細なコミュニケーションを分離させようという試みの中で、次のレベルの並列言語が生まれた。このレベルにおいて、唯一実在する標準が、高性能FORTRAN即ちデータをFORTRAN90に並列拡張したものである。現在、様々な分野で、いろいろな研究がおこなわれ、特に並列C++についての確固とした標準書の草案をつくらうとするエスピリ基金によるヨーロッパ・プロジェクトの研究活動は注目に値するが、今のところ大した成果はあげられていない。

並列計算用コンピュータ言語は2つの種類に分けられる。一番目の種類の言語は、特性が明確にされているために、プログラマーはアプリケーションの中の並列性を容易に確認することができる。二番目の種類の言語においては、並列コンパイラに内在する並列性を並列コンパイラが検知できる能



力、及び並列コンパイラが並列コードを自動的に生成できる能力にプログラム言語が大きく影響を受ける。自動的に並列性を検知することができることによって、プログラマーは並列性を意識してアプリケーションを記述する必要はなくなったが、そのかわりコンピュータサイエンスという見地から、理論上のいくつかの難題に直面する。

並列コンパイラは1970年代半ば頃から存在してきたが、現在、能率的な汎用並列コンパイラは存在していない。コードの最適化におけるスケジューリングに関するいくつかの難しい問題と関連しているため、手続き型のコードと同じくらい能率的な汎用コードを開発できるか疑問視されている。

従って、超高速計算用コンパイラは、販売業者によって市販される様々な共用メモリープログラムツールとしばしば組み合わせられ、超高速計算用FORTRANがコンパイラを使用する時、並列データ構造を記述せよという指示に従うことになる。これらのツールは、スレッドライブラリー、もしくはメッセージ交換インターフェイスのトップで使用される。これらを確実に、能率的に利用するためには、かなりの特殊な技術が必要である。

ソフトウェアの最後の層は、デバッグ装置やプロファイラ、そして性能モニター等の並列プログラムツールを保護している。現在のところ、これらのプログラムツールは数少なく、販売業者間で特別使われる程度である。軽量なプログラミングツールも出回りはじめてはいるが、単一プロセッサシステムで利用可能なツールの数と比較すると、並列ソフトウェアにおける遅れははっきりとしているが、これからの研究開発に応じて発展する可能性はある。

### 3. 8. 3 並列シミュレーション・プログラム開発の事例

科学及び工学シミュレーションの分野で、コンピュータに関する様々な問題に取り組むためにソフトウェアの開発と改良に膨大な時間と費用がかけられてきた。それを批判する意見は、既存のコードを古臭い、誰からも愛されない“古いコード”として片づけてしまい、現在の高性能コンピュータには適さないと考えるが、FORTRAN言語及びC言語の何百あるいは何千ものラインを再構成（再利用）できるという実用性を見逃している。時間的なこと、既存のソフトウェアやデータとの互換性、保守のしやすさ、使い方が容易で費用対効果も高いなどのたくさんの理由から、走り書きのコードを書き換えるよりは、むしろ既存のコードを並列プラットフォームにポートする方法がしばしば適用される。

ポート化は、アプリケーションの特質及びアルゴリズム構成に大きく左右される。シミュレーションコードの中には、この既存コードを直接、並列プラットフォームにポートする方法では処理しきれないものもある。例えば、広範囲に分配されたデータ間の包括的な相関関係を有するようなアルゴリズムの場合には、走り書きのコードを再構成した方が容易なこともある。しかし、多くのアプリケーションにおいて、既存コードを直接並列プラットフォームにポートする方法で適切に処理できる。詳細に検討設計し、分配処理の必要性をよく理解することにより、非常に効率性の高い並列処理が可能になると考えられる。

並列化において重要な問題は、書き換えコードに関する研究者の知識が曖昧であるということである。特定のアプリケーションにコードを使用する科学者たちは、そのコードを自分たちで開発していくであろうし、そのアプリケーションについての専門知識を有しているグループも、それを開発していくであろうが、コンピュータサイエンスの分野では必ずしもそうではなく、いくつかの科学的、及び技術的問題が引き起こされる可能性がある。例えば、並列コードの分野には詳しくないコンピュー

ター科学者によって書き換えられた時、書き換えられた、もしくは並列化されたコードが妥当なものであるかについて非常に疑わしいこともあり得る。並列化に適されるコードがそのように書き込まれるため、一般に並列化テクノロジーをこのような問題に適用するのは非常に困難であると考えられる。ここでは、原子力コード分野での並列化について概観する。

### (1) タスクファーム (AEAのMascot)

それぞれ独立した計算に分類できるようなアプリケーションにおいて、タスクファームの手法は必要に応じて並列化するための優れた方法になることがある。タスクファームは、マスタープロセスがワークロードをそれぞれのタスクに分割して、それらのタスクを1つひとつのスレーブ・プロセスに割り当てる働きをしている。

Mascotとは、AEAのデコミッションングとラドウェストによって書かれた、モンテカルロ法による安全性評価プログラムである。その主な特徴は、いくつかのケースが与えられた時に、結果的にそれぞれ安全性がどのように評価されるのか、いろいろと算出できることである。Mascotは、最初クレイ Y/MPハードウェアを使って処理操作が行えるように設計されていたが、ベクトル化ができないという特性が本来備わっていたために計算効率が低い。計算効率化を高める方法として、RISCワークステーションと同一機種のカラスタで処理操作できるように設計し、並列バージョンで処理操作を行えるようにすることであった。

エンジン並列コンピュータセンターによる、この並列バージョンのpMascotは、マスタープロセッサに入出力及び制御機能を行うようにさせ、任意の数のスレーブプロセッサに計算させるものである。タスクファームに基づいた並列化作業のフレームワークを迅速に始動させるために、エンジン並列コンピュータセンターによって設計されたタスクファームライブラリ (PUL-TF) が使われている。コードには、さらに異常公差機能が導入され、それにより、中断時間内に完了されなかったタスクが別のスレーブプロセッサで処理されるようになった。

このタスクファームの手法をpMascotに適用した結果、並列コードがたった2つのSGIワークステーションでY/MPコードよりも約4倍も高速になったことから、タスクファームの手法のpMascotへの適用が成功であったことは明白である。4台のワークステーションでpMascotを処理操作させた際、処理時間を削減 (Y/MPでは70分かかっていたのが50分になったという例が100もある。) することができた。さらに、タスクファームモデルが本来スケラブルであるため、pMascotは高いレベルの並列効率で処理操作された。

### (2) 地下水の流動のモデル化(AEAのNammuとNapsac)

地下水の流動をコンピュータを使ってシミュレーションするのは、地層処分ヴァーチャル・エンジニアリング・システムの重要なアプリケーションの1つである。有限差分や有限要素の手法に基づいて、このシミュレーションをするために、いくつかのコンピュータパッケージプログラムが開発されている。

数値計算をするためには、多大な計算能力を必要とするいくつかの方程式が使われるようになる。地下水の流動のパッケージは、今のところ限られたユーザーにしか使われておらず、ユーザーからの需要が限られているために複雑な問題をシミュレーションすることはできない。

並列処理問題として、地下水の流動をシミュレーションモデルを取り上げたものが、AEAテクノロジーによってつくられたNapsac及びNammuプログラムである。これらのコードは、地下水の流動、亀裂及び気孔性媒体における核種の移動などをそれぞれモデル化するために使われる。AEAテクノロジーとの共同プロジェクトによって、エジンバラ並列コンピュータセンターは、より大規模でより複雑なモデル化を可能にするため、これらのパッケージプログラムの並列バージョンを開発した。

どちらのコードも非常に複雑で何万ラインものFORTRANコードを有している。それらのコードは同一のコンピュータを共有しており、そのため、エジンバラ並列コンピュータセンターは、このコンピュータを並列化を中心とした研究に利用している。並列化は領域分割方式に基づいて設計が行われた（すなわち、どのプロセッサにもモデルセクションが割り当てられている。）。各サブドメインにおける並列計算、単一のプロセッサを使った順次計算、そして最終的な並列計算の3つの段階でコンピュータによる全体的な処理操作が行われるようにソフトウェアが再構築された。このようにして、直接的な計算方法が保持される一方で、並列化によって処理速度が早くなっている。メッセージ交換（MPI）用の標準ライブラリを使ってプロセス間のコミュニケーションも行われた。この作業を完成するのに一人の人間の年齢ぐらいの年月が費やされた。

その結果、適度の数のプロセッサを使うと能率的に処理が行われることがわかり、データを能率的に分解することの必要性が強調された。このことはまた、例えば直接的な計算方法よりは、むしろ反復型の計算方法を使うなど、たくさんプロセッサを使って適切に問題処理できるような解法アルゴリズムについての研究をさらに深めていくことが必要であることを示していた。

### 3. 8. 4 並列計算プログラムの開発

問題を速く解くための1つの方法は、その問題を細分化し、すべての部分を同時に解くようにアルゴリズムを組み立てることである。細かく分ければ分けるほど、問題の解決は速くなるが、分散させる効果が無くなるまでに細かく分け過ぎては問題そのものの意味を失う。このような、簡単かつ明快な観察が、すべての並列プログラミングの基本となっている。また、これは、2章にて論じたオブジェクト指向シミュレーション・モデルの考え方にうまく適合する。

「並列プログラム（parallel program）」は、大きな、または難しい問題を解くために、細分化手法を使ったプログラムである。ソフトウェアを組み立てるということは、組織的または技術的な挑戦である。並列プログラムは、多くのプロセッサを同時に使って実行することを目的としている。それぞれのプロセッサは、問題の一部を分担して、全体が同時に実行を進める方式である。1つの問題を解くのに複数のプロセッサを使って、1台のプロセッサの場合の複数倍の性能を実現することが並列計算の狙いである。現実には、このような理想的な線形な性能向上をいつでも実現できるわけではないが、並列処理は非常に多くの種類の現実の重要なプログラムを高速に実行するための、確実な方法と考えられる。

並列プログラムを実行する複数のプロセッサ（ハードウェア）は、すべてが1個の箱の中に実装されていてもよい（この場合には、マルチプロセッサまたは並列処理計算機となる）し、またネットワークで結ばれた独立した自律的な計算機の集合であってもよい。マルチプロセッサのプロセッサ台数は、数台から数千台にまで亘るであろう。また、これらのプロセッサを接続するには多くの方式がある。ネットワークはLANでもWANでもよく、ネットワークの中の計算機はどのようなものでもよい。

並列処理の特徴に、「非同期並列処理 (asynchronous parallelism)」がある。非同期並列処理とは、(おおまかにいうならば) それぞれのプロセスは好きな処理を実行できるような並列処理のことである。これに対して、「同期並列処理 (synchronous parallelism)」はすべての処理が同期を取って進められ、基本的には同時には同じ処理しか実行できないという特別な方式である。

ソフトウェアの面では、一般に、ソフトウェアを複数プロセスの集合として組み立てる理由は、いろいろあるといえる。並列処理による高性能の達成というのは、その1つにしか過ぎないであろう。また、性能向上以外の狙いを持ったマルチプロセス・プログラムがある。それは、協調ソフトウェアと呼ばれ、それはコンピュータサイエンスの分野以外にも見られるものである。

世界は全体的にみれば、非同期集合を集めて作られた非同期集合である。社会システム、経済システム、機械システム、生物システムは、すべてこのように考えることができる。人間の集合も、プロセスの集合が使っているのと同じソフトウェア構造によって説明できる。協調ソフトウェアの芸術と科学を扱う1つの方法は、協調システム一般の解析というより、広い課題への小宇宙的な紹介とすることである。これは、新しいモデル化の戦略であるとも言える。

協調ソフトウェアは、多くの独立した活動から、いかに1つの仕事が合成されるかという研究のための場でもある。並列プログラミングにおいては、計算言語と協調言語の両言語が必要である。計算言語で記述された個別の活動の集合を、統一されたプログラムに組み立てるための鍵(キー)となるのが協調言語である。従って、計算やローカルなデータオブジェクトの処理には、計算言語 (Fortran, C, lispなど) を使う。並行実行、ならびにそれらの間での通信の実現には、協調言語を使う。また、標準的な計算言語の機能と協調言語の機能を1つの超言語に含めることもできる。計算言語の例はC言語である。協調言語の例はLindaである。

### (1) 並列性

プログラムの実行を高速にするための実証された手法が並列処理である。どの時代においても、最も強力な計算機は並列処理計算機であると思われる。何故ならば、最善を尽くして、その時点での最高の技術を使って、最も高速のプロセッサを設計したとすれば、それを2台使えばより高速のシステムを構築できるのは明らかであるためである。他方、中位レベル、または最下位レベルのシステムが上げられる。おおまかにいえば、時分割処理計算機やファイルサーバ、またはワークステーションやパーソナルコンピュータ (この2つについては急速に差異がなくなりつつある) が、それぞれ対応する。

### (2) 協調

これからのソフトウェア開発の現場では、「すべてのプログラマにとって、並列処理は基本的なものとなる」であろう。さらに、「協調がコンピュータサイエンスの中心的な話題になる」であろう。並列化技法について学ぶということは、協調プログラミングの一種について学ぶということであると考えられる (並列技法の多くは、他の種類の協調応用にも文字通り適用可能である)。協調プログラミングについて学ぶということは、すべての計算の基本を学ぶということにもなる。個々の「エージェント」を接合してプログラムを作っていくというプロセスを指して協調と呼ぶこととする。ここでいうエージェントとは、例えばオブジェクト、プロセスやタスク、またはスレッド (thread) など、

それ自身で独立した実行主体のことである（それ自身で独立した実行とは、他と並行かつ非同期的な実行のことである）。

個々のエージェントを協調させるということは、接合してできあがった集合をプログラムとみなすことができるということである。協調させられたエージェントは同じ問題を処理している。協調に当たっては、これらの独立な活動体がお互いに必要に応じて通信したり同期をとったりできるようになっていなければならない。協調言語は、このような要求を満たすように設計されている。協調は概念的にも重要である。協調は、ソフトウェアの空間の基礎となる2つの直交座標の1つとして扱えるからである。有用な計算機システムを構築するには、計算と協調が不可欠である。すべての計算は、最終的には外部世界とつながらなければならない。つながれるということには、協調するということが含まれている。計算機とその利用者は、その計算が何か意味のあることを行うことを目的としている限り、互いに協調関係になければならない。

この簡単な観察の帰結は次のとおりである。オペレーティングシステムは、見方によっては、特別の協調言語といえる。また、利用者／計算複合体は協調そのものである。さらに、ソフトウェアプロセスと人間の「プロセス」の両方が協調するためには、これを実現するための新しいソフトウェアが必要であるということになる。ここで「新しい」といったが、これは必ずしもその例すら私たちが分かっていないということの意味はしているわけではない。ただ、その概念がまだ具体的に一般化されてはいないということである。このソフトウェアを、「チューリングウェア (Turingware)」と命名する。人間とプロセスが、普通の協調プログラミング手法によって、単一構造の中に織り込まれる。この複合体の中では、プロセスも人も、お互いに自分の相手がプロセスなのか人間なのかを意識しない点に、新しい情報モデルが考えられる。

UNIXのコマンドインタプリタのようなシステムは、場合によっては端末のスクリーン（ということとは究極的には人間ということになる）と普通のファイルを区別なく扱うことができる。その意味で、これは「チューリングウェア」ともいふべき性格を持っているともいえる。しかし、究極のチューリングウェアにおいては、人間は単にデータ供給者の片方というだけではない。人間はネットワークのノードとして、積極的に計算に関与する。

言い方を替えるなら、システムのモジュールの1つは人間そのものである。目標は3つある。まず最初の目標は、速くて柔軟で高機能な情報の交換の枠組みを作るために、並列プログラミングの手法を採用することである。なお、この枠組みのことを、「協調枠組み」と呼ぶことにする。例えば、統合化された地層処分ヴァーチャル・エンジニアリング・システムはチューリングウェアであるべきである。

並列プログラムと分散システムは、どちらも協調プログラムである。並列プログラムは高速実行のために、並行性を利用する。分散データベースや電子メールおよびファイルサーバのような分散システムは、計算機が物理的に分散配置されているので、これを活用するために並行プロセスを使っている。50台のワークステーションから構成されている分散メーラシステムでは、それぞれのワークステーションの上で1つずつのプロセス、総計50のプロセス、が走行している。しかし、この目的はなんにも1台の計算機上でのメーラ (Mailer) の50倍の性能を実現することではない。協調が目的である。

協調プログラムを構成する構成要素は、それぞれ1つのプログラム言語で書いてもよいし、またいくつかの異なるプログラム言語で書いてもよい。言葉を替えるならば、協調プログラムは多様な

言語のプログラムでもある。協調プログラムの構成要素は、距離的にもまた時間的にも離れていてもよい。2つのプロセスが同時に2つの異なった計算機で実行され、一方が他方に情報を送る場合には、その情報は空間をまたがって送られることになる。もし、この2つのプロセスが同じ計算機で実行される場合でも（例えば一方が火曜日に実行されて、他方が木曜日に実行される）、空間的に分散した場合と論理的に同じ協調問題となる。「火曜日プロセス」は「木曜日プロセス」に情報を送るが、これは空間をまたがった情報伝達ではなく、時間をまたがった情報伝達である。結果は、同じように協調プログラムとなる。「時間をまたがった協調」言語は、普通はファイルシステムまたはデータベースの形をとることが多い。このため、時間をまたがったプロセス集合と空間をまたがったプロセス集合とは違うものという認識をしがちであるが、基本的な概念は全く同じである。並列処理という範疇で考えられた多くの協調手法は、そのままファイルシステムやデータベースシステムに適用できる。

これからは、利用者は小さいプログラムを組み合わせると大きいプログラムを作るという場面により頻繁に遭遇するであろう。このような手法は、これまでの技術の集積を補うものである。コンピュータネットワークは、もはや特別のものではない。ネットワークを統合システムとして扱うことには、むしろ望ましい。今後は、どのプログラミング言語をとっても、それだけでよいということにはならない。少なくともある程度の期間は、多数の言語が存在し続けるであろう。多数の言語が混在したプログラムを扱えることは、明らかに望ましいことである。特殊目的のハードウェアも現われ出した。ここにグラフィックマシンとデータベースサーバと汎用計算ノードがあれば、これら3つの計算機をすべて使うような応用を考えるのは、きわめて自然なことである。そして、プロセッサの数は増え続けるであろう。プロセッサが安価でたくさんあるというような環境では、協調プログラミングは必須である。応用プログラムを実行するよりも、たくさんプロセッサがあるような時代になれば、1台のプロセッサしか使えないようなプログラムは完全に非社会的となるであろう。そのような面から、今後の統合化システムの開発において協調計算技術の開発は重要な課題である。

### 3. 8. 5 協調作業のための並列パラダイム

並列アルゴリズムを使ってどのようにプログラムを作るのだろうか。その過程において、次の3つの基本的な事項に特に注目すべきである。それらは、結果並列法、手順並列法、および専門家並列法である。これらの言葉の意味は後で定義する。これらの3つを、並列性の基本パラダイムとする。これらは、並列性について考える上での、また並列プログラムを設計する上での、さらに並列性を解析する上での、3つの異なったアプローチとなる。これらの基本パラダイムに対応して、3つの並列プログラミング手法（概念を実用的なプログラムに変換する実際的な手法）がある。それらは、メッセージ交換、分散データ構造、およびライブデータ構造である。それぞれのプログラミング手法は、並列プログラムにおけるプロセスの役割とデータの分散を異なった視点から見ている。基本パラダイムとプログラミング手法は、必ずしもこれだけに限られるものではない。しかし、ここでの目的は、協調枠組みに適用できる概念的なパラダイムとプログラミング手法を理解し、それらの間の関係を調べることである。

#### (1) パラダイム

並列プログラムは、それぞれのパラダイムに対応して、自然なプログラミング手法がある。それぞ

れの手法を、お互いに明確に関連づけることができる（言葉を変えていうならば、「手法 x を使ったプログラムを手法 y を使ったプログラムに、次のような明確と定義された方法で変換することができる」）。したがって、並列プログラムを書くには、問題を最も自然に表わすパラダイムを選び、そのパラダイムに最も合った手法でプログラムを書く。もし、できあがったプログラムが十分に効率的でなかった場合には、自然な手法の代わりに更に効率のよい手法を考えプログラム変換をする。

並列性を、プログラムの結果、プログラムの動作の手順、およびプログラムを構成する専門家の集合という3つの観点から捉えることができる。例えば、家を建てたいとする。このときに、大勢の人が働く並列処理を採用するのは当然の成り行きである。しかし、並列処理が実際に活用される場面はいくつかに分かれる。まず、並列処理を最終の生産物（結果）によって考えることができる。結果は多くの個別の構成要素に分けられる。前面の壁、後ろの壁、両脇の壁、内部の壁、土台、屋根などがある。結果をこれらのように構成要素に分類すれば、それぞれの構成要素の製造を並行して進めることができる。そして構成要素の製造が終わり次第、それらを組み立てていく。土台の製造に作業者を一人、前面の外部壁の製造に作業者を一人、さらに両脇の壁の製造に作業者を一人といったように作業者を割り当てる。すべての作業者は、作業を同時に開始する。別々の作業者が、土台の据付や外部壁の組み立ておよび屋根の棟上げなどに従事する。他人の作業の終了を待ち合わせるが必要になるまで、それぞれの作業者は並列に作業を進める。まとめると、それぞれの構成要素を作る作業に対応して一人ずつ作業者が割り当てられる。そして、他の作業の終了を待つことが必要になる時点（これがどこかは、仕事の内容によって自然に決まる）まで並列に作業を進める。これを結果並列法という。

他の考え方として、並列性を仕事をする作業者の観点から眺めることができる。家を建てるには、それぞれの技術を持った作業者が必要である。例えば、測量士、土を掘る人、基礎を作る人、大工、屋根ぶきなどである。家を建てるのに必要な、それぞれの技術に対応する専門家を集めて、建築グループを構成する。建築は全員で一緒に始めるけれども、最初は、ほとんどの作業者は手持ち無沙汰で待つことになる。しかし、建築が進むにつれて、多くの作業（したがって作業者）は一緒にできるようになる。大工と基礎工は一緒に並行して働くことができるようになる。また、屋根ぶきは、鉛管工が管路を設置し電気工事屋が配線をしているかたわらで屋根をぶくことができる。大工は木工作業をするが、その間に多くの仕事が並行して進められる。このような進め方は、特に流れ作業的（パイプライン）な仕事に適している。流れ作業では、1つの製品が次から次へとラインを進んでいくことによって完成されていく。もし、数軒の家を建てるのであれば、大工が1軒目の仕事をしている間に、基礎工は2軒目、そして測量士は3軒目の仕事を、それぞれすることができ。このような作業法は、1軒の家を建てる場合のように製品が1個しかない場合でも並列性を引き出す方法として適用できる。作業者は、それぞれ特定の仕事に割り当てられた専門家である。仕事の性質からの制限に抵触しない限り、お互いに並行して作業する。このような並列性を専門家並列法という。

最後に、家を建てる時にやらなければならない仕事の手順という観点から並列性を眺めてみよう。仕事のリストを書き出して作業を集める。作業者はそれぞれ何か仕事を選んで、それを始める。仕事が終わったら、作業者は次の仕事を選択する。作業者にはそれぞれ特別な区別はない。作業者は、初めから、予めどれか特別の仕事に割り当てられているのではない。必要になったら、どのような仕事でもする。仕事の手順に従って仕事が進む。作業者は必要に応じて、何回でもこの手順を参照する

ことになろう。手順に現われる仕事としては、お互いに順序関係のない仕事の集合もある。この場合には、その集合の中から好きな仕事を選ぶようにと作業者に指示することになる。また、仕事の間に順序関係がある場合もある。その場合には、まずこれを行い、次にこれに取りかかれといった指示をすることになる。まず、土台がいる。土台の構築は、手順の中では仕事の集合として書いてある。次に骨組みがいる。次は屋根である。そして、壁やしっくいへと進む。「何でも屋」のチームを作る。メンバは誰でも何でもできる。まず、みんなで集まって土台を作る。次に同じメンバで壁を作る。さらに屋根へと進む。次に何人かは鉛管の配管をする。その間に残りの人間は電気工事や他の仕事をする。誰が何をするかは、全く任意に決められる。まとめると、手順書から仕事を選んで作業者をそれに割り当てて、その仕事に従事させるということを全部の仕事が終了するまで繰り返す。作業者は、仕事上の制限にでくわすまでお互いに並列に作業を進める。このような方法を手順並列法という。

このようなパラダイムは、抽象的な対象について整理することを要求するが、統合化システム開発における基本的なアーキテクチャ設計思想となるため、システム設計の初期の段階で十分に検討されなければならない課題と考えられる。これらの並列のパラダイムの区別は、時には曖昧になることもあると思われる。

また、これらを混ぜ合わせた方法も考えられる。例えば、専門家並列法に手順並列法を併用することを考えてみよう。専門家のチーム（例：大工）が、大工仕事を手順並列法に従って行うというのが、その例である。いずれにせよ、上記の3つのパラダイムは、問題を扱う上で明確に区別できる3つの考え方である。それは、例えば地層処分性能評価のような複合問題を対象とするシステムを開発するときの問題解決の基本戦略である。

以上のように、結果並列法は最終製品の形態に注目した考え方である。専門家並列法は作業者の組合せに注目した考え方である。そして、手順並列法はすべき仕事の表に注目した考え方である。

これまで、家の建築を題材に話を進めてきた。次の（そして最終的な）課題はソフトウェアである。しかし、これらのパラダイムは、人間のありとあらゆる活動にまんべんなく適用できると考えられる。これらのパラダイムのソフトウェアに関する適用性は下記のように考えることができる。

- ・最終結果となるデータ構造体を中心に並列処理を考えることができる。  
結果のすべての要素について同時に計算することによって、並列処理を達成できる。
- ・複数のタスクの処理手順を基に並列性を考えることができる。  
複数のタスクの実行に、多くの作業者を割り当てる。
- ・ある種の論理的なネットワークを構成している専門家の集合に基づいて、並列性を考えることもできる。この論理的なネットワークのノードの中で同時に実行できるものがあれば、それが並列性になると考えられる。

それでは、どのパラダイム（どの並列性）を使うかということは、どうしたらわかるであろうか。

再び、家の建築について考えてみよう。結果としては、3つのすべてのパラダイムが使われる。あらかじめ工場で部品（壁、屋根、階段など）を作っておいてこれを現場で組み立てる場合には、これらの部品は、全てバラバラにそして（理論上は）同時に工場で作られる。

これは、結果並列法そのものである。他方、「納屋作り」においては、作業者のグループは必要な仕事のリストに出てくる項目を次から次とこなしていくことになる。これは、手順並列法の一形態で



ある。しかし、最も普通の方法は、手順並列法を採用したある種の専門家並列法である。それぞれの作業員（またはチーム）は専門を持った専門家である。そして、それぞれの専門家が同時に仕事を進めていくことによって、並列性が生まれる。そして、同じ専門家チームの多数の作業員が、手順に従って作業することによって2番目の手順並列法が生まれる。

ソフトウェアの場合にも、問題によっていずれかのパラダイムが他よりも自然な選択となる。選択は問題に依存している。場合によっては、選択は自然である。また場合によっては、全部のパラダイムが等しく自然である。このように複数の選択の余地があることが、混乱と当惑の基になることが多い。しかし、これはむしろ並列性に富んでいてプログラムには並列性の選択の幅が広い場合の多いことの証ととらえるべきである。多くの場合、並列プログラムを設計する最も容易な方法は、結果のデータ構造体を基にすること（結果並列法）であると思われる。

- ・自分のプログラムは、結果として多数の要素を持つデータ構造体を出力するのか（またはこのような構造体を基に結果を考えることができるか）を考えてみる。
- ・そのデータ構造体の各要素が、お互いにどのような関連があるのか、また入力とどのような関連があるのかを正確に指定できるかを考えてみる。もし、これが可能であれば、結果並列法に基づいてプログラムを書くことは（適当なプログラム手法を知っているものとして）簡単である。おおまかには、そのようなプログラムは、次のようになるであろう。（データ構造体を、\*\*のように構成する。この構造体の全要素について同時に計算する。すべての値が求められたら終わりとする。）

結果の構造体の要素は、お互いに依存しないこともある。そのような場合には、すべての計算は同時に始まって並列に進行する。または、ある要素の計算は他の要素の値が決まらないと始められないかもしれない。そのような場合には、全要素の計算は同時に始まるが、要素によっては計算は遅れることになる。そのような要素については、それが依存している要素の計算が終わるまで、計算を開始できないことになる。

その結果、並列処理技術は単に計算速度の高速化を狙う算法ではなく、むしろ、協調計算システムのアーキテクチャ設計に係わる基本的な技術手法と考えることが妥当であると思われる。また、そのような立場から並列処理を捉えた場合、並列処理は協調計算の基本パラダイムとして以下の3つのアプローチが考えられる。

- ・結果並列法 : 対象を構成要素の並列性から組み立てる  
例えば、現象全体を個別の現象結果を並列させて説明する
- ・専門家並列法 : 対象を処理の並列性から組み立てる  
例えば、現象全体を個別の現象過程を並列に連成させて説明する
- ・手順並列法 : 対象と手順（シナリオ）の並列性から組み立てる  
例えば、現象全体をそれを生じるシナリオを並列に組み合わせて説明する

以上のように整理すると、協調計算に係わる並列処理パラダイムは、地層処分性能評価そのものに対するパラダイムであり、重要な考え方を示唆していると考えられる。それはまた、性能評価手法という枠組みを組み立てるための、具体的な方法論を情報処理モデルによって組み立てる可能性を示し

ている。従って、本件は、オブジェクト指向におけるパラダイムとともに、今後の地層処分性能評価シミュレーション・システム開発における重要な研究開発課題である。

### (3) プログラミング・パラダイム

メッセージ交換では、多くの並列プロセスを生成して、それぞれのデータ構造体をどれかのプロセスに内包させる。プロセスはメッセージの交換によって通信する。メッセージ交換法では、プロセス間で共用されるデータオブジェクトはない。それぞれのプロセスは、自分のローカルデータ・オブジェクトにアクセスすることしかできない。通信のためには、プロセスはローカルスペースから他ヘデータオブジェクトを送る必要がある。このためには、プログラマは明示的にプログラムの中にデータ送信とデータ受信命令を組み込まなければならない。

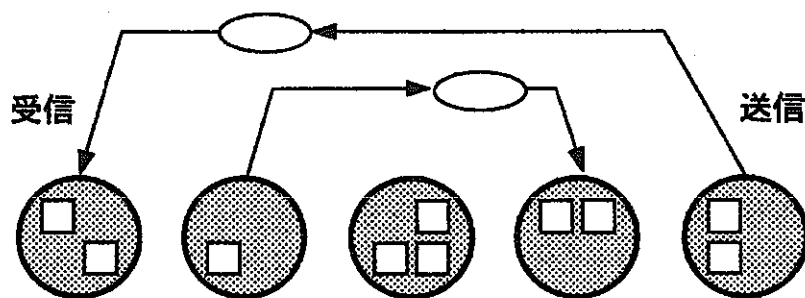


図3. 8-1 メッセージ交換の考え方

図3. 8-1は、メッセージ交換でプロセスの数と、それらの間の関係がプログラムの構造を決定する様子を示している。並行プロセスの集合はメッセージを交換して通信する。それぞれのデータ・オブジェクトは、いずれかのプロセスの中に包含される（プロセスは円で、データオブジェクトは四角で、メッセージは楕円でそれぞれ示されている）。もう一つの考え方としては、プロセスを独立した単位としてみなすことをやめて、結果として得られるデータ構造体の形でプログラムを作成する方法がある。この場合、このデータ構造体の各要素が、それぞれ独立したプロセスに相当すると考える。

計算が終了したところで、これらはデータオブジェクトとなる。これらのプロセス相当のデータは、メッセージを送って通信するのではなく、それぞれデータ構造体の要素として「参照」し合う。もしプロセスPがQのためのデータを持っているならば、PはQにメッセージを送るのではなく、Pは処理を終了して結果を出す。Qはそれを直接読み出す。このような仕組みをライブデータ構造体プログラムという。この考え方は半逐次的なデータフローをモデル化することができるので、ネットワークモデルによる平行処理などを考える場合に都合が良いと思われる。

このような情報処理モデルでは、データ構造とその要素の数、そしてそれらの間の関係がプログラムの構造を決定する。それぞれの並行プロセスは、いずれかのデータ・オブジェクトの中に包含される。それらは、その要素の通信についてのみ責任を持っているため、通信のための明示的なメッセー

ジの送受命令は使わない。他のプロセスが生成した値が必要になった場合、そのプロセスが内包されているデータ・オブジェクトを参照する。メッセージ交換もライブデータ構造体も、それぞれすべてのデータ・オブジェクトが並列プロセスに分散されているという点においては類似している。

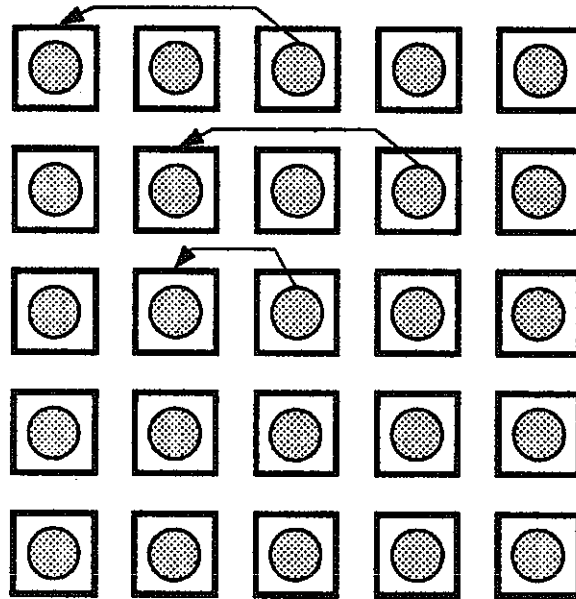


図3. 8-2 ライブデータ構造体の考え方

メッセージパッシングでは、プロセスはプログラマによって明示的に生成されて、通信も明示的に行われ、データを繰り返して他のプロセスに送ることができる。他方、ライブデータ構造体では、データ構造体を作っていく過程においてプロセスが暗に生成される。またデータ構造体の要素を参照することによって、通信も暗に行われる。それぞれのプロセスは、プログラムの他の部分から参照される1個のデータを作るだけである。データ構造体が、すべてプロセスに内包されるという考え方（メッセージ交換）から、プロセスがデータ構造体に融合するという考え方（ライブデータ構造体）の間には、データ・オブジェクトとプロセスの集合の間の区別をする中間的な方法がある。データ共有のために、多くのプロセスがデータオブジェクトや構造体に直接アクセスをする。共有データオブジェクトがあるので、共有オブジェクトにデータを残すことによって、プロセスは通信して協調することができる。これを分散データ構造体プログラムという

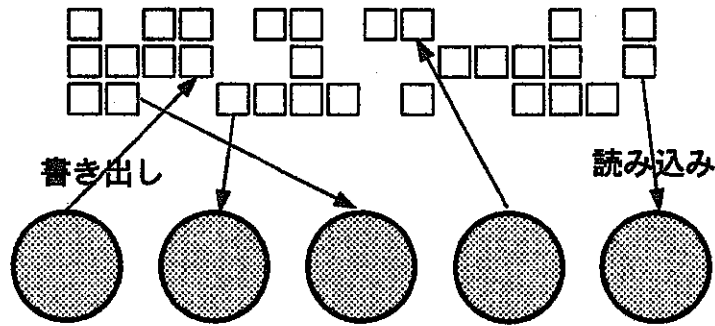


図3. 8-3 分散データ構造体の考え方

分散データ構造体の考え方は、並行プロセスとデータ・オブジェクトがプログラム構造の独立した部分となると捉えることである。プロセスは共用データオブジェクトを読み書きすることによって、通信すると考える。図3. 8-3にその考え方を示す。結果並列法は、ライブデータ構造体プログラムで自然に表現できる。例として、ベクトルの加算を考えてみる。この応用の中核はライブデータ構造体である。ライブデータ構造体はSというn要素のベクトルである。Sの各要素の中に、それぞれの位置について $A[i] + B[i]$ を計算するプロセスがある。プロセスが終了すると消滅して、結果のみが残る。専門家並列法はメッセージ交換が利用しやすい。プログラムを組むには、それぞれのネットワークノードに対応してプロセスを設けて、メッセージでエッジ間の通信を実現する。

## (2) n体問題へのアプローチ

前述してきた並列化パラダイムについての例として、簡単なn体問題シミュレータを考えてみる。シミュレーションの繰り返しごとに、各物体と残りの物体間の作用力を計算し、これに基づいてそれぞれの物体の位置を更新する。この問題を家の建築の問題と同じように扱ってみる。結果並列法、手順並列法ならびに専門家並列法の3つの方法がある。結果並列法では、問題を次のように言い換えることができる。n体を与えられて、シミュレーションをq回繰り返したいとすると、問題は、j回目の繰り返しの後のi番目の物体の位置が、要素 $M[i,j]$ であるような行列を計算することになる。行列の0列が、それぞれの物体の初期位置であり、最後の列が最終位置である。ライブデータ構造体を使うための第1ステップの設計はこれで終了する。第2ステップは、それぞれの要素の値を他の要素の値を使って更新する処理を定義することである。繰り返しjにおける物体iの位置を計算する関数を、 $position(i,j)$ と書くことにしよう。当然 $position(i,j)$ は1回前の繰り返しにおけるそれぞれの物体の位置に依存する。

それは、行列のj-1列の値に依存する。適当なプログラミング言語を使って、この処理を書けばよい。 $M[i,j]$ の値を、 $position(i,j)$ で計算するプログラムを書く。関数 $position$ を呼び出すごとに、暗にプロセスが起動される。そして、すべてのプロセスの起動と実行が並列に行われる。もちろん、1番目の列の計算が終わらなければ2番目の列の計算は始められない。関数 $position$ が、まだ計算の終わっていない $M[x,y]$ の値を必要とする場合には、その計算が終わるのを待たなければならない。0番目の列の値は初期値として与えられる。これを使って1番目の列の値は並列に計算できる。そして2番目以降の

列の計算も同様に進む。もし物体の間に働く力が対称であれば、上のプログラムは必要以上の余計な計算をしていることになる。物体AとBの間の力は物体BとAの間の力に等しいからである。この問題は簡単に解決できるが、ここでの目的は最も簡単な方法を示すことにある。

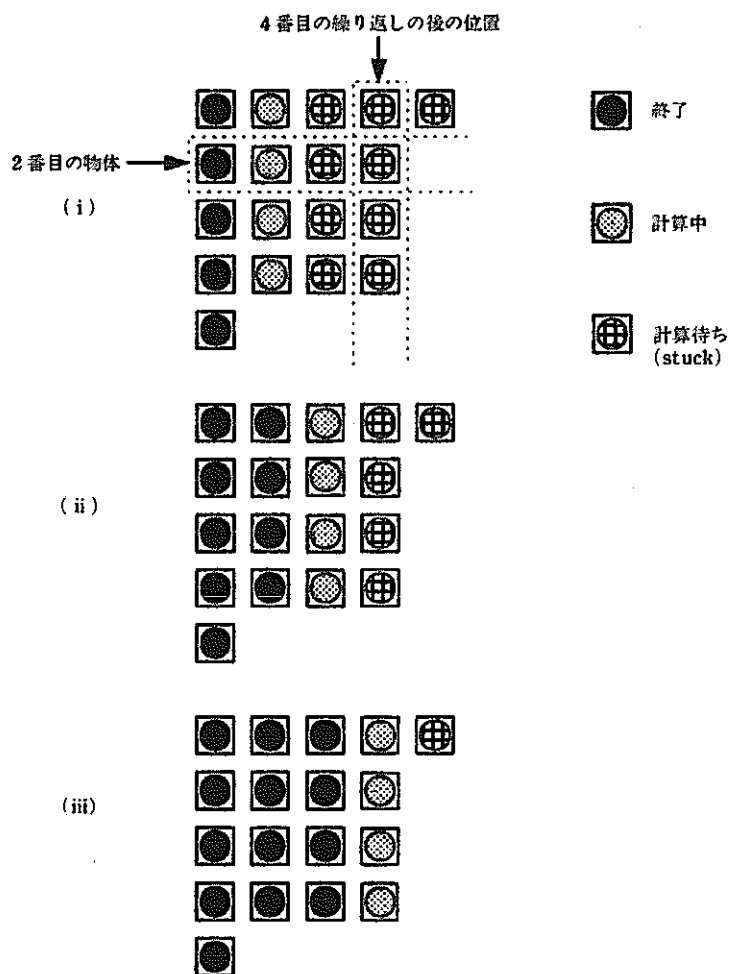


図3. 8-4 ライブデータ構造体によるn体問題解決へのアプローチ

始めに $nxq$ の行列を作り、それぞれの要素の中にプロセスを作る。要素 $M[i,j]$ の中のプロセスは、それぞれの物体の最後に計算した位置が入っている前の列の値を使って、 $j$ 番目の繰り返しの後の $i$ 番目の物体の位置を計算する。列 $j$ のプロセスは列 $j-1$ のプロセスが終了するまで実行を開始できない。列 $j-1$ のプロセスが終了すると、列 $j$ のプロセスは並列に計算できる。以上の手順で行列全体の計算が終了するまで、各列の計算を並列に続けることができる。また、手順並列法によって問題を解決することもできる。仕事の手順は、「集合に含まれているすべての物体について、次の位置を計算するという処理を繰り返し適用すること」である。プログラムを書くには、まずマスタープロセスを作って、次にそのマスタープロセスに $n$ 個の初期タスク・ディスクリプタ（各物体に対応している）を生成させる。

最初の繰り返しでは、ワーカープロセスの集合の中のワーカーはすべて平等であり、タスクディスクリプタが無くなるまで、次々にワーカーはタスクディスクリプタを取る（これによって、すべての

物体は新しい位置に移ったことになる)。この同じことを繰り返す。ワーカーが1個しかない場合には、繰り返しごとの計算時間は $n$ の二乗に比例する。2個になると、これが $n$ の二乗/2になる。最後の繰り返しの結果の物体の位置情報は、分散テーブル構造体になってしまう。これをそれぞれのワーカーが直接参照する。

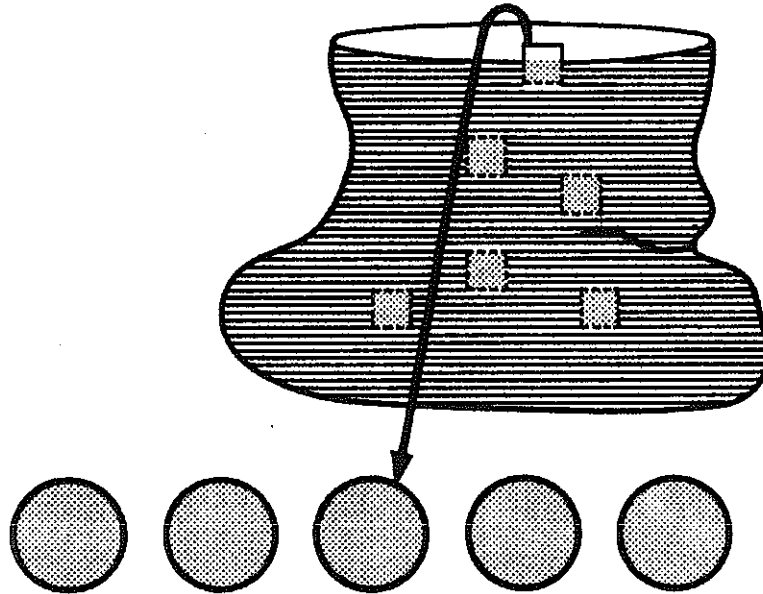


図3. 8-5 分散データ構造体によるn体問題解決へのアプローチ

それぞれの繰り返しの後で、ワーカーは分散バッグからタスクを取り出して、対応する物体の新しい位置を計算する。各物体の最後の位置に関する情報は分散テーブルに入っている。それぞれの計算の後で、ワーカーは表（テーブル）を更新する方法もある。最後の位置に関する情報は、後で必要になるかもしれないので消さないでおく。また、新しく計算されたデータをマスタープロセスに送る方法もある。この場合には、マスタープロセスが各繰り返しの後で、テーブルを一度に書き換える。最後に専門家並列法では、まずプロセスの列を生成する。それぞれのプロセスは、それぞれの物体に対応している。すなわち、プロセスは、シミュレーションの間、それぞれ特定の物体の位置の計算に責任を持つ。各繰り返しを始めるに当たって、プロセスはそれぞれメッセージを使って、自分が担当している物体の位置を通知する。プロセスはすべて同じ振舞いをする。繰り返しの初めに他のプロセスにデータを送るとともに、他のプロセスからデータを受け取る。メッセージの中にはプロセスが分担している物体の位置を計算するのに必要なデータが含まれている。このような処理を繰り返す。

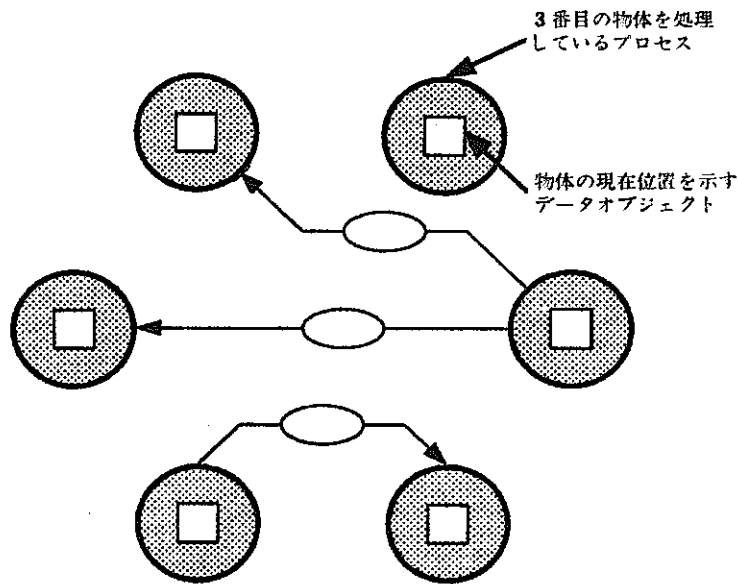


図3. 8-6 メッセージ交換によるn体問題解決へのアプローチ

ライブデータ構造体プログラムは $nq$ 個のプロセスを生成 ( $q$ は繰り返しの数で、 $n$ は物体の数) し、分散データ構造体プログラムは、好きなだけのワーカーを生成するが、このメッセージ交換プログラムは $n$ 個のプロセス (物体1個当たりプロセス1個) を生成する。

他の2つの方法では、プロセスが各物体の最後の位置に関する情報を必要とするときには、グローバルデータ構造体を参照した。また、ライブデータ構造体版では、このグローバルデータ構造体はライブデータ構造体で、その中にプロセスが包含されていた。メッセージ交換版では、プロセスは自分の外部のデータオブジェクトにはアクセスしない。プロセスはメッセージを送り合って、情報を交換する。

### (3) パラダイムと効率

これまでに述べてきたプログラミング手法は、問題に則したパラダイムを選んで、そのパラダイムに適したプログラミング法でプログラムを書き、もし必要ならば、そのプログラムをより効率のよい他の方法を使ったプログラムに修正するというものであった。もし、自然な方法が効率のよいプログラムにつながるのであれば、もちろんプログラムの修正は必要ない。そうでなければ、手法と効率の関係を理解することが必要となるためである。並列協調処理では、取り扱うエージェントの数が増えるにつれて、計算量が爆発的に増大することが考えられる。従って、一度作り上げたアルゴリズムをより効率的なものにすることは重要な課題である。ここでは、この関係についての一般論について述べた後で、この効率向上のための修正について考える。これまで述べてきたパラダイムの主な関係を図3. 7-7に示す。

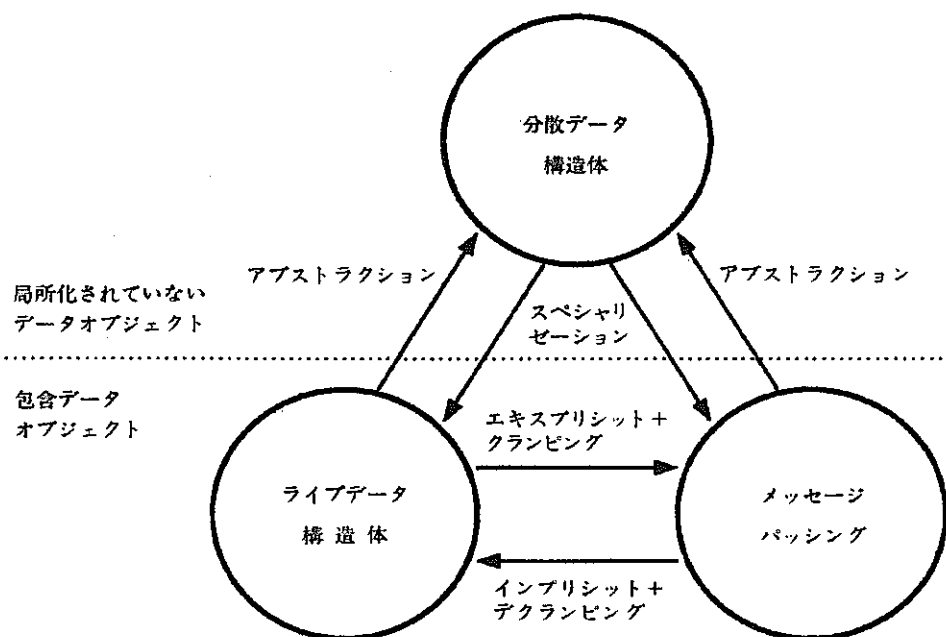


図3. 8-7 並列パラダイムの関係

ライブデータ構造体とメッセージ交換は、いずれも束縛データオブジェクト (captive data object) に基づいている。すべてのデータオブジェクトは、いずれかのプロセスに結び付けられている。分散データ構造体手法は、非集中データオブジェクト (de1ocalized data object) に基づいている。オブジェクトは、いずれのプロセスにも結び付いていないで自由である。ライブデータ構造体またはメッセージ交換プログラムを分散データ構造体に変換するには、アブストラクション (abstraction) を使えばよい。アブストラクションは、データオブジェクトをそれが結び付いているプロセスから切り離して、分散データ構造体に入れる。こうすることによって、プロセスは、もはや単一のオブジェクトやオブジェクトの集合に注意を払う必要はなくなる。分散データ構造体からライブデータ構造体やメッセージ交換プログラムに変換するには、スペシャリゼーション (specialization) を使う。スペシャリゼーションは、それぞれのオブジェクトをどれかのプロセスに結び付ける。以上の考察から、ライブデータ構造体とメッセージ交換の間には緊密な関係があることがわかる。ライブデータ構造体をメッセージ交換に変換するには、通信を明示的に行う必要が出てくる。このためにクランピング (clumping) を使うことがある。ライブデータ構造体を使ったプログラムにおけるプロセスは、他のプロセスへ情報の明示的な通信を行う必要はない。単に終了して、結果としての値を生成するだけである。

メッセージ交換プログラムでは、データを送るプロセスは明示的に「メッセージを送る」命令を実行しなければならない。ライブデータ構造体プロセスが他のプロセスからのデータを使うときには、そのデータ構造体を参照する。これに対してメッセージ交換プロセスでは、「メッセージを受け取る」命令を明示的に実行しなければならない。メッセージ交換が、ライブデータ構造体の単なる言い換えにしか過ぎないとしたら、なぜ後者から前者への変換にこだわるのであろうか。もちろん、そうではない。メッセージ交換手法は、クランピングを通じてより大きな自由度を提供する。ライブデータ構



造体プログラムのプロセスは、1個の結果を計算して消滅する。それ以上に、別の結果を計算して提供することはできない。従って、メッセージ交換では、プロセスは好きなだけ結果を計算することができる。そして、それらを好きなときにメッセージとして送ることができる。プログラムが実行されている間、結果を作り続けることができる。ここにクランピングの意味がある。また、1個のメッセージ交換プロセスに、全部のライブデータ構造体プロセスの仕事をさせることもできる。

#### (4) アブストラクションとスペシャリゼーション

以上のように、いくつかの変換について述べたが、それらの利点は何であろうか。図3. 5-7では単純なネットワークの上で、いろいろなパスを選択することができる。それらのすべての詳細について今後の研究が必要である。例えば、結果並列法が最も自然に適用できそうな問題があったとしよう。これに対して適当なライブデータ構造体プログラムを書いたが性能は悪かったとする。そこで何らかの変換が必要になったとする。まず、なぜこの例を特に取り上げたのであろうか。問題が適当であれば、ライブデータ構造体プログラムは設計も簡単で表現も適切にできよう。また並列性も大きくなるであろう。もちろん並列性の程度は、結果の構造体の大きさと要素間の依存度による。しかし、ライブデータ構造体手法によるプログラムは往々にして、プロセスの数も多くなり、それぞれのプロセスは極く少しの処理しか行わないといった細粒度の並列性 (fine grain parallelism) になることが多い。したがって、最新の並列処理計算機の上では性能が悪くなりがちである。

例えば、結果としてデータ構造体の大きさが10,000要素もの配列になったとすれば、この方法では10,000ものプロセスが生まれることになる。もちろん、このようなプログラムを効率よく実行できないという理論的な理由はない。しかし、大部分の最近の並列処理計算機では、多数のプロセスを生成して協調させるにはオーバヘッドが大きすぎる。これは特に分散メモリ計算機では顕著であるが、軽量プロセスが実現できるとされている、共用メモリ計算機においてすら無視できないのが実情である。

せっかく並列処理をしても、多くのプロセスが細粒度の計算をする結果、並列処理の効果はなくなってしまふ。もしライブデータ構造体プログラムを効率よく実行できるのなら、それはそれでよいが、アブストラクションを使い、同じアルゴリズムを分散データ構造体プログラムに変換することによって、もっと効率のよいプログラムに容易に変換することができる。ライブデータ構造体を受動的な構造体で置き換えて、プロセスを概念レベルで1段上のものとする。それぞれのプロセスを単一の要素に対応させるのではなく、多くの要素の面倒をみるようにする。

例えば100個のプロセスを生成して、それぞれのプロセスに100個の要素を計算させる。その結果、元のプログラムよりも粒度の大きいプログラムとなる。どれだけのプロセスを生成するかについては、プログラマが適当な数を選ぶことができる。このようにして、多数のプロセスを作ることによるオーバヘッドを避けることができる。そのようにして作ったプログラムでもまだ十分に効率がよくないかもしれない。各プロセスから単一のデータ構造体に読み書きする。その時このデータ構造体は、どこかの共用メモリに格納しておかなければならない。共用メモリへのアクセスは、私用メモリへのアクセスよりも時間がかかる。最もこれは普通は問題なこともあるが、分散データ構造体が最適であることもいえる。

前述のn体問題について、ライブデータ構造体プログラムをアブストラクションやスペシャリゼーションの出発点とすることもできる。ライブデータ構造体プログラムは $mxq$ 個のプロセスを生成した。

それぞれのプロセスは関数positionを1回だけ起動して終了する。分散データ構造体プログラムをアブストラクションで生成してみる。Mを分散データ構造体とする。これは全プロセスがアクセスできる受動的なものである。0番目の列には各物体の初期位置を入れ、残りの列は空いている。k個のプロセスを生成する。それぞれのプロセスは、行列の一部を計算する。j番目の列を計算するには、プロセスはj-1番目の列に書いてある前の位置を参照する。プロセスの数はプログラマが制御できる。もしそれが適当ならば、2~3個のプロセスのみを使うのもよい。数台のプロセッサしかない場合である。このようにすればプロセス管理のオーバーヘッドを小さくできる。元のプログラムから、この新しいプログラムを作ることは簡単である。新しいプログラムは元のプログラムに比べると、多少詳しさと理解しやすさには欠けるが、最後にスペシャリゼーションによって、最もオーバーヘッドが小さいメッセージ交換プログラムを作ることができる。

それぞれのプロセスは、Mの一部を自分のローカル変数空間に格納する。Mは、もはや単一の構造体としては存在しない。プロセスは、最後に計算された位置情報に直接アクセスすることはできないので、これらの情報は、メッセージとして授受されなければならない。それぞれの繰り返しの後で、プロセスはメッセージを交換する。このメッセージには、プロセスが計算した最後の繰り返しの結果の位置情報を入れる。この結果、プロセス管理のオーバーヘッドも小さくて、分散データ構造体を参照するオーバーヘッドもなくなる。この最後のプログラムは、これまでのものよりもより複雑である。プロセスは各繰り返しごとにメッセージを授受して、ローカル変数を更新しなければならない。また、このプログラムでは重要な概念上の変更がある。先の2つのプログラムでは、通信はデータ構造体の参照という位置付けであった。これは他のプログラムでも同じような概念である。これに対して、最後のプログラムでは、通信はメッセージ交換という形態をとる。これは標準的なプログラミング手法に比べると、概念的に全く新しいものである。

アブストラクションとスペシャリゼーションをいつ使うかという選択は、プログラミングの現実的なものである。それは応用分野、プログラムスタイルおよび並列処理計算機などに依存する。例えば、CLinda言語を使って最新の並列処理計算機の上でプログラムすれば、メッセージ交換に帰結するスペシャリゼーションを利用することは、ほとんど必要ない。ほとんどの問題は、効率のよい分散データ構造体プログラムにできる。この場合には、効率のよいプログラムを得るためには、分散データ構造体のためのアブストラクションが必須であるということである。ライブデータ構造体を選ぶのが自然な場合、プロセスの数は多くてオーバーヘッドも大きくなる可能性がある。それを避けるために、アブストラクションによって分散データ構造体プログラムに変換する。メッセージ交換ネットワーク型のプログラムが自然であることもあるが、プロセスの数も多くプロセス間通信も頻繁になる可能性がある。この場合、アブストラクションによってメッセージパッシングから分散データ構造体に変換できる。

### 3. 8. 6 並列問題として地層処分性能評価

前節までの検討で、並列処理技術は単の高速計算を目指した技術ではなく、ソフトウェアの協調作業により対象となるモデルを組立て問題解決にあたるという新たな方法論に期待できることが、最も重要な関心事であることが解かった。また、並列処理モデル組立のパラダイムはオブジェクト指向を含む新しい情報処理モデルの概念により示されるものであるため、現在は研究中のパラダイムである

と考えるべきである。しかしながら、前節までに概観してきた、手順並列、結果並列、専門家並列等の情報処理の考え方は、地層処分ヴァーチャル・エンジニアリング・システムのアーキテクチャ設計の根幹となる技術に触れているものと考えられる。何故ならば、それらのパラダイムは非常に多くの新しい概念を含むものであるが、オブジェクト指向のパラダイムと同様に、多数プロセスとシナリオが混在している、地層処分性能評価手法開発に係わる情報処理面からの一つの可能性として期待できるためである。ジェネリック・アセスメントとしての地層処分性能評価は、その枠組みを組立て、各プロセスを数理モデルとしてプログラムし、それらを協調させて実行し、結果を得るということを目的とした活動である。それをコンピュータ内部に組み立てられた仮想処分システムについて実行できるようにするには、システムのアーキテクチャは、統合化手法として並列処理を取り入れることが必須となる。

#### (1) 結果並列法 : Result Parallelism

結果並列法の考え方は、モデルを構成要素の並列性から組み立てることである。例えば、現象全体を個別の現象結果を並列させて説明する場合が考えられる。この方法では、並列処理は最終結果となるデータ構造体について並列性を考える方法である。この方法を性能評価シミュレーションのアーキテクチャとして採用することを考えた場合、つぎのことが指摘できる。第一に、計算負荷の高い問題に対する速度向上による恩恵が期待できる。例えば、大規模3次元地下水流れ計算等では、問題を未知数の行列式等、計算結果の並列性の高い形で扱うことができるためである。従って、単体プログラムの計算速度の向上を計ることができるであろう。連成問題においても、従来開発されてきた連成数理モデルから導かれるプロセスを並列化するという意味で高速計算が期待できる。いずれにしても、結果並列法は、並列計算を考える場合の最も基本的な考え方であるが、これからのモデル化手法として期待できる協調計算技術開発を目指すというより、計算速度の向上を目指すべきである。

#### (2) 専門家並列法 : Specialist Parallelism

専門家並列法は、対象を処理の並列性から組み立てる。例えば、現象全体を個別の現象過程を並列に連成させて説明する。これは、問題がある種の論理的なネットワークで接続される専門家の集合として並列性を考える方法である。性能評価シミュレーションについて専門家並列法を考えると、それは、プロセス・システムの協調を計算を実現する手法として考えられる。プロセス・システムは性能評価の枠組みを決める考え方である。それは、注目する事象を個別のプロセスに分解して、連成させて考える方法である。この方法を用いると、性能評価シミュレーション全体のアーキテクチャ概念をプロセス協調計算として扱うことができる。このとき、個々のプロセス・システムは専門分野として、地下水流れ、地球化学、岩盤力学、温度分布等、個別のプロセスが相互に協調してシミュレーションを進めるとして考えることができる。それらのプロセスはメッセージ交換を行って、平行して計算を進めるであろう。そのようなプロセスをコーディングするにあたり重要なことは、メッセージ交換のインターフェイスである。例えば、温度場が状態と変化させたらそのフィードバックを岩盤力学プロセスは受けなければならない。そのためには、インターフェイスの統一や、メソッド、関数の仕様を統一する必要があると考えられる。

### (3) 手順並列法 : Agenda Parallelism

手順並列法は、対象と手順（シナリオ）の並列性から組み立てる。例えば、現象全体をそれに生じるシナリオを並列に組み合わせて説明する。これは、複数のタスク（エージェント等）の処理手順を基に並列性を考える方法である。この方法は、対象をネットワーク問題として考えることから始める。従って、地層処分性能評価では、FEP（Feature, Event, Process）から作成できるインフルエンス・ダイアグラムに見られるような処理を対象とするとき有効な並列化手法であると思われる。実際には、この手順並列法に前述の専門家並列法を混在させて、インフルエンス・ダイアグラムを基準としたプロセス・システムの状態遷移シミュレーション・アーキテクチャを考案することが理想的と思われる。

#### 3. 8. 7 試作並列シミュレーション機能

地層処分ヴァーチャル・エンジニアリング・システムのアーキテクチャを設計するための予備的な評価として、メッセージ交換による専門家並列機能を試作した。メッセージ交換仕様は、異機種ワークステーションとPCを混在させたネットワーク環境で、TCP/IPソケット、及びMPIを利用した。図3. 8-8にその概念を示す。

図3. 8-8に示すように、試作システムは2台のシリコングラフィックス社製ワークステーションと1台のサンマイクロ社製ワークステーション、及び2台のPCから構成した。これらのコンピュータは次のとおりに機能を（専門家として）分担している。第一に2台のシリコン・グラフィックス・ワークステーションであるが、一台は専用の3次元視覚化プロセッサとし、他方をシステム全体を操作するコンソール・プロセッサとした。また、サン・ワークステーションは専用のデータベース管理プロセッサとした。残る2台のPCはシミュレーションの進行状況を表示する、専用のパラメータ表示プロセッサと選択シナリオを表示するシナリオ表示プロセッサとした。なお、視覚化に利用したシミュレーションは、計算量等の条件から試験的に3次元の疑似的なシミュレーション・データを造りだすプログラムを開発して用いた。それにより、従来より開発されてきたFOTRAN言語によるシミュレーション・コードとのインターフェイスも合わせて検討した。図3. 8-9に視覚化プロセスとの協調概念を示す。図から解かるように、視覚化プロセスは専用のワークステーションにて稼働する。この視覚化プロセスは別のコンソールプロセス（システム全体を操作する命令を管理するプログラム）から起動される。

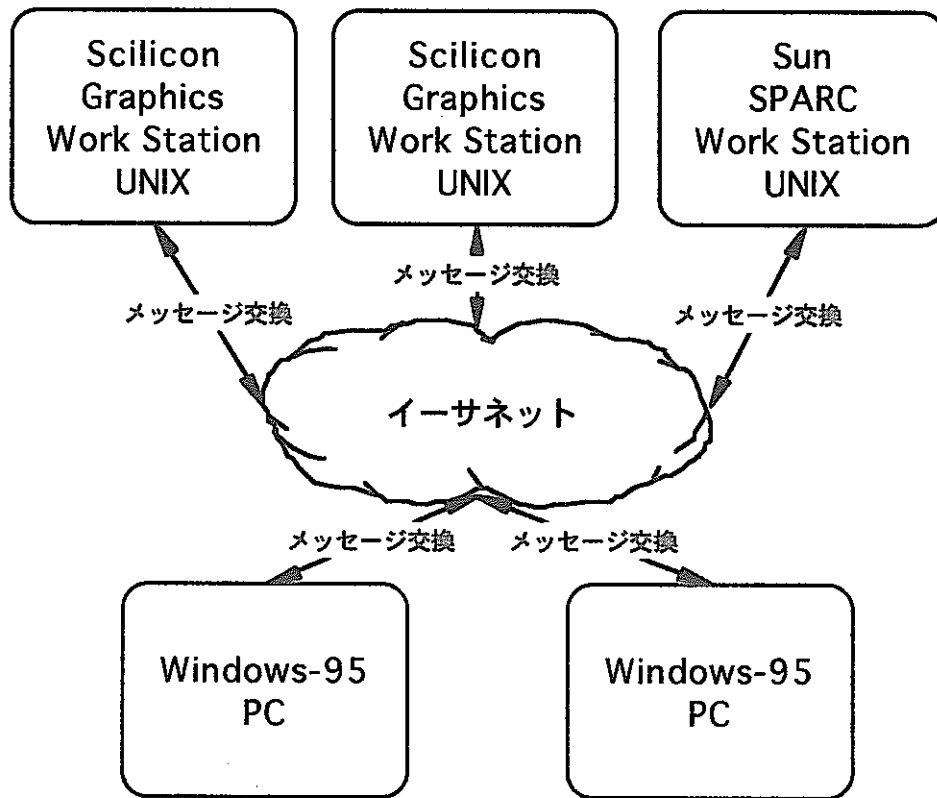


図 3. 8-8 試作システムのメッセージ交換による協調処理概念

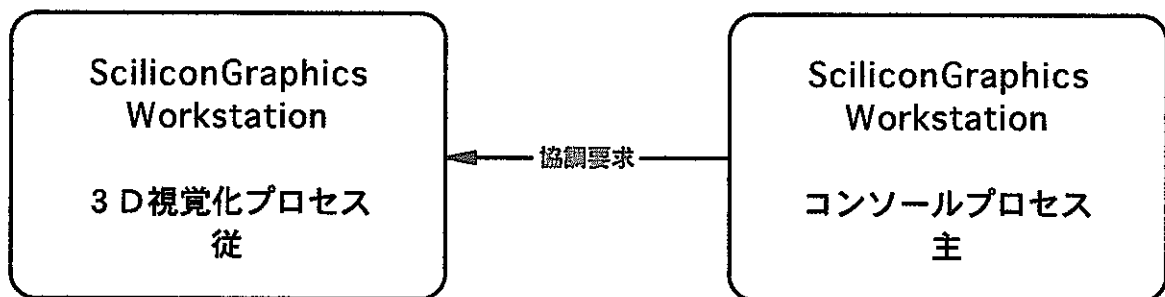


図 3. 8-9 3D視覚化プロセスの協調要求概念

協調の開始は、シミュレーションに係わる各種のパラメータ等をコンソール・システムと対話しながら決定し、シミュレーションの準備が整った時点で、コンソールプロセスがネットワークを通じて視覚化プロセスを起動し、並行動作を行うよう初期設定する。同様に、コンソールプロセスはシミュレーション・パラメータ表示用PCにも同様に初期設定メッセージを送信する。

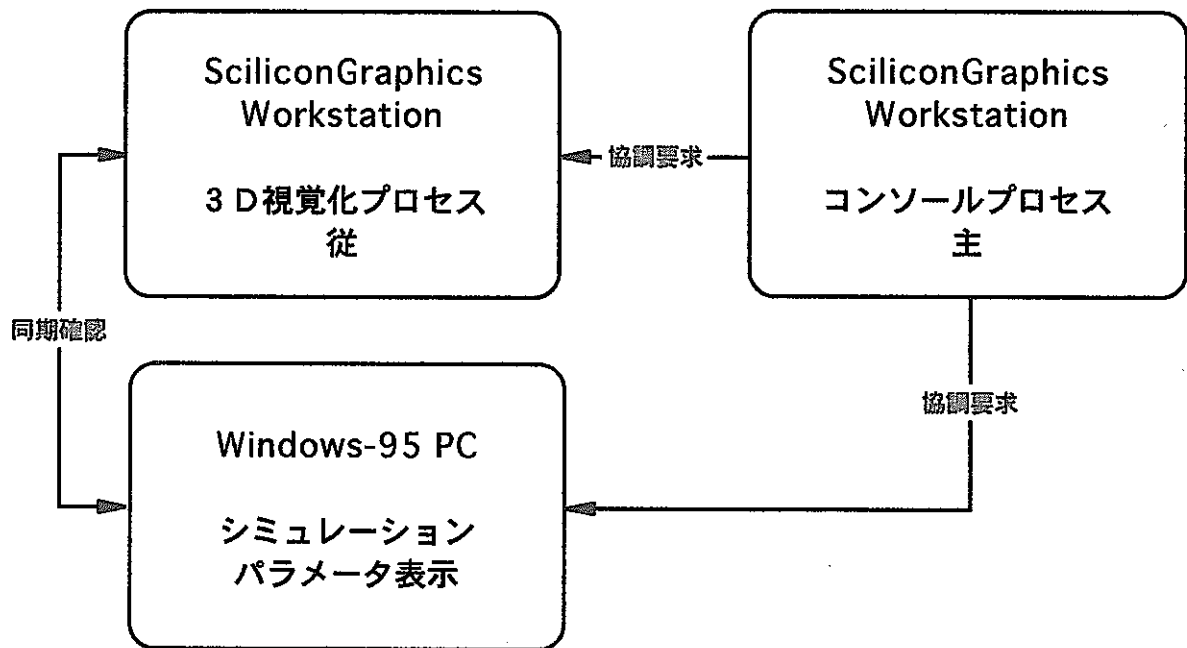


図 3. 8-10 PCプロセス協調要求概念

図 3. 8-10 にパラメータ表示プロセスへの協調要求概念を示す。パラメータ表示プロセスは、シミュレーションに係わる共通パラメータ（例えば、時刻、サイクル等）を表示するプロセスである。将来は、このプロセスが複数のモデルを協調させて計算を実行する主計時（マスターロック）として動作するよう研究する予定である。

図 3. 8-10 に示すように、コンソール・プロセスから要求された協調指令に沿って起動されたパラメータ表示プロセスは、3D視覚化プロセスと同期確認を行う。それにより、視覚化プロセスは、パラメータ表示プロセスの発信する時刻信号にそって動画再生を行う。今回の試作で利用したコンピュータは、計算能力が実時間動画化計算を実行するには充分でないため、計算結果を読みだして動画化することにした。その概念を 3. 8-11 に示す。

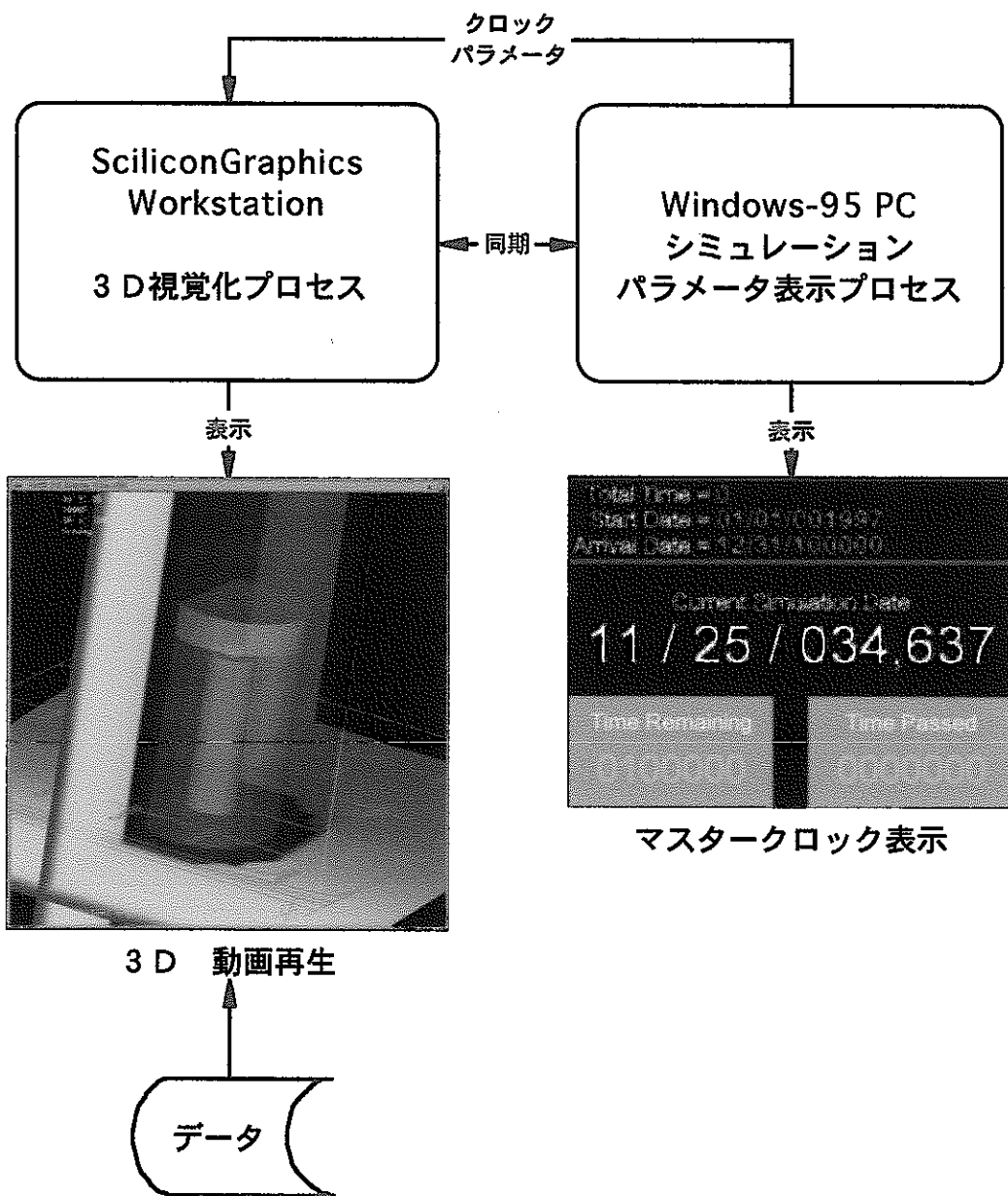


図3. 8-11 クロック・パラメータによる協調処理

図3. 8-11に示すように、試作したプログラムは、PCにより実行される主計時プロセスと3次元動画再現プログラムの協調により動作するものである。この機能の動作確認により、複数の異なったコンピュータに並列的に配置されたプログラムを、一つの主計時プロセスによって協調動作させる技術を確認した。

これにより、地層処分ヴァーチャル・エンジニアリング・システムの基本的なシミュレーション・アーキテクチャを、マスタークロック・プロセスにより、並列的に協調して動作する構成として設計することが可能であることが解かった。それは、図3. 8-12に示すような並列モデル協調計算アーキテクチャのプログラム可能性を示すものである。但し、そのようなアーキテクチャの開発

は従来の、FORTRANプログラムに比べ飛躍的に複雑なプログラミングを要求すると考えられる。

以上の検討から、並列したモデルの協調計算処理を考えることができる。このようなアーキテクチャによる連成問題へのアプローチも新しいシミュレーション手法として考えることができる。それは、恐らく複雑に関連するモデルを組み合わせ、視覚化シミュレーションを行う機能を実現する一つの可能性として期待することができる。そのためには、以下の課題についてより研究する必要がある。

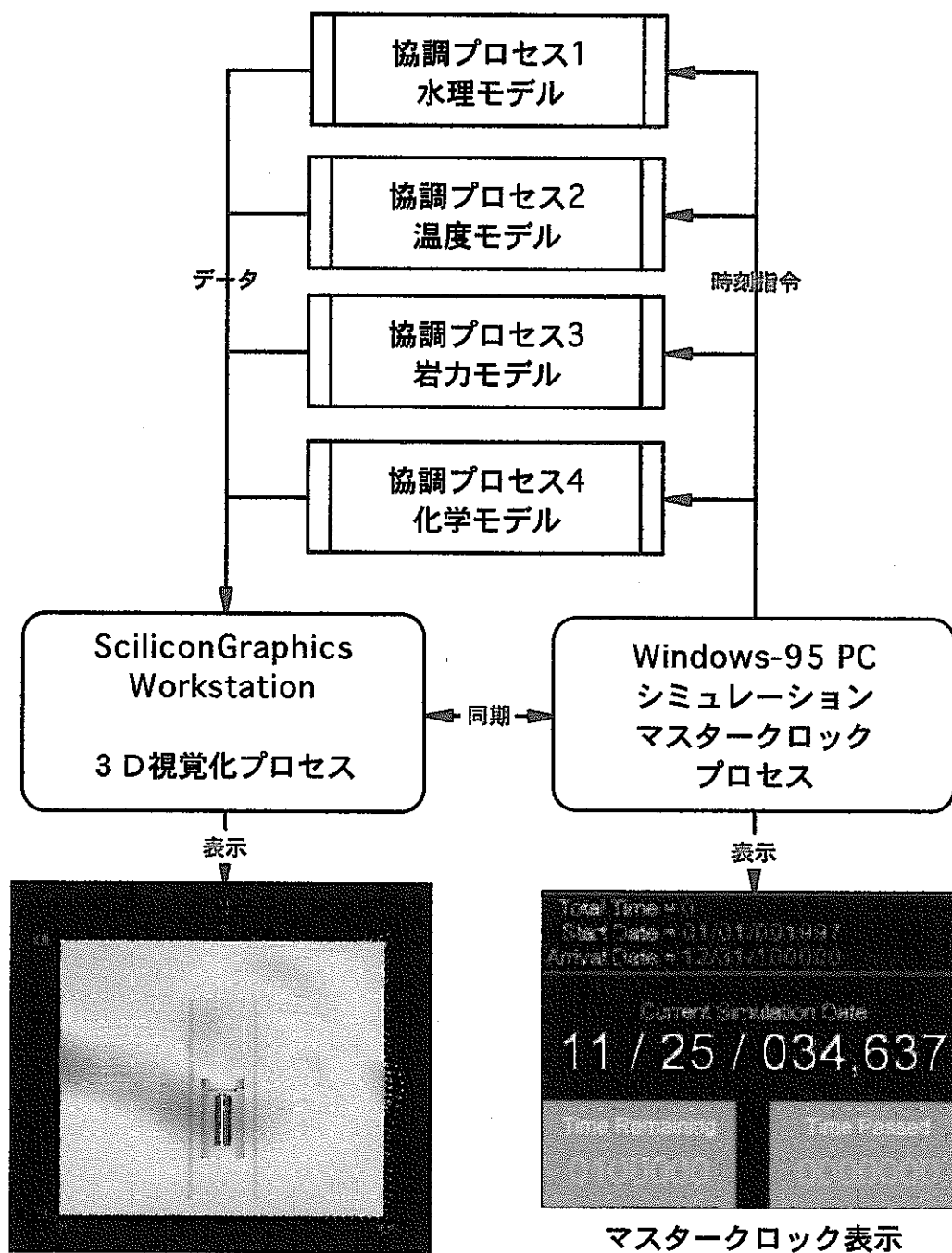


図 3. 8 - 1 2 並列モデル協調計算処理の概念



### (1) プログラミング技術の課題

協調計算のプログラミングは並列処理を基礎とするであろう。その時、並列処理は単なる計算速度の向上を狙ったものではなく、モデルとアルゴリズムの並列化により、直列計算（従来型）では扱えない（表現できない）問題を取り扱うことを主眼とするべきである。例えば、手順並列法、結果並列法、専門家並列法といった並列問題のモデル化について、その特徴を良く把握し、協調計算プログラムのアルゴリズム開発に応用する必要がある。また、並列協調計算を実現する情報処理モデルの基本的な考え方として、オブジェクト指向のモデル化を導入することも大切である。更に、将来の人的リスク等の問題を考えた場合、独立した判断によって動作するオブジェクトとしてのエージェントと言う概念を把握し、モデルに組み込む手法を検討することも大切であると考えられる。そのようにして、目標とするシミュレーション機能は、マルチ・エージェント/マルチ・オブジェクトな並列協調計算をマルチコンピュータで実現するものとしてアーキテクチャを設計するべきである。

### (2) 数理モデルの課題

並列協調計算を利用して、数理モデルを組み合わせた計算を実行することが考えられる。その時、個々のモデル空間座標は3次元であることが望ましいが、モデルとしては最も還元されたものを扱うことが良いと考えられる。何故ならば、自由度の高い連成モデルは分割できないため、協調計算には向かないと考えられるためである（それらは、むしろ単体で利用すべきであろう）。但し、協調計算においては、個々のコードが、計算サイクルや時刻は他のコンピュータ（試作システムではPC）のマスタークロックから受けること、連成させると物理量が繰り返し計算となるもの（例えば、圧力Pは、熱を連成させると密度の変化と勾配の釣り合いを解く繰り返し問題になるであろう）には、特別なメッセージ交換技術の開発が必要と考えられる。

### (3) 新しいパラダイムの受容

いずれにしても、本研究で取り上げているシミュレーション手法（マシンパラレル・モデル協調計算）は、全く新しいシミュレーションの考え方である。この考え方は、シミュレーションと捉えるより、仮想世界における数値実験と考えるべきである。そのような、新しい科学に対する考え方が受け入れられることが重要な課題である。

## 3.9 シミュレーション問題への新たな観点

高レベル放射性廃棄物地層処分プロジェクト全体からみた性能評価の位置づけは、設計建設された処分システムが計画通りに性能を発揮するかを予測する手法を開発し、それをもって確認することである。従って、性能評価と言った場合、あたかも工場から出てきた組み立てられたばかりの試作車を試験場に持ち込んで、総合的な安全性、経済性、運動性能等を評価する場合と同様な立場から多重バリア方式の処分システム性能を評価しなければならない。そのような観点からの問題の捉え方は、個別の設計論拠の確認に係わる現象解明やモデル開発という観点とは異なった、いわゆるジェネリック・アセスメントの手法開発を要求する。

例えば、前述の試作車の場合、エンジンのシリンダから予測できそうな限界は、そのエンジンの凡その出力であり、車両全体の構造に係わる情報がなければ、運動性能を予測することは不可能である。

同様に、地層処分システムの性能も、個別の現象（例えば、地下水の地球化学的特性）単体から全体を予測評価する枠組みを組み立てることはできないと考えられる。そのように、地層処分の技術開発に係わる重要な観点は、処分システムの設計建設に係わる個別要素評価手法は、参考になる技術を提供すると考えられるが、そのままでは、性能評価の一現象説明手段としてしか利用できないため、処分システムの設計に係ると技術と、性能評価に係る技術を明確に分離して開発していくべきである。そのような観点から、性能評価においては、その枠組み（アセスメント・フレーム）を決めるための手法としてシナリオ開発があげられる。シナリオの開発は、計画する処分システムが計画通りに性能を発揮するか確認するために必要となる、支配的な事象とその関連を論理付けることである。そのための科学技術的方法がシナリオ開発法であり、確率論的手法（PSA：Probabilistic Safety Assessment）、文章（FEP：Feature, Event, Process）、現象の関連（PS：Process System）等、幾つかの手法が提案されている。

また、シナリオの開発においても、前述の設計と評価における問題の混合の様に、施工シナリオと評価シナリオは分離して考えなければならない。もちろん両者の関連はあるが、性能評価の枠組みを決定するにあたり、最も単純な枠組みから出発し、評価シナリオと施工シナリオ、操業シナリオ等を混乱させないことが大切である。例えば、建設施工時のシナリオとして、発破による掘削影響、ボアホール等により水理が変化することが知られているが、これは建設施工シナリオとして安全性設計課題として扱うべきである。また、操業時に生じるリスク（落盤による坑道の封鎖等）も分離して考えるべきである。更に、性能評価の枠組みに加える必要はあると考えられる、断層、地震等、確率的かつ動的なシナリオ（変動シナリオ）も分離し、当初は静的な地下水による核種漏洩と移行シナリオを取り上げることが基本的アプローチである。

#### （１）評価フレーム造りの難しさ

以上のように、性能評価シナリオを単純化しても、対象とする処分システムの性能を評価する枠組みは非常に複雑なものとなるであろう。最低限のパラメータを見積っても、それは地下水の水理、地球化学的性質、および廃棄体の発する熱の影響から生じる地下水二相流れや、岩盤応力変化、破砕体への影響による浸透流の変化に伴う化学的遅延効果の変化等、非常に複雑で再帰的な（フィードバックのある）現象を扱わなければならない。さらに、評価においては、注目する現象が、これらの個別現象の連成現象であることが当初からの課題である。従って、地層処分性能評価のアプローチの難しさは、始めから連成した問題として対象を考察し、モデル化しなければならないことである。このことは、地層処分性能評価と言う問題が、個別現象解析と異なった、本質的な難しさと、現象理解の方法論の相違を内在していることを示すものである。以上の観点から、本項では、そのような複雑な系の理解を、従来の要素還元論に基づく数理解析的手法とは別な考え方として、コンピュータ内部で複雑な仮想世界を組立て、数値実験する考え方を予備的に検討し、今後の地層処分ヴァーチャル・エンジニアリング・システム開発における、性能評価シミュレーションの考え方の参考とする。

#### （２）地層処分ヴァーチャル・エンジニアリングにおける性能評価シミュレーションの考え方

従来のシミュレーション・コードの開発手法は、一般に複雑な物理現象等を簡単な過程に分割し、自由度を下げ支配方程式を簡略化することにより、計算可能な数値計算モデルを組立てプログラミン

グする方法がとられてきた。それらのコードは、研究開発現場に於ける個々の事象評価用ツール等として開発され、安全性研究等の成果を上げてきた。従って、従来のシミュレーション・コードの開発は、“一般的な工学的手法として普及している要素還元論的方法に方法の基礎を置いており、分解された個別過程の専門的な現象の領域について、精密に記述されたモデルに理論の基礎を置いている”と考えられる。また、このような決定論的方法とは別に、確率論的安全解析等の全体シナリオが評価の戦略となるような領域については、統計的手法を応用した研究が進められてきた。それは、核燃料サイクル施設の故障率等から事故発生確率や猶予時間を推定する、いわゆる事象の起因に関する評価手法であり、発生事象の規模評価となる決定論的手法としての、個別現象評価の必要性の論拠を導くための手法として位置付けられていると考えられる。

以上の観点から、従来のシミュレーション・コード（数理モデル）開発手法は、個別の物理化学過程を説明する知識としての成果であると考えられる。しかし、性能評価問題のような複雑で大規模な統合化シミュレーションを具体化するためには、物理現象の解明に係わる情報処理技術とは別な新たなソフトウェア工学分野の研究開発課題が生じると考えられる。なぜならば、計画する性能評価シミュレーションの手段としての計算科学技術の具体的な姿は、コンピュータ、ネットワーク、グラフィックス、データベース等、様々な情報処理技術の領域で実現されるものであり、物理現象等の解明の成否は、直接ソフトウェア工学の影響を受けると考えられるためである。

また、地層処分ヴァーチャル・エンジニアリングは、実験の場を単純にハードウェアからソフトウェアに移行するという考えではなく、新しいソフトサイエンスの領域の学問を築く活動と捉えるべきである。それは、従来の還元論的な物理科学の方法論と、高価で柔軟性に欠ける実験装置という脇役にとって代わる21世紀の科学の戦略と考えるべきである。そのような観点から、地層処分ヴァーチャル・エンジニアリング技術の領域は、従来のシミュレーションと呼ばれる、個別の数値計算ソフトウェア開発技術の領域と明確に区別して考えるべきである。当然、従来通りの手法により、個別現象解明のための技術の開発は、今後も推進するべきであり、それらは地層処分ヴァーチャル・エンジニアリングの推進に多くの基礎基盤的貢献が期待できるものである。

一般に、研究活動の目的は知識の生産である。実験は、研究という知的活動を進めるために行なう外界からのデータ収集である。研究者は、自己の知験や直感とデータを照らし合せて新しい知識を生成する。そのようにして、理論が組み立てられていく過程は、要素還元と逆の統合の過程である。この統合の過程は、還元の過程に比較し複雑であり、未知の領域を含む科学の方法として考えられる。そのような観点から、地層処分ヴァーチャル・エンジニアリング技術は、旧来の方法論に立脚した視点からの改善案ではなく、新しい科学技術の戦略として捉えるべきである。例えば、地層処分ヴァーチャル・エンジニアリングは、実験の困難な対象をソフトウェアに置換するといった単なるシミュレーション要求ではなく、より複雑な現象を総合的に解明するための知識の統合化アプローチを考えるべきである。

### (3) 新しい考え方（数値実験装置としてのコンピュータ・システム）

地層処分ヴァーチャル・エンジニアリングにおける性能評価シミュレーションの基本戦略は、物理現象という科学の領域について、それを個別の、より簡単な過程に分解して得られた数理モデルを数値的に解くために開発された、解析コード等と呼ばれるプログラムを統合化し、協調して計算させ、

処分システムをとりまく現象を長期に亙り総合的に評価することである。この際、具体的な研究の範囲は、大規模な統合化シミュレーション・システムを開発する際に必要となる、様々なソフトウェアの統合（Software Integration）に関する開発と管理の手法である。例えば、良く知られた非線形有限要素法は、シミュレーション手法として成功してきた手法である。それは、対象となる現象過程を工学モデルとして数値的に評価し、設計対象の安全性等を判断する研究などに利用されている。それは、一つの工学的手段として、研究開発現場で受け入れられた数値計算ツールと捉えることができる。

このような場合、解析コードに対する研究開発現場の関心は、研究開発の核心に触れた対象（すなわち現象そのもの）である。例えば、それは評価すべき工学モデルと、それを評価するための解析コード自身の基礎方程式、各種の計算条件を構成する方程式、そして、解の工学レベルでの検証・確認である。また、解析コードの品質（検証・確認）は、研究室の試験設備や、大規模な実験設備等から得られた試験結果等との比較により判断されることが通例であった。対象とする解析コードが、与えられた条件のもとで、試験結果を模擬すると判断される数値を出力した事により確認される。そのような判断は、解析コードのツールとしての妥当性を外部から判断する方法と考えられる。

以上のように、研究の対象が、個別の専門的な現象領域である場合は、そのような判断基準によって、解析コードのツールとしての妥当性を評価することで、十分に工学レベルの実用に供したと考えられる。さらに、それらのコードは、一般的にブラック・ボックスとして利用されてきており、プログラミングに関する細々とした内容は、本質的な議論の対象にはされなかったと考えられる。

例えば、開発したコードの出力する数値が、試験結果をどれほど正しく模擬するかについては、多くの関心が寄せられてきたと思われるが、プログラマが領域変数を用いて反復計算をするアルゴリズムを導く時、移植性を考慮したコーディングを採用しているのか、それともベクトル計算機のような特別のアーキテクチャの機能を引き出そうとしたコーディングを採用しているのか、または単に動作することのみに関心をもっていたのか、等の計算論の核心をついた部分についてはあまり多くの関心は払われなかったと思われる。

#### （４）ソフトウェアのスケールファクターに伴う問題

以上のように、複雑な現象過程をコンピュータによる仮想世界で再現する試みを期待すると、技術の領域に情報科学が重要な位置を占めてくる。それは、コンピュータがプログラム言語という方法を唯一の動作手段としていること、また、コンピュータは工業生産品であり、メーカー固有のアーキテクチャがその動作と不可分であること、さらに、どれほどコンピュータが高速になっても、数学的には近似計算しかできないこと、プログラムは大規模複雑化するほどバグ（誤動作）も増えると言った問題を引き出すであろう。

これは、単一目的のソフトウェア開発から、大規模な統合化ソフトウェアの開発に移行する際にソフトウェアのスケールファクターの影響として生じてきた領域の問題であると考えられる。それはまた、過去のソフトウェア開発に内在していた問題が、大規模ソフトウェアの統合化を目指す現在のプロジェクトの課題として表面化してきたものである。これらを整理すると、地層処分ヴァーチャル・エンジニアリング・システムの性能評価シミュレーション・アーキテクチャを開発するために考慮しなければならない、ソフトウェア科学・工学の領域は大別して以下の通りとなる。

### 分野1：ソフトウェア統合化のための理論分野の研究領域

大規模な統合化シミュレーション・システムを開発するための方法論は、個別の専門領域に対象を分割し、自由度を減少させることにより、方程式等を簡素化してプログラミングを容易にし、それら個別のプログラムを統合して、多方面からシミュレーションし、最終的に支配的なモデルを導き出すと言う、決定論的シナリオによる方法論の妥当性を研究する領域である。

それはまた、確率論的方法との比較、連結、そして、複雑系モデルとして、トップダウンとボトムアップの間に複数の計算モデルがコンピュータの中の仮想世界で協調して計算を進めるという考え方を位置づけ研究する領域が重要と考えられる。それらは、ソフトウェアの構造的な手法の研究、ソフトウェアの複雑度の影響領域の研究、数値計算の信頼性に関する研究、論理処理、記号処理のシミュレーションへの応用、モデル協調計算手法に関する研究、カオス、複雑系、進化計算学の研究等である。

### 分野2：ソフトウェア統合化のための工学分野の研究領域

大規模な統合化シミュレーション・システムを開発するための工学的な手法として、次の領域が研究対象として上げられる。それらは、ソフトウェアのライフサイクル管理手法、ソフトウェアの品質管理手法、マルチ・コンピュータ・シングル・システム・アーキテクチャの設計開発法、システムの機能を高度化する知識情報処理や認知科学の応用研究等である。

#### 3. 9. 1 複雑な現象と複雑なモデルとして捉える方法

地層処分システムの性能評価のような問題は、社会モデル、経済モデル、一般環境モデル等、各種存在している。これらの問題は、一般に大規模システム (Large Scale System) や複雑系と呼ばれている。反対に、単純モデルに還元された物理問題等を単純系と呼ぶ。例えば、複雑系は日常生活の至る所に顔を出し、郡市の交通網とか株式市場における価格変動とも関係がある。しかもそこに見られる複雑なプロセスは、単純系とは根本的に異なるものである。ふつう単純系は、相互作用の弱い少数の要素からできているか、もしくは逆に、容器内の気体や銀河系のように非常に多くの要素からできているか、後者の場合、その振る舞いを研究するには統計的平均化という手法が使われる。

これに対して一般の複雑系は、車両のドライバーやディーラーといったエージェント (系を構成する行為者) からなるが、その要素の数は中規模である。ここで注意しなければならないことは、一般のエージェントとして扱われる対象は、自律行為者であることから、本研究が目指している現象モデルとは多少異なることである。従って、特にそのような物理化学現象 (物理化学的な振る舞い) を自律的に再現するオブジェクトを現象エージェントと呼んで区別する。地層処分性能評価シミュレーションの狙いは、この現象エージェントの協調計算を開発することである。

そこで、複雑系におけるエージェントは、グローバル (全体的) な情報ではなく、ローカル (局地的) な情報に基づいて決断を下し、自らの行動ルールを更新していくことになる。こうして、複雑系、すなわちローカルな情報に基づいて行動する適応能力のある中規模のエージェントの集合は、今まで科学が築き上げてきた単純系とはあまりにも異なっている。つまり複雑系の研究は、自然界や人間社会を科学的に探求する上で新しい考え方を与えると思われる。科学で最も敬意が払われる原理の一つに、いわゆる“科学的”手法がある。この方法で欠くことができないのは、“再現可能な”実験と

いう考え方である。仮説を確かめるためには実験を行い、実験を行う研究室がなければ、科学の理論は生まれてこない。実験室には、様々なものが考えられる。

例えば、化学実験室には、試験管、蒸留器等が完備しており、場合によっては、原子核を調べるために粒子加速器が置かれることもある。こうした実験室は、ごく普通に見かけるものだが、いずれも単純系の物質構造を研究するために使われてきた。一方、複雑系の情報理論的研究のための実験室として、研究者たちはコンピュータの内部に現実世界の複雑系を模擬した仮想世界を創造することを考える（ヴァーチャル・ラボラトリ）。それによって、現実に近い形で再現可能な実験が期待できる。

そのようにして、複雑系を探求する科学者たちは、ガリレオの時代に生きた物理学者と全く同じ立場にいる。そのようにして、どんな状況に対しても、コンピュータ内部に様々な世界を創り出せる能力を人間は持っていると考えることが第一歩である。コンピュータを使えば、現実の複雑系に対して無数のシミュレーションを納得するまで遂行でき、対象となる問題領域を単純なサブシステムに分割して考える必要などない。

### (1) カオス、影響、感度、モデル化戦略

地層処分性能評価では、特に事象と事象の影響、感度が問題となる。このような問題は、対象を連成した現象エージェントとしてとらえ、ダイナミックに評価する課題を生じる。特に、ソフトエウア工学（気象モデルの離散方程式のプログラミング）から発見されたカオスは、その代表的な例である。同様に、非線形系は必ずしも構造安定ではない。つまり、モデルを少し変えたことにより、結果が大きく変わりうるということが数値計算の分野で知られている。このことは、処分システムの性能評価でカオスを考えるべきかは別として、カオス系でモデルのパラメータの値をわずかに変えた場合や、モデルの表現のしかた（変数の選び方）をわずかに変更した場合に、その変化に対する敏感な依存性として成り立つということである。ここで大切なことは、そのような疑問（処分システムを取り巻く環境がカオスのように変化する可能性があるのか？という疑問）が出されても、複雑系のモデルを基盤とした、マルチ現象エージェントによる協調計算法は対応できるということである。

さらに、モデルの示す振舞いが、観測に対して敏感に変化し不安定であることもある（このような性質を記述不安定性と呼ぶ）。記述不安定性は、自然現象のモデル化の在り方に変革を迫るであろう。それは、「最も信頼しうるミクロの方程式から出発し、それを適度に粗視化したマクロな方程式を導き、最良と思われる近似を使ってモデルを導出する」という従来の立場は、もしモデルが定量的に記述不安定であれば、その立脚点を失ってしまうであろう。つまり、“真のモデル”に近いほど、定量的によいという信念の根本にあるモデル間の滑らかな“距離”というものが存在しないかもしれないからである。その場合でも、モデルは定性的には正しい結果を与えることがあるであろう。しかし定性的レベルの議論においては、もっと有力なモデル化が有りうる。それが構成論的に仮想世界をつくりあげる手法である。このような方法を通してのみ、あるクラスの複雑現象が、自然の様々なレベルで共通に存在するのを知ることができ、また、どんなクラスの系がどのような現象を導くのかを予測することができると考えられる。こういった仮想世界の構築によるアプローチは、特に地層処分性能評価のような、歴史性という要素の入った現象の理解には欠かせないのではないかと考えられる。歴史的に、一つのパスを通して起こってきたような現象は、再現性を重んじる従来の科学の方法論では扱えないのではないのかと言う議論がなされてきた。例えば進化である。これに対し、人工的な世

界をつくることは、有りえたかもしれないパスを眺観していくことを可能にするので、進化や脳のよ  
うな発展、過去の気象等、変化する系の問題への新しい切口を開きうるのではないだろうかと考えら  
れる。以上のように、複雑系の理解においては次のような特徴があるように思われる。

現象により密着してたてられたモデルが、必ずしも普遍的記述力をもつとは限らない。  
逆に現実からやや距離をおいたモデルが、大きな記述力をもちうるであろう。  
そこで、仮想世界を計算機の中に構成し、その仮想世界を研究することで、逆により  
よく現実の系を理解できると考えられる。この特性は、仮想世界と現実との関係を、  
いかにとらえるかが重要な課題となってくる。

計算機科学は、ヴァーチャル・リアリティを「現実」のものとした。一方、カオスを計算機で計算  
することは原理的に不可能である。カオスを特徴づけているのは、非加算無限個の非周期解と加算無  
限個の周期解である。一方、計算機は有限オートマトンであるので、どんな無限性も計算できない。  
計算結果は結局は周期解になってしまう（ポアンカレの回帰定理：周期は非常に長いこともある）。  
さらに多くのカオス系では、個々の軌道の近似計算ができないことが知られている。微分方程式や差  
分方程式のような偶然性をいっさい含まない決定論的な方程式は、初期条件を与えれば、以後の解軌  
道を完全に決定するはずである。ニュートン力学でよく知られているように、運動方程式と初期条件  
は論理的に独立である。しかし、方程式がカオスを生成する場合、軌道の初期条件鋭敏性により、軌  
道に関する将来の予測は不可能になる。この状態において、運動方程式と初期条件は観測を通じて相  
互依存することになる。つまり、カオスは両者の間には、もはや観測の仕方によって、きわめて敏感  
にその状態を変えるので絶対的な因果関係は認められない。以上の議論は、複雑な物理現象をコンピ  
ュータ・シミュレーションに置き換える時の本質的な問題を提示する。それは、現象と数理モデル、  
そしてプログラムという、異なった世界に等価な論理を持ち込もうと努力する活動に対する基本的な  
前提条件である。

## (2) ソフトウェア工学面からみた問題

前述してきたように、一般に研究活動におけるシミュレーション・プログラム開発の狙いは、仮説  
として組立たモデルについて、実験等の外界から得られた情報に基づき、評価や検証を行なうための  
手段として、コンピュータを利用しモデルの数値による近似解を得ることである。これは、経験から  
得られた知識を文字によって記録すること、更にそれを発展させて数字と数学記号により、より精密  
な量と変化の割り合い等を記述する活動の次に現われた知識の記述戦略と考えられる。

シミュレーション・プログラムの開発においては、複雑な物理化学現象を簡単な過程に分解し、そ  
れら分解した過程について、個別数学モデル（方程式）を準備し、それを数値的に近似計算するプロ  
グラムを開発するという手順がとられてきた。コンピュータの基本動作は2進法による論理演算であ  
るため、実際にはモデル式を数値的に近似する離散式を導き、その離散式からコンピュータが実際に  
動作可能な2進の命令と呼ばれる一連の規則をディジタル信号に変換するための操作が必要になる。  
そのための手段としてプログラミング言語があり、計算機械としてのコンピュータがある。従って、  
開発されてきた個別の解析コードは、開発段階では統合化を想定して計画されたものではないと言え  
る。そのことは、開発された様々な計算コードを組み合わせることにより、現象全体を模擬すること

が技術的に可能か？そのためには、従来の計算コードは再利用可能か？どのように組み合わせれば良いのか？また、たとえ組み合わせることができても、それが妥当な選択であるのか？コードの組み合わせから生じるソフトウェア全体の複雑度の増加はどのような問題を生じさせるか？等、避けて通れない情報科学の分野の課題が潜在している考えられる。

また、開発した解析コードの出力する数値が、工学試験結果をどれほど模擬するかという疑問についての比較検証や、検証された解析コードの出力する数値と、新たに開発、もしくは改良整備したコードの出力する数値を比較するクロス・チェックにしても、判断の基準は工学試験結果であるが、対象としているコードの本質は、形式的なプログラミング言語により記述された規則であり、その本質は、言語とアルゴリズムという抽象的な哲学的原理という実験とは別な領域に出发点をおいている。それは、ソフト・サイエンスの領域に本来属するものであると考えられる。その様な観点からは、与えられたコードが、試験結果を上手に模擬する数値を吐き出すと言われても、それがソフトウェアとして高い品質を保持しているものなのか？また、それが計算する範囲のモデルの普遍性はどのようなのか？将来的に他のモデルを計算するコードと組み合わせ利用できる可能性があるのか？という質問に対する明確な回答は見当たらない。

従って、従来の解析コード等のソフトウェア開発は、個別の専門領域の要求から発生し、開発されたコードの妥当性は、それ自身の動作結果として生じるアウトプット（一般に数値）を工学試験結果等と比較検証すること、いわば、外部からの評価で、その信頼性、利用可能性等の妥当性が判断されてきたと考え、内部からのソフトウェア自身としての評価については、その手法等、未知の領域が残されている。それはまた、数理モデルとコンピュータ・プログラムは異なるものであることを認識する重要な観点である。

### （3）複雑な数値計算（実験）における創発という概念

私達が複雑な系に求めていることの一つに、系に予め人為的に埋め込まれていない性質が現れてくることがあげられる。このような自発的な生成に対して、最近、創発という言葉が使われ始めた。創発に関する明確な定義はまだない。しかし次のように考えることはできるだろう。現れた現象を説明する為にはそのプロセスを追っていくしかない、つまり、問題設定だけから、より短い説明を行うような情報の圧縮ができないということは、その必要条件である。創発という概念を最初に現代的な意味で用いたのはボランニーである。現在、この概念が広く使われる理由は、トップダウン対ボトムアップという、人工知能や神経回路網などでの論争を超えたいということと思われる

そのような複雑で大規模なシステム（地層処分システムの環境中挙動もそのような対象である）の評価を行う場合、計画したシナリオから、自発的な事象の生成等が考えられる。そのような場合、創発という概念は、不確定性や感度とともに重要な概念となるであろう。これらの分野の議論では、「上」（トップ）のレベルは少数の自由度で記述され、「下」（ボトム）のレベルは、非常に多数の自由度が関係しているという前提が置かれているように思われる。ボトムアップでは、「上」の性質は「下」のレベルから、一種の「秩序パラメーター」として生成され、それが巨視的な振舞いを決めるとしている。逆にトップダウンでは、「上」のレベルからの少数の指令が「下」の振舞いを決めるとしているわけである。創発に多く期待されているのは、この両者の間の流れを双方向にするというものであると思われる。創発という言葉には双方向の流れを通して、まえもってプログラムに「入



れておかなかった」ことが、どちらのレベルに還元されることなく、自発的に生成されるのではないかという期待が込められていように思われる。

それでは、この自発的生成とはいかなる意味であろうか。この文脈で、カオスを含め、あらゆるもののシミュレーションに見られるように、コンピュータにプログラムを組んで走らせた結果に本当に創発が見えるのであろうか。また、プログラムを組んだ段階では予期できなかったことが、シミュレーションの結果として現れることがしばしばある。そこで、これを創発と呼びたくなるのではあるが、これは我々の認識が、生成に対してプログラムのレベルでは働かなかったことを意味しているだけである。さらに、有限状態の世界の中で、有限ステップの中で現れることで、プログラムに予め「入れておかなかった」ものが生成されることは論理的に有りえない。

そこで一つの可能性は、無限のテープや状態を持ったマシンの導入である。無限のテープを利用できる可能性を持っていて、時間も可能性として十分長くとれるのであれば、プログラムの実行は無限に続く可能性があり、それはチューリングの停止問題に見られるように決定不能である。その結果、プログラムに書き込んだこと以上のことは出ないとは、必ずしもいえないと思われる。そこに創発の一つの可能性がみられるのである。さらに、テープが連続無限の状態を持っていたらどうなるのであろうか。これは、実数計算を真正面からとりあげる研究によって考えられるであろう。そのように、実数計算論と従来のデジタルな計算理論との関係を明らかにすることにより、後者に基づいた形式的体系では基盤づけが困難な創発への道が開かれる可能性があると考えられる。

この他、よく議論される無限極限の一つの例として、個数を無限にした極限がある。統計力学における熱力学極限がその例である。この場合、個数を無限にする極限を先にとってしまうことで、多くの性質を導き出すのである。たとえば、熱平衡系の相転移の問題では、この極限により特異性が現れることが知られている。確かに、それによって、ゆらぎの増大など興味深い性質が現れるが、こういった性質は少数自由度での記述が可能であり、それ故、目指そうとしている「単純な記述には落ちない問題」というクラスには属していない。

ここで注目したいのは、カオスのもつ根源的なコントロール不可能性である。はじめに述べたように、カオスでは、力学系を駆動するためのプログラムが用意されているというところから出発したにもかかわらず、結果は、そのプログラムではコントロールできずに、「確率」を導入せざるを得なくなったわけである。さらに、カオスは「情報」を生成することができるのである。そこで再度強調すれば、問題にしている系にカオスが含まれていれば、プログラムで書いておいた記述以上に複雑な振舞いが原理的に現れる可能性がある。ただし、これは力学系としての創発への必要条件にすぎないであろう。

創発を捉えるには、さらに内部からの観測主体をふまえた考察が必要になってくる。これは今後の重要な課題である。カオスが内在しているような要素間の連鎖や、カオスを用いた計算理論、特に実数（ないし実数時間）計算理論は創発を考える上でも重要となっていこう。一方、我々は、カオスをかなり満足のいく程度にコンピュータで計算することができる。これは、状態数が有限といえども、大きければ無限の“影”を見れるということを示唆している。そこで、何らかの意味で、無限の彼方に創発を起こしうるような系があれば、我々のもつ計算機でも“創発の影”を十分見てとれる可能性があることになると思われる。

以上のように、大規模で複雑なモデルをコンピュータで扱うことを考えた場合、創発という概念が

無視できなくなる。それは、1セットの支配方程式から出発した旧来の計算コードと呼ばれる、単純な計算手続きから構成される計算と、前章でみたような複雑な並列計算の間では、計算量も当然であるが、モデル自律度と複雑度はかけ離れている（より実数計算論に近い？）と考えられる。また、カオスは大気モデルから観測されたように、有限状態の計算論の中でもそれを見る可能性があると考えられる。従って、地層処分ヴァーチャル・エンジニアリングに於ても、開発を進めることにより“創発の影”を将来見る可能性を期待できると考えられる。

#### (4) 複雑モデルの関係性のダイナミクス

ここで述べる多くの要素の間関係性のダイナミクスの研究は、トップダウンやボトムアップなどとは異なる新しいアプローチによって基礎づけられる。例えば、人工知能研究に典型的に見られるトップダウン・アプローチでは、上のレベルでの指令によって、下のレベルの振舞いは決められている。ボトムアップ・アプローチでは、そういった上からの指令なしで、下の要素間の相互作用によって、上のレベルが生成されることに重点を置いている。特にこのアプローチでは、トップダウン・アプローチにみられる上のレベルの硬さを解消し、上のレベルを自己組織しようとすることに特徴がある。しかし、この場合、上のレベルが自己組織化されると、一般には、要素の関係は固定されてしまう。すなわち、上のレベルが一度できあがれば系全体の状態は固定されてしまう。この意味において、ボトムアップ・アプローチは多くの要素からの秩序パラメーターの生成という、伝統的な統計力学の枠組みを脱却していないと思われる。

この二つのアプローチのそれぞれの欠点を解消するために、次の段階として上と下のレベルの間にフィードバックを入れるという考え方がでてくる。しかし、このようなフィードバックを入れたアプローチも、複雑系のアプローチとしては多くの点で不十分であると思われる。その意味で下のレベルの要素の自発性はなく、設定された環境によって決まっているといえる。さらにいくつかの問題点がある。要素の間関係は基本的に固定されているという点、要素は一様な性質を常に保ち続けているという点などである。ここで扱おうとするアプローチの特徴は以下のようなものである。それは、そのまま複雑系の特徴を表している。

- ・上のレベルは必ずしも少数自由度で規定されていない。  
上のレベルが存在するか否かといった知識は必ずしも始めから与えられていない。
- ・下のレベルの各要素は自発的な運動を起こす。
- ・外からの境界条件を仮に与えたとしても、各要素の運動は予測/制御されていない。
- ・各要素の関係自体が変動しうる。

このように見えてくると、上と下のレベルなど存在しないとした方がすっきりするように思われるかもしれない。このことは、どんな小さな差も巨視的なレベルへとつながっていくという、カオスのもつミクロとマクロの分離不可能性の当然な帰結であると思われる。

明白にどこが上のレベルであるという境界が、定まった箇所に入れられる問題が分離不能性の特質である。ここで、あえて上下のレベルという言葉を用いたのは、そのように捉えた方が分かりやすい瞬間や状態が数多くあり、その場合には、そういったレベルの生成、崩壊、競合、共存、階層性を見ていく方が理解しやすいことがあるからである。さらには、あえてレベルを区別する境界を導入して

矛盾を導くことで、レベルの不在性が示せる場合もあるであろう。ここまでの議論は、かなり抽象的なことから、それぞれが何を意味するか明らかではないかも知れないが、以上のような考え方が、モデルのダイナミクスに自然と現れてくると思われる。特に、地層処分性能評価シナリオの開発においては、この関係性のダイナミクスは重要な論点であると思われる。それは、ダイナミックな関係は、トップダウンやボトムアップといった固定したモデル化のアプローチでは示しきれないということである。複雑系が何らかの意味での非還元性を意味する以上、それを捉えていく新しい方法論が必要とされるであろう。そのための可能な方法としては以下のような例があるように思われる。

- ・相空間を解剖しつつ、その構造自身の変化を静的、動的両面から見ていく方法
- ・多彩な現象の中から、そこに内在する抽象的構造と関係性を再構築していき、それを通して一般的な概念に至る方法
- ・いくつかの基本的な概念要素を組み合わせる人工システムを構成していく方法
- ・マイクロでもマクロでもない中間のレベルから上下のレベルをにらみつつモデルをつくる方法
- ・関係や過程などの動的状態を扱える集合論体系に立脚し、複雑系をとらえる新しい言語体系を構築する方法
- ・従来の直感に反する事象を数学的に基礎づけ、計算機の中で起こる数学的経験を豊富にすることで新しい直感を養いながら理論化する方法。

また、全体の示す性質が、構成要素の性質に分割しては理解できないという複雑系の特徴は、還元論型方法論の限界を示している例であると考えられる。

### (5) 構成的理解

地層処分性能評価問題などでは、従来の記述的な方法だけでは理解されない巨大で複雑対象を相手にしている。そのような、複雑な現象の構成的理解のための問題意識の触発はカオスの研究によってなされた。従って、カオスは今後の研究の重要な課題となるであろう。特に、創発とカオスの関係は、現在はマルチエージェントによるモデル協調計算等の考え方でプログラムして、可能性を確認することができるため、現実的な工学面での応用も遠い将来ではないように思われる。

たとえば、乱流のオンセットに対してカオスは普遍的な見方を与えた。それは、乱流の満たすべき方程式を整然と解いていくより、場合により直接的には流体力学とは結び付かない簡単なカオスのモデルの研究が乱流への洞察を与えることがあったと解釈できる。カオス的な観点をさらに発展させることによって、記述的というより、構成的な理解の重要性が強調されるように思われる。前節に述べたように、結合写像格子 (CML) は、ある意味では時空間での複雑な現象への構成的なモデルの一つの例を与える。例えば、頻繁に議論される大気大循環モデル (GCM: Global Circulation Model) もこの例となっていると期待される。複雑系は、こういった「世界を構成していく」方法により、複雑現象の共通な構造を探っていくことを目指しているが、そのような「普遍性」に基づくアプローチは、従来の統計力学と変わらないのではないかという批判も考えられる。

### (6) 仮想世界の理論構築

以上の検討から、地層処分性能評価のような複雑でダイナミックな系を構成するモデルについては

以下のように基本的な考え方を整理することが出来る。

- a. 自然は複雑であり、人類はそのごく表層部分を明らかにしたにすぎない。
- b. 単純な還元論的思考の限界はカオス理論の中で明確にされた。
- c. 複雑現象に対して、単一の力学的予測は破綻する。
- d. 観測者の視点を明示的に自然の中に導入することにより、純粋客観でも純粋主観でもない相対的な観点が入り込む座標の存在が示された。
- e. 純粋客観の視点は論理によってのみ与えられ観測によっては与えられない。すなわち純粋客観の世界は論理的な世界である。
- f. コンピュータによって創られる「人工的」な世界と「自然」を絶対的な基準で区別することは不可能である。
- g. コンピュータの中で世界を構築するという工学的な手法は複雑現象の理解へと到達するおそらく唯一の方法を与える。

自然科学と工学は、もはや一体として考えるべきであり、その区別は存在したとしても相対的なものである。そして、ヴァーチャル・エンジニアリングが、そのような観点で研究開発されることにより、新しい科学哲学や技術を生み出すことが期待できる。従って、新しい科学技術の観点としてのヴァーチャル・エンジニアリングの基本思想として、次のようにまとめることができる。

従来の観点では、力学的予測を諦めることは力学的法則が存在しないことを意味した。しかし、新しい観点では、力学的法則がたとえ存在したとしても、力学的予測はできないことが示される。従って、b. が示すように、現象理解や定式化と現象予測の間にも、単純な還元論的思考を働かせることはできないのである。次にd. により、観測結果は観測手段に相対的にしか意味をもたないので、単純に観測結果を「真理」と認めることは危険である。観測結果は、確かに特定の観測手段にもとづく「事実」を与えるが、それは観測行為に相対的な真理である。

このように、実験による客観性の獲得を諦めざるをえないが、しかし一方e. により、論理を信じ数学的構造を客観性の基準に据えることができる。そのためには、コンピュータでの実験やシミュレーションを複雑現象の理解にとって非常に大切なものと捉えるが、それは次のような理由による。

コンピュータ実験では、コンピュータは素過程として論理的に動いている。しかし一方で、コンピュータ内で起こる現象を理解したり、定式化したりするとき必要になることは、それら論理の筋をすべて追うことではなく、コンピュータ内の現象を通して養われた新しい直観に基づくことである。このとき、まず信じることは、素過程での論理の正当性である。これを信じた後、今までの物理学的直観とは異なるかもしれない新しい直観を養わなければならない。たとえ、ある事実が我々の物理的直観に反したとしても、このような意味で「数学的事実」であれば、それを真理とみなすことをためらってはならないであろう。数学は、それ独自の自律運動を遂げているのであるが、それとは別に、複雑現象に対する数学的定式化はコンピュータの中で形式化するのが適当である。

また、f. とg. により、複雑現象をコンピュータの中で形式化、モデル化することで、我々は新しい型の「予測」が可能になることをみる。それは、系の構造と歴史に対する可能な過程を提示することで単一の現象の「先」を予測するのではなく（それ自体は、すでに示したように不可能である）、構

造と歴史の集合を予測するのである。a. でみた認識は、科学の非力を意味するのではなく、自然の超越性を意味しており、そのことにより、逆に我々の自然認識が f. と g. のような新たな方法を生み出す契機を与えたとみるべきである。そのようにして、複雑現象の理解をめざした研究から新しい自然認識へ到達することが期待できる。それはまた、人工世界と自然との相対化であり、観測行為の新しい定式化をめざすものである。そのようにして組み立てられた仮想世界で新しい科学技術を開発することができると思われる。それが、ヴァーチャル・エンジニアリングの基本的な思想である。

### 3. 9. 2 ゲームとシミュレーション

コンピュータ・シミュレーションの分野には、物理化学問題のシミュレーションとは別の、ゲームの観点からのシミュレーション（ゲーミング・シミュレーションと呼ばれる）がある。本項では、このゲーミング・シミュレーションについて、その特徴を概観し、シミュレーション・システム開発の参考とする。ここで、シミュレーションについて、次のように考える。シミュレーションとは、現実のあるいは提案されたシステム、プロセス、環境が持つ中心的な特徴あるいは要素についての操作的モデルである。この考え方は、以下の側面を強調する。

- ・シミュレーションとは、ある形式のモデルである。
- ・そのモデルとは、静的なものでなく動的なモデルである。
- ・対象システムの選択された要素のみが含まれている。
- ・対象システムといっても、いくつかの異なる種類がある。

科学技術系のシミュレーションは、従来から研究開発の現場で広く普及して来たため、考え方が普及しているが、それ以外のモデルのシミュレーションについては今後の課題と考えられる。

例えば、地層処分性能評価シミュレーションは、漏出した核種が生活圏に到達しないことを、もしくは、例え到達しても長時間に亘る地層からの移行の結果、核種の放射線強度が天然の放射線源より下回る程度に崩壊していることを確認することに目的がある。実際には、地層処分プロジェクトの観点からは、核種の移行に起因する非科学技術系の問題も考えられる。例えば、処分システムの建設と操業に係わる、安全性、経済性、社会心理等の“処分プロジェクトの及ぼす社会システムへのインパクト”も重要評価事項であると考えられる。そのような問題は、物理科学に基礎をおいた数理モデルによる記述という観点より、むしろゲーム性の強いモデルとなると思われるが、シミュレーションにおいては、システムの中心的特徴が同定され、モデルとして表現されることで、現実世界のシステムと同じような動きが現われる。さらに、ゲーミング・シミュレーションの特徴ある応用として教育が上げられるであろう。例えば、地層処分性能評価のような動機（核種が生活圏まで漏出してこないか？）を確認するための複雑な科学は、一般人や学生には動機を理解できてもプロセスを理解することは難しい。そのような場合、ゲーミング・シミュレーションは、より解かりやすく説明することが期待できる。

#### (1) 考え方の違い

ゲームという言葉は、全体的にせよ部分的にせよ、プレイヤーの意思決定に基づいて動作するシミュレーションに用いられる。ゲーム的側面を持つシミュレーション、すなわちゲーミング・シミュレ

ーションにおいては、環境と参加者の活動とがゲームとしての特徴を持っている。プレイヤーには、演ずべき役割、達成すべき目標、行うべき活動、何が成しうるかの制約、そして各プレイヤーの行動や、システム内での他の要素（偶然を含む）の作用の結果としての（正と負の）利得が与えられる。ゲーミング・シミュレーションでは、ゲームにおける役割、目標、活動、制約、結果、そしてそれらの結び付きが、現実世界から抽出され形作られる。先に用いた言葉でいえば、ゲーミング・シミュレーションは、現実世界のシステムのこれらの要素をシミュレートしているのである。例えば、立地問題などの土地利用ゲームを考えると、ある地域と、その周辺での土地利用の決定に影響を与える要因の中で、比較的基本的なもののいくつかを、ゲーミング・シミュレーションの中で抽象化して表現している。このときプレイヤーは、それらの要因により設けられている制約のもとでプレイする。

プレイヤーは、どのように資源を用いるかを意思決定することで、施設を建設し、運用し、維持する。ゲームの始めにプレイヤーには、税制、許される土地利用の範囲、雇用主の給与支払額と雇用者の所得、そして、企業と労働者のための交通・輸送費が情報として与えられるであろう。プレイヤーは、定期的に資産を再評価しなければならず、このとき減価した建物を修繕するかどうかの意思決定を行う必要がある。もしプレイヤーが、施設や地域活性にある程度以上の費用をかけていないと、プロジェクト推進が難しくなる可能性が高くなるであろう。このような問題には、ゲーミング・シミュレーションの手法が物理学のモデルと代わって重要となる。

## （２）ゲームとシミュレーション

シミュレーションとゲームの間の区別は明確でない。この２つの言葉はしばしば同義語として用いられる。実際、風洞や宇宙船の例、あるいは医者や医療スタッフを訓練するための救急救命シミュレーションのような、ゲームでないシミュレーションは数多くある。同様に、いくつかのゲーム（特に多くのカードゲームやボードゲーム）は、シミュレーションではない。その構成要素が、現実世界のシステムやプロセスを表現するものとして設計されていないからである。ゲーミング・シミュレーションは、ゲームとシミュレーションの双方の特徴を持つ、教育・訓練用の教材として開発されるソフトウェアの総称である。ゲームや略式のドラマのようなものとして考えるのではなく、参加者が外側でなく内側に入り込んだ、一種の事例研究のように考えるべきである。

## （３）ゲーミング・シミュレーションと教育

ゲーミング・シミュレーションは、これまで数多くの教育・訓練目的のために、設計され、利用され、成功を収めている。その主なものは以下のとおりである。これは、一般人が計画する地層処分システムがなぜ安全に放射性廃棄物を隔離できるのかについて、仮想的に体験学習するための有効な手段であるとも考えられる。

- ・動機づけと興味づけ  
 主題とする事柄に関する動機づけと興味づけ。  
 特定的话题をさらに研究するための動機づけと興味づけ
- ・教育、情報の提供  
 あるいは他の形式ですでに与えられた情報の強化事実全体像またはシステム的な理解。特定の役割のシステム全体に対する関係選択肢、政策、課題についてのより広い認識特定の政策や事象により起こりうる結果
- ・技能開発  
 批判的な思考と分析。意思決定、コミュニケーション技能企画書の作成、放送や新聞の記事原稿の執筆、予算書の作成といった特定の技能求職活動やチーム内労働者の管理、危機事態への対処といった将来発生するかも知れない特定業務への準備
- ・態度変容競争や協調といった社会的価値他の役割にある人々に対する同感
- ・自己による評価、他者による評価

個人の知識、技能、思い込み、態度。リーダーシップ能力に対する自覚参加者の知識、業務、思い込み、態度。リーダーシップ能力に対する教師（またはトレーナーや雇用主）の評価以上に挙げた目的の項目は、個々の教師やトレーナーの狙いの多くを含んでいるが、ゲーミング・シミュレーションだけが、これらの教育上の狙いを達成できる唯一の方法ではない。

実際に、講義、事例研究、ロール・プレイング、視聴覚教材などを通じて、教師やトレーナーは、これらの狙いを追求し、たいていは達成してきた。ゲーミング・シミュレーションは、ある種の目的を達成するためのよりよい方法といえる。ゲーミング・シミュレーションは、目的によっては他の技法を代替することもあるが、他の技法と連係して用いると効果をより発揮するもので、一般的にいつて、教師やトレーナーがすでに持つ技法の「道具箱」に応用すべきものである。

#### （４）教育とゲーミング・シミュレーション

従来、教育や訓練の標準的な形は、講義だけか、あるいは講義と討論の組み合わせによるものであった。基本的な問題点は、伝統的な講義形式においては、学習者が情報の単なる受け手であるということにある。教室内の学習者は、自分の考えを発表することはほとんどないので、彼らの関心は、頻繁に他の目的やより大きな関心のある事柄に移りやすい。このため、教師は複雑なトピックを述べる際には、適切な箇所より始めなければならず、終わりまでその道筋に沿っていかなければならない。このような直線的なアプローチでは、学習者にとって全体の本質を把握することが難しい。

このアプローチでは、学習者は、一度に一断片しか見ることができないことが問題である。直線的アプローチでは、決して全体の理解に結びつかない。このことは、事象と行動の“同時性”を理解させる場合に顕著である。同様に、伝統的な教室での“議論”は、教師による講義のあとに、コメント、質疑、解答という形で行われる。そのような議論は、多くの場合、最も重要なコメントを持ったから人ではなく、グループの中で最もお喋りな人の意見から始まる。同じように、与えられた話題を考察するのに費やされる時間は、話の内容の相対的な重要性というよりは、話し手のこだわりの程度（そして、それを打ち切ることに對する教師の躊躇の程度）で決定される。さらに、議論が、講義のなか

での最初の論点より始まり、話された順序で順番に進んでいくことは稀である。

したがって、多くの場合、質疑応答は論理的順序とはまったく関係なく行われる。3番目に、講義あるいは言語モデルを利用するときの問題点は、“システム”としての特性を明確に表現しづらいことにある。与えられた要素や行動の間の多重の関連や、その帰結を言葉で表わしていくのは難しい。1つの社会システムを単にそれに飛び込んでいくだけでは、それがあまりに複雑すぎて、理解不可能なことが多いのであるが、一方で、言葉による記述では、過度の単純化を避けられない。ゲーミング・シミュレーションによる教育はどのように違うのか。これには、いくつかの答えがある。

第1に、ゲーミング・シミュレーションでは、講義とは反対に、教室のひとり一人は“能動的な学習者”である。彼らは意思決定をし、結果を受け入れ、立場を明確にすることで、システムが機能するようにしなければならない。プレイ後の事後分析において、何がなぜ生じたのか、それがどのように現実世界に対応しているのかということに焦点を当て、さらに、ゲーミング・シミュレーションの限界について議論する。

第2に、学習者は様々な“視野”から活動する機会が与えられるべきである。学習者は、情報の単なる受容者であってはならず、時として、発信者、判定者、交換者となるべきである。

第3に、どのような活動にも、それ自体の目的と動機があるべきで、活動が（成績のような）ある結末に至るための単なる手段であってはならない。即ち、効果的な学習環境の中では、活動は自己目的的である。

第4に、学習者は権威への依存から解放されるべきであり、自分自身の力で考えることを許容されるべきである。このようにして、学習者は学習プロセスにおいて、より“生産的”になれる。

第5に、学習環境は学習者の活動に対して反応するようになっているべきである。環境は、単に学習者にフィードバックを与えるだけではなく、学習者が“思慮深く”自分の進歩を評価できるよう支援すべきである。

教育現場では講義を行い、その後で議論をするという形式でなく、むしろゲーミング・シミュレーションにおいてこそ、学習者は提供された論題の全体像を経験する。というのは、ゲーミング・シミュレーションでは、システムの構成要素は1つ1つ別々にではなく同時に与えられるからである。さらに、プレイの後で行う議論や分析は、無遠慮な発言や断定的な主張といったものでなく、通常、役割によって構造化される。

また、ゲーミング・シミュレーションは、特に“システム”としての特性を表現するのに有効である。システムというものは、言語モデルを通じてよりも、むしろ系統図や絵のような、図式モデル（graphic model）、あるいは物理的表現を使った方が明確に記述できると長い間認識されてきた。ゲーミング・シミュレーションの教育学的成果を評価する研究は数多い。ゲーミング・シミュレーションによって、教育上の問題のすべてが解決されるわけではないが、非常に多くの社会システムや社会問題について、教育や学習のための刺激的な機会が提供されることは明らかである。ゲーミング・シミュレーションは、多くの公共政策の論議では、意思決定者にとって、全体的な見通しとシステム的な理解が必要となるが、これこそゲーミング・シミュレーションを行うことで得られるからである。

##### (5) ゲーミング・シミュレーションとコンピュータ・シミュレーション

抽象化の程度としては、ゲーミング・シミュレーションは、ロール・プレイングとコンピュータ・



シミュレーションの間に位置する。コンピュータ・シミュレーションでは、構造やプロセスについての数学的モデルを詳しく展開するためにコンピュータを利用する。システム内の様々な関係を表現するために、数学的記号や方程式が使われ、時間の経過に対する変化を調べるためにモデル方程式に示す通りの演算を繰り返し行う。

例えば、特定の問題解決にモンテカルロ・シミュレーションとゲームのどちらかを選ぶ場合には、2つの方法論の長所と短所を考慮しなくてはならない。シミュレーションは、ランダムなシナリオ、極端なシナリオ、統計的出力分布といったものをすばやく容易に作り出す能力がある。ところが、ゲームは、ただ一度あるいは統計的に有意でない数回の反復しかできない。シミュレーションは、経営者が対処すべき問題の範囲を理解するのに役立つ計算結果を与える。反対に、ゲームは、「実用的」な感じを与えるとともに、シミュレーション設計者が仮定した人々の反応ではなく、実際に人々がどのように反応するかを知る機会を与えてくれる。

分析技法およびシミュレーション技法が、構造の比較的単純な競争状況および紛争状況をモデリングするのに適しているのに対して、ゲームでは、現実の状況に特有の多面的な複雑性を導入し、理解することが可能になる。コストを考えると、ゲームは、設備やスタッフへの投資を必要とするだけでなく、膨大な事柄を考察しなくてはならなくなる可能性がある。シミュレーションは、統計的に有意なデータを得るために、膨大な大型コンピュータ運用費用が必要となる可能性が高い。一般的にゲームには次のことがいえる。

それは、人間の反応や系の相互作用が予想できなかつたり、人間が研究対象のプロセスの一部となっているような探索型研究に、専門家の考えを引き出すために、状況の本質、特に競争的状況の本質を理解させるための訓練に適している。

このようにゲーミング・シミュレーションは、プロセスの最初か最後に人間の意思決定入力があるような状況や、問題が比較的単純な状況に適している。従って、処分システムの社会的許容性に係わる意志決定を目標とする性能評価という観点からみた場合、地層処分ヴァーチャル・エンジニアリング・システムには、将来、ゲーミング・シミュレーションの思想（技術）を取り入れるべきであると考えられる。

### 3. 10 動画化シミュレーション・プログラムの試作

本項では、試作した動画化シミュレーション・プログラムについて仕様を中心に記述する。尚、詳細は別紙（技術報告書第2分冊）参照のこと。動画化シミュレーション・プログラムは、ウイスコンシン大学マジソン校宇宙科学技術センターで試験的に開発されたVis5Dと呼ばれるソフトウェア・システムを、地層処分性能評価シミュレーション・コードで利用できるよう改訂して試作した。動画化シミュレーション・プログラムは5次元長方形の形をしたデータ上で稼働する。そのデータは“格子”の各地点における実数で、3つの空間次元と1つの時間次元、更に複数の物理的な変数を列挙するための次元を扱う。動画化シミュレーション・プログラム・システムには、動画化シミュレーション・プログラム視覚化プログラム、5次元データ“格子”を管理、分析する幾つかのプログラム、データをファイル書式に変換する命令と原始コードのサンプル、動画表現の仮想世界を構築する処分システム・オブジェクト等がある。

### 3.10.1 プログラム仕様

以下に、動画化シミュレーション・プログラムの実行に必要なハードウェアとソフトウェアについて、及びシステム環境仕様について示す。

#### (1) システムの必要条件

動画化シミュレーション・プログラムは、以下のシステム環境で動作する。メモリー容量は最低32MB以上が望ましく、グラフィクス機能には最低8ビットのカラーディスプレイが必要である。

- ・シリコングラフィックス社ワークステーション  
IRIXのVer.4.0.1もしくは、それ以上のもの。
- ・IBMワークステーション RS6000  
モデル320H以上、AIXのVer.3もしくは、それ以上のもの。
- ・ヒューレットパッカード社ワークステーション 7000または9000シリーズ  
XP-UXA.09.01もしくは、それ以上のもの。
- ・DECワークステーション アルファ  
OSF/1 V.1.3もしくは、それ以上のもの。
- ・ライナックスと互換性があるIBMのPC  
Pentium 75MHzもしくは、それ以上のもの。または、ライナックス1.0もしくは、それ以上のもの。

#### (2) 動画化シミュレーション・プログラムの構成

動画化シミュレーション・プログラムは、“Vis5d.4.2”ディレクトリに格納されている次のファイル及びサブディレクトリから構成されている。

README	READMEテキストファイル
README.ps	READMEポストスクリプト・ファイル
NOTICE	GNUパブリック・ライセンス・ファイル
PORTTING	Vis5Dポーティングノートファイル
LAMPS.v5d	サンプルデータファイル (LAMPS)
SCHL.v5d	サンプルデータファイル (ボブ・シュルシンガーの雷雨モデル)
OUTLSUPW	世界大陸地図ラインファイル
OUTLUSAL	米国地図 (州未分割) ファイル
OUTLUSAM	米国地図 (州分割) ファイル
EARTH.TOP0	地球地形ファイル
api.htm	Vis5dのAPI (Application Program Interface) に対するハイパーテキスト
script.html	Vis5dのスクリプト言語に対するハイパーテキスト
*.tcl	例題スクリプト
Vis5d	Vis5d視覚化プログラム
v5dappend	v5dファイル追加用
v5dinfo	v5dファイル・インフォメーション・ユーティリティ
v5dstats	v5dファイル・統計用・ユーティリティ
v5dedit	v5dファイル・見出し編集用・ユーティリティ
v5dsimport	v5dファイル・変換用・ユーティリティ
gr3d_to_v5d	McIDAS GR3Dファイルのv5dフォーマット変換用・ユーティリティ
comp_to_v5d	comp5dファイルのv5dフォーマット変換用・ユーティリティ
listfonts	SGIのフォントIRIS GL用配列・ユーティリティ
src/	Vis5dのソースコード格納サブディレクトリ
util/	Vis5dのユーティリティ格納サブディレクトリ
lui5/	LUIユーザーインターフェイス・ライブラリ格納サブディレクトリ
Mesa/	Mesa3Dライブラリ格納サブディレクトリ
import/	v5dsimportソースプログラム格納サブディレクトリ

userfuncs/	ユーザー作成解析機能格納サブディレクトリ
contrib/	コントリビューションファイル格納サブディレクトリ
convert/	サンプルデータ変換用ソースコード格納サブディレクトリ

動画化シミュレーション・プログラムの動作は最低32MBメモリ必要とする。システム環境のメモリが32MB以上あるならば、動作のための設定値を変更することが望ましい。設定値の変更はサブディレクトリ/src内のvis5d.hインクルードファイルのMBSパラメータ値を変更する。また、地系図およびマッピングファイルの省略値を変更する場合は、サブディレクトリ/src内のvis5d.hファイルのT OPOFILEまたはMAPFILEパラメータを変更する。なお、vis5d.hファイルの変更後は、プログラムを再コンパイルする必要がある。動画化シミュレーション・プログラムをネットワーク上で複数のユーザーで利用するならば、Vis5dプログラムと各データファイルを一般的に使用される/usr/localといったサブディレクトリ内に移動したほうが利用上都合がよい。その場合、vis5d.hファイル内容に各ファイルが何処に位置（保存）しているか変更しなければならない。

### (3) 動画化シミュレーション・プログラムへのデータ入力

動画化シミュレーション・プログラムは、5次元の長方形空間を持つデータにより動作する。5次元については、まず3次元として（行（横）、列（縦）、高さ（または緯度、経度））の立体空間、4次元は時間であり、5次元目は、気温、気圧、水の容量等、複数の物理的な変数を列挙したものである。動画化シミュレーション・プログラムでデータファイルを表示するためには、必要なパラメータ（行、列、高さ、タイムステップ等）を設定する。例えば、地理的な位置と方位（マッピングの投影図）、変数の名称、各タイムステップや現在の日時等である。動画化シミュレーション・プログラムは、v5d形式とcomp5d形式のファイルに対応し、それぞれは圧縮された3次元書式データであることから処理速度が早く効率的である。動画化シミュレーション・プログラムに外部の解析コードを組み込む（接続）ためには、v5d形式ファイルに変換するプログラムの作成が必要である。v5d形式ファイル変換は、v5dライブラリの機能を用いて行うことができる。次に示す4つのサンプルプログラムは、動画化シミュレーション・プログラムへのデータ入力例を示したものである。これらは、convert/サブディレクトリに格納されている。

foo_to_v5d.f	投影する長方形を想定し線による垂直座標を設定 (FORTRAN)
foo2_to_v5d.f	投影する垂直座標に合致させるために各変数に垂直レベルの異なった数値を与える (FORTRAN)
foo_to_v5d.c	投影する長方形を想定し線による垂直座標を設定 (C)
foo2_to_v5d.f	投影する垂直座標に合致させるために各変数に垂直レベルの異なった数値を与える (C)

変換プログラムを作成するにあたっては、v5dCreate（またはv5dCreateSimple）、v5dWrite、v5dCloseの機能を使用する。v5dCreateとv5dCreateSimpleはv5d形式ファイルを作成する。v5dCreateは作成のための基本パラメータを設定するのみであり、v5dCreateSimpleはv5dCreateより複雑なパラメータを与えることができる。作成機能は、CおよびFORTRANバージョンがある。

### 3.10.2 基本操作

動画化シミュレーション・プログラム使用法について記述する。それは、マウスを使うことによって操作する。尚、操作に応じた画面のコピーを付録-Aに添付した。

#### 3.10.2.1 操作開始

動画化シミュレーション・プログラムは、以下のコマンドで実行を開始する。

v5d [options]

[options] は以下のとおりである。

##### -alpha

スクリーン制御方式を透過伝送機構でなくアルファ・ブレンドとする。

##### -area N

[SGI only] は読み込むためのMcIDAS 領域ファイルの順序の一番目を指定し、それから立体ボックスの内部を表示する。

##### -box x y z

縦横比か立体ボックスの比例を指定させる。省略値は、2 2 1である。

##### -barbs

速度ベクトルの地点で速度のbarbsを使用する。

##### -date

時計の'yyddd'の地点での'dd month yy'を使用する。

##### -font xfontname

##### -font gfontname height

立体ウィンドウ内の時計をテキストラベルに対して使用するフォントを設定する。また、'vis5d'とだけタイプし、オプションを調べることによってシステム上で使用されるフォントのオプションの形を確定することができる。最初の形はXウィンドウシステムのフォントの名称である。システム上のXのフォントのリストを見るためxfontsのコマンドを使用する。2番目の形はIRIS GLのフォントの名称とサイズ (72=1インチ) である。システム上のGLのフォントのリストを見るため動画化シミュレーション・プログラムで格納されているリストフォントのコマンドを使用する。

例1 : vis5d LAMPS.v5d-font fg-30

例2 : vis5d LAMPS.v5d-font Helvetica30

**-full**

境界線のない、全画面サイズのウィンドウとして立体ウィンドウを開く。

**-funcpath pathname**

ユーザーのFORTRAN機能を検索するためのディレクトリを設定する。

例 vis5d LAMPS.v5d-geometry 640x480-10+10

**-hirestopo**

高解像度の地形図を表示する。これは高速図形ハードウェアを持ったシステム上でのみ推奨される。

**-legend position size**

カラーキャプションの位置とサイズを設定する。位置の値は、各々1（底部、省略値）、2（頂点）、3（左）、4（右）である。サイズはキャプションバーの高さで、10から1000の間（省略値=128）にある。

**-log [a] [b]**

線形の代わりに、対数軸の高さを表示する。オプションの変数であるaとbは、高度/気圧の方程式内の基準で、パラメータ因子でもある。省略値は、各々1012.5と-7.2である。

**-map file**

OUTLSUPWの省略値と違うマッピングファイルを使用する。

例 : vis5d LAMPS.v5d-map OUTLUSAL

**-mbs n**

32メガバイトのメモリサイズを仮定したシステムを置き換える。

**-path pathname**

電流の代わりにマッピングと地形のファイルに対する異なった経路を使用する。

例 : vis5d LAMPS.v5d -path/usr3/data

**-projection p**

射影の表示を設置し、省略値はその自然の投射（データファイルから獲得される）の中でデータを表示させる。pは次のどちらかである。

cylindrical 円柱の地球上のデータを表示

spherical 球状の地球上のデータを表示

最初の3文字が重要である。

例 : vis5d LALMPS.v5d -projection spherical

#### -quickstart

動画化シミュレーション・プログラムの開始時には、たとえファイル全部がメモリ内で適合したとしても、どの格子もロードしてはならない。格子は必要に応じて読み込まれる。このオプションはNFSを通してファイルを読み込む時に役立つ。

#### -rate ms

省略値のアニメーションの率を変える。msは枠の間においてミリ秒の最小限の遅れである。省略値は100msである。

#### -script script.tcl

自動的に実行するvis5d/Tclスクリプトを設定する。

#### -sequence filename

地形図を超えたテクスチャマップの印象の順序を持つファイルを設定する。なお、全てのシステム上で使用可能ではない。

#### -texture rgbfile

SGI .rgbのファイルを、地形図上のテクスチャマップを設定する。なお、全てのシステム上で使用可能ではない。

#### -topo file

EARTHTOPOの省略値でなく他の地形図ファイルを使用する。

#### -trajvars uvar vvar [wvar]

どの変数が軌跡の追跡に使用されるかを設定する。省略値はU、V、Wである。

例：vis5d LAMPS.v5d-trajvars U2 V2 W2

#### -vertical v

縦の座標系、データファイルから得られる省略値を設定する。vは以下の内の1つである。

- 包括的 -汎用体内で線形に、均等に行送りされたレベル
- 均等 -km内で線形に、均等に行送りされたレベル
- 不均等 -km内で線形に、不均等に行送りされたレベル

vの最初の3文字だけが重要かつ必要とされる。追加の変数に対しては指示がある。

例：vis5d LAMPS.v5d-vertical nonequal

#### -wdpy xdisplay

異なったXディスプレイの上にウィジェットを置く。スライドとビデオの作成の組合せにおいて役立つ。

例：vis5d LAMPS.v5d-full -wdpy pluto:0

### -wide w

画素（省略値は1.0）内のラインセグメントの幅を設定する。ビデオ作成に役立つ。

例：vis5d LAMPS.v5d-wide 3.0

### -wind2 uvar vvar [wvar]

Hwind2、Vwind2、Stm2ベクトルスライスを描写する時に使用する、U、Vと（任意選択で）Wの速度構成要素の変数の補助的なセット名を指定する。同時に視覚化したいと思う速度ベクトルの構成要素の2つのセットに役立つ。

例：vis5d MYDATA -wind2 U2 V2 W2

変数なしで動画化シミュレーション・プログラムを開始させた場合、全てのコマンドライン選択とキーボード機能のリストを得ることになる。他方動画化シミュレーション・プログラムはデータファイルの読み込みによって開始する。

## 3. 1 0. 2. 2 コントロールパネル

動画化シミュレーション・プログラムを起動すると、2つのウィンドウが表示される。立体ウィンドウはスクリーンの右側に、コントロールパネルは左側に位置している。その左上は、現在時刻を表示するアナログとデジタルの組合せの時計である。コントロールパネルには幾つかのボタンが集まっている。

[ANIMATE]	[STEP]	NEW VAR	EXIT
[TEXTURE]	TOP	SOUTH	WEST
[TOPO]	[MAP]	BOX	CLOCK
SAVE	RESTORE	GRID #'s	CONT #'s
[ANIM-REC]	REVERSE	[SAVE PIC]	[PERSPEC]
SCRIPT	INTERP	UVW VARS	

これらのボタンは動画化シミュレーション・プログラムの主要機能の制御に使用される。ボタンの幾つかは、大括弧 [] に囲まれており、動画化シミュレーション・プログラムが開始すると空白になって表示される。これは現在のデータファイルに対してボタンが当てはまらない時に起こる。それはボタンがコマンドライン選択に矛盾するか、または、その機能がハードウェアに対して使用不可能かのどちらかである。ラジオボタンの次の集合は、マウスを用いて立体ウィンドウを制御するものである。

- 正常 : 正常なマウスモードは、立体ウィンドウ内の図形を循環、急上昇、パンすることに使用される。
- 軌跡 : 速度の軌跡の作成と描写のために使用される。
- スライス : 水平な縦のスライスの再位置付けのために使用される。
- ラベル : 立体ウィンドウ内でテキストラベルを作成、編集するために使用される。
- 証明 : 立体格子を通して立体カーソルを作動させることにより、個別の格子の値を検証するため使用される。

これらのモードは相互に排他的であり、その時点で唯一つが選択される。これらのボタンの右側は、マウスボタンキャプションである。選択されたモードに対するボタンの表示を次に示す。

Hwind1      Vwind1      HStrm      Hwind2      Vwind2      VStrm

速度ベクトルのスライス (HwindかVwind) は、速度の方向内の地点である小さな矢印を描くことによって速度の値を描写する。各回線区分の長さはその振幅を表示する。回線区分の末尾は、全て立体ボックスを通して水平、または縦の面で固定されている。水平の速度の流線型のスライス (HStrm) は、水平面に流線型を描くことによって速度の値を描写する。縦の速度の流線型のスライス (VStrm) は縦面に流線型を描くことによって速度の値を描写する。スライス面の位置は、"Slice"モードになっているマウスで変更することができる。

コントロールパネルウィンドウの底の部分は、ボタンの平面マトリックスを包含する。各行はデータファイル内の物理的変数に対応し、各列は図的表現の型の1つに対応する。正確な行と列を選択することによって、立体の同一面、水平の輪郭のスライス、縦の輪郭のスライス、水平の色彩スライス、縦の色彩スライス、或いはボリュームレンダリングというような、どんな変数をも見ることができる。このボタンのマトリックスは、ウィンドウ内で適合するより多くのボタンの行があるならば、画面移動することができる。ボタンマトリックスの画面移動をするには、マウスを使ってスクロールバーをドラッグしても、或いはキーボード上の上下矢印のキーを押してもできる。

どの図形の表示も、左マウスボタンをクリックすることによって制御される。各々の図形の型は、スイッチを入れると現われる小さなポップアップ制御ウィンドウを持つ。制御ウィンドウは各図形の型によって異なる。表示を切り換えず、図形の制御ウィンドウを持ち出すには、中央のマウスボタンを使用する。図形が表示されている時は、その表示中の異なった変数を識別しやすくするボタンと同じ色になる。図形の色を変えるには、右マウスボタンをクリックすると4つのスライダとともに小さなウィンドウが現われる。赤、緑、青の基準を変えることにより、どんな色をも作ることができる。

コントロールパネル・ウィンドウに他のウィンドウが覆い被さった場合、マウスポインタが立体ウィンドウにある間に、"F1"キーを押して前面に持っていくことができる。これは特に'-full'オプションの使用時に役立つ。

### 3. 10. 2. 3 画面操作

コントロールパネルの中でボタンの一番上の集合は、動画化シミュレーション・プログラムの主要機能を操作する。

#### ANIMATE

このトグルボタンは、動画を出すか、または閉じる。前面の動画には、左か中央のマウスボタンを使用し裏面の動画には右のボタンを使用する。1つのタイムステップでデータファイルが見える時は現われない。動画を遅く、または早くする時は、マウスのカーソルが立体ビューウィンドウにある間に、SとFのキーを押す。



## STEP

このボタンはどのマウスボタンを押すかによって3通りの使用法が可能である。

左ボタン : 1つのタイムステップより以前の段階

中央のボタン : 最初のタイムステップへ行く

右ボタン : タイムステップの後方ヘデータファイルが1つのタイムステップと共に見える時は、このボタンは現われない。

## NEW VAR

物理的変数を複写するか、または、外部の解析機能呼び出すために使用される。

## EXIT

プログラムから出ること。EXITの確認を問うためウィンドウが現われる。

## TEXTURE

テクスチャマップがロードされていたら、その表示を出したり消したりして切り換える。

## TOP

どのマウスボタンを押すかによる。

左または中央 : 立体ウィンドウを省略値の最初の視点へリセットする。

右 : 立体ウィンドウを最後の視点へ設定する。

## SOUTH

どのマウスボタンを押すかによる。

左または中央 : 立体ウィンドウを南の視点へ設定する。

右 : 立体ウィンドウを北の視点へ設定する。

## WEST

どのマウスボタンを押すかによる。

左または中央 : 立体ウィンドウを西の視点へ設定する。

右 : 立体ウィンドウを東の視点へ設定する。

## TOPO

地形図の表示を切り換える。地形図ファイルが見つからない時はこのボタンは現われない。地形図の色を編集するには、右マウスボタンでTOPOをクリックする。

## MAP

マッピングラインの表示を切り換える。マッピングファイルが見つからない時はこのボタンは現われない。マッピングラインの色を編集するには、右マウスボタンでMAPをクリックする。

## BOX

立体ボックスの表示を切り換える。

## CLOCK

時計の表示を切り換える。

## SAVE

最新の図形と色を保管する。同一表面、スライス、速度の軌跡、色の各変数を準備後、それらを保管し復元するために用いる。ファイル書式は、Tclスクリプトである。

## RESTORE

SAVEボタンで保管された情報を復元する。

## GRID #s

緯度、経度、キロメートル内のデータファイルの境は、通常ボックスの端に沿って表示される。格子座標内の番号を表示するには、このボタンを使用する。

## CONT #s

輪郭線スライス上に描かれた番号は、このボタンでスイッチオン・オフを切り換えることができる。

## [ANIM-REC]

このボタンは動画のように作動するが、立体レンダリングが遅いとシステム上の動画を早くする。各タイムステップがレンダリングされたら、画像がメモリ内に保管される。動画のループが画像を回復する時は、メモリから動画が早くなる立体ウィンドウへ複写される。

## REVERSE

立体ボックスと時計は、黒い背景では通常白く描かれる。このオプションはそれを反転させ、白い背景に黒いボックスと時計を描く。

## SAVE PIC

立体ウィンドウ内画像をファイルへ保管する時に使用する。どのようなシステムを使用しているかによって、多くの異なったピクチャファイル書式を支援する。SGIシステム上で、IRIX CDROMから導入された'tops'、'frombin'、'togif'等のプログラムが確実に得られなければならない。SGIシステム上でOpenGLを使用する時は、'fromxwd'プログラムも必要である。

## PERSPEC

透視と直交の見える投射の間で切り換える。

## SCRIPT

動画化シミュレーション・プログラムTclスクリプトの実行に使用される。このボタンをクリックすると、Tclスクリプトの実行のため選択した所にファイルリクエストが現われる。

## INTERP

シェルウィンドウ内でTclコマンドを入力すると動画化シミュレーション・プログラムの対話式の解説を始める。INTERPが稼働中は、動画化シミュレーション・プログラムは中断される。

## UVW VARS

軌跡と速度のスライスを計算するのに使用する変数の名称を指定することができるウィンドウを開く。

## LEGENDS

立体ウィンドウ内のカラーバーキャプションの表示を切り変える。

### 3. 1 0. 2. 4 視点操作

'Normal' (通常) のマウスモードでは、マウスは立体ウィンドウ内のデータを見るために使用される。左のマウスボタンを押し、カーソルが立体ウィンドウ内にある間にマウスを動かすと、立体画像が回転する。なお、ボックス回転の自由の3つの段階の2つを制御できる。左マウスボタンを解放し押し直すことによって、ボックス上の"grip" (握り) を変えることができる。一連のマウスの動きの中で、左ボタンを解放し押し直すといった実行に伴い、その動きを通してボックスの制御を覚えていくことになる。

中央のボタンは、マウスをどう動かすかによって全く異なる2つのことを制御する。中央のボタンを押し下げて遠ざけるように滑らせるとボックスは拡大する。マウスを手前に向かって滑らせると、ボックスを縮小させることになる。中央のボタンを押し下げてマウスを右に滑らせると、目に見えない面を (例えばクリッピングされた面) ボックスの内容の視点の切り取りを生じさせるボックスの中へと移動させる。マウスを左に滑らせると、クリッピングされた面を手前へ持って行き、最後は全くボックスの外へ出してしまふ。

右マウスボタンを押すとウィンドウ内のボックスが変換される。これはボックスの中央にない物を掘り下げて検討 (拡大?) したい時に使用する。ボックス回転に対する回転の中央は、ボックスの中央というよりも、むしろ画面の中央に位置している。同一の表面 (立体の輪郭を持つ表面) は、同一値を特定することにより境になる立体の巻を表す。同一表面は指定された同一高度、内部の容量は同一値より多い (または、少ない) 値を含む。外部の容量は同一値より少ない (または、多い) 値を含む。コントロールパネルのボタンマトリックス内のボタンの最初の列は、同一表面の制御を行う。左マウスボタンでこれらのボタンの1つをクリックすることにより、現われるスライダとOKボタンをとともにポップアップウィンドウを出す。スライダ上の同一値を選択し、全てのタイムステップに対しての同一表面を発生させるためOKボタンをクリックする。

ANIMATE (動画) を切り換えると、同一レベルである軌跡の表面をタイムダイナミクスに見るこ

とができる。表面が動画と共に非同期的に発生することから、時計針の回転のような全てのタイムステップに対する表面を見ることは出来ない。新しい表面は連続的な時計の周期上に現われる。

中央のマウスボタンで同一表面ボタンをクリックすると、表面をオンまたは、オフに切り換えずにポップアップウィンドウを呼び出す。

### (1) 同一表面の色

同一表面は一色で全体的か、または他の物理的変数の値による色のどちらかで描かれる。同一表面の色を変えるには、右マウスボタンで適切な同一表面ボタンをクリックする。ウィンドウは、変数名の列（最初のボタンは"monocolor"と表示されている）、赤、緑、青、透明の表示がしてある4つのスライダと共に現われる。省略値によっては単色が使用される。同一表面の色を変えるには、赤、緑、青のスライダを動かすだけである。

"monocolor"でなく他のボタンをクリックした場合、動画化シミュレーション・プログラムが他の物理的変数による同一表面を描くことに気づく。赤、緑、青のスライダは、カラーテーブルの編集プログラムで置き換えられる。マウスを使って新しい曲線を描くか、または、キーボードの上下左右のカーソルキーによって、カラーテーブル（マッピングデータ値を色へ）を変えることができる。

例えば、LAMPS.v5dのデータファイルを見るとき。速度の同一表面の速度を40メートル/秒に設定する。その同一表面は青である。右マウスボタンを使ってSPD同一表面ボタン上にクリックするとカラーウィンドウが現われる。ウィンドウ内のTボタンをクリックすると、気温にともなって同一表面が色づく。マウスボタンか、またはカーソルキーを押すことにより、カラーテーブル内の赤、緑、青の曲線を"drawing"（描く）ことによって、気温の値からマッピングを色に変更することができる。

### 3. 10. 2. 5 スライス

スライスは、立体ボックス内でデータの平面交差する場所を見せることができる。これらのスライスは水平、または縦に配列されていて、軌跡の線、色づけされたスライス、速度ベクトル、速度の流線の何れかを描くことになる。

コントロールパネル上のボタンの最後の集合はボタン、スライスを制御する5番目の列に渡る秒のマトリクスである。水平の軌跡スライス、縦の軌跡スライス、水平の色付けされたスライス、縦の色付けされたスライス等、各々に対するボタンの列がある。データファイルがU、V、Wの変数を含んでいたら、速度ベクトルのスライスボタンの行が出てくることになる。水平の速度ベクトルと縦の速度ベクトルの各々に対する2つのボタンがある。スライス上で起動/回転するには、左マウスボタンで適切なボタンをクリックする。スライスに対する初期の位置はボックスの中央である。緯度、経度、高度の項の中の正確なスライスは、各スライスの一角の近くの細かい数字の位によって与えられる。図形座標の代わりに格子座標としての番号を印刷するには、コントロールパネル上のボタンである"GRID#s"を切り換える。スライスの位置は、マウスを使って対話しながら（双方向に）変換可能である。それには、最初にSLICEラジオボタンを選択することによってSULICEモードにすべきである。スライスを移動し、マウスで簡単にスライスの角を指すには、右マウスボタンを押し、新しい位置にドラッグする。縦のスライスは更に頂点か底部の角の中央を“つかむ（グラブ）”か、それをド

ラッグすることによっておこる垂直の動作へと移動する。スライスが動画モードにある時に動かされるが、新しいスライスが非同期的に計算されるので、跳ね上がりが発生する。

### (1) 輪郭線スライス

水平、または、縦の輪郭線スライスが表示されると（ボタン列2と3）、小さな制御ウィンドウが現われる。このポップアップウィンドウ内に輪郭線間で使用するための時間間隔を入れることができる。その間隔を変更する新しい番号をタイプインするだけである。負の間隔を指定すると正の値が実線で描かれる一方、負の値を持つ全ての輪郭線が破線で描かれる。時間間隔の値の後で、(a, b)の値を選択的に指定することができる。その値は、描かれているaとbの間の輪郭値のみを生じさせる。例えば、以下の数値を入力する。

-10 (-30, 20)

これは、破線で描かれている負の線を持つ10の時間間隔において、-30と20の間の値に対する輪郭線において生じる。コントロールパネル上の"CONT#s"ボタンは、スライス内部の輪郭番号の表示を切り換える。

### (2) 色付けされたスライス

水平または、縦の色付けされたスライス（ボタン列4と5）が表示されると、カラーテーブルウィンドウが現われる。このポップアップウィンドウ内で、データ値からのマッピングを色に変換することができる。コントロールパネルボタンのLEGENDSを選択すると、立体ビューウィンドウ内でカラーテーブルも表示される。

ウィンドウはデータ値の範囲全体に渡って、赤、緑、青の図形を表す。赤、緑、または青の機能を変更するには、左、中央、または右のマウスボタンを押し、新しい機能を得るためにマウスでドラッグする。省略値によって、横の行のデータ値が青に、高データ値が赤へとマッピングされる。マウスの代わりにキーボードのカーソル（矢印）を使って、その形状と省略値の機能の曲線を変更することができる。曲線を左か右に動かすには、左/右キーを、曲線の形状を変えるには、上/下のキーを押す。データ値の機能としてのスライスの透過伝送機構の変更を行う時には、マウスまたは上/下キーを使用する間、シフトキーを押したままにしておく。

カラーテーブルウィンドウに対する、その他のキーボード制御の番号は次のとおりである。

- r 赤、緑、青の値をリセットする
- R 透過伝送機構の値をリセットする
- c 色を画面の外のクリップボードへコピーする
- p 色を画面の外のクリップボードから貼り付ける
- s 色の値をファイルへ保管し、シェルウィンドウ内にファイル名を入力
- l ファイルから色の値をロードし、シェルウィンドウ内にファイル名を入力

### (3) 速度ベクトルスライス

速度ベクトルスライスは、HWIND-1、VWIND-1、HWIND-2、VWIND-2とラベル表示されたコントロールパネルの中央の近くのボタンで表示されている。これらの図形に対するポップアップウィンドウは、速度ベクトルの密度と縮尺を制御する2つの分野を持つ。縮尺パラメータは、描かれているベクトルの長さを掛け算するため使用される。全てのベクトルの長さを倍にする場合は2.0を入力する。長さを半分にする場合は0.5を入力する。密度パラメータは、どの位の速度ベクトルが表示されているかを制御する。この値は0と1の間である。ベクトルの数を1/2にするには0.5を、1/4ならば0.25を入力する。その両方のパラメータに対する省略値の値は1.0である。

### (4) 速度流線スライス

速度流線スライスは、ラインセグメントを連結する速度の経路を示す。ポップアップ制御ウィンドウは、流線（縮尺パラメータは使用されていないことに留意）の密度を制御する。密度パラメータは、どの位の数の流線が表示されているかを制御する。この値は0.5と2.0の間である。流線の数を1/2にするには0.5を、2倍ならば2.0を入力する。省略値の密度は1.0である。

### (5) スライスの色

スライスの制御ボタンの色は、スライスそれ自体の色と一致する。スライスの色を変えるには、右マウスボタンでスライスのボタンをクリックする。赤、緑、青のスライダを持ったウィンドウから色を変更するためにスライダを動かす。

## 3. 10. 2. 6 ボリュームレンダリング

ボリュームレンダリングは半透明の色のもやのような立体視界を表示するための技術である。とはいえ、ある物理的変数のボリュームレンダリングは見られず、他の物が正確な色のマッピングで効果的に表示されている。コントロールパネル上のボタンの縦の6列はボリュームボタンである。一回に1つしか表示されない。ボリュームレンダリングが始動すると、カラーテーブルとともにポップアップウィンドウが現われる。このカラーテーブルは、上述の色付けされたスライスで描かれたのと全く同じ方法で使用されている。マウスかキーボードを使用し、データ値を色付き、透明にマッピングする機能を変えることができる。再度、シフトキーを押さえている間、マウスで曲線を描くか、上/下キーを押すかをすれば、透過伝送機構を変えられる。ボリュームレンダリングは以下のことを行う。

- \* 立体ボックスの軸が、視界の方向に対してほとんど並行に確定するため、現行で見える変形を調べる。
- \* データ値を色付き、不透明にマッピングする軸に対してperpendicularである、多くの色付けされたスライスを作る。
- \* 裏にある色付けされたスライスを前面の順番にする。頂点のアルファ値は、各スライスの間及びその内部での移行が順調に行われるように補間、調整される。

## 3. 10. 2. 7 速度ベクトル

速度ベクトルの軌跡は、速度のトンネル内で立体ボリュームの煙の線の痕跡を通して空気の動きを追跡する。軌跡モードにするには、コントロールパネル上のTRAJECTORYラジオボタンを選択する。

ポップアップウィンドウが画面のボタンの近くに現われ、立体カーソルがボックス内部に現われる。この立体カーソルは、新しい速度軌跡がどこに作られるべきか指定するために使われる。主コントロールパネル上のSTEPボタンも重要であり、それは軌跡を生み出す所のタイムステップを選択するために使用されるからである。速度ベクトルの軌跡は一セットで処理され、8つのセットがある。各セットは、セット1、セット2、...セット8とラベル表示されたボタンを持つ軌跡ウィンドウ内で表現される。各セットは個別に表示され、色付け、或いは消去される。新しい軌跡を作成する時、地点、時間等に対応するセットにグループ分けするとよい。

軌跡作成の最初の段階は、立体カーソルで位置を選択する。右マウスボタンを使って立体ボックス内部の立体カーソルの周辺をドラッグする。立体カーソルは軌跡に並行な面の中で平面に動く。左マウスボタンでビューボックスを回転させると、右マウスボタンでカーソルを設定する動作を交互に行うことによって、立体カーソルはビューボックス内部のどこにでも設定可能になる。

次に、コントロールパネル上のSTEPボタンでタイムステップを選択しなければならない。軌跡が作成される時、それは現在のタイムステップから最後のタイムステップへと順に進んで行き、時間内は最初のタイムステップに向かって後ろ向きに遡るようになる。

最終的に、現在のカーソル位置とタイムステップにおいて軌跡を作成するには、立体ウィンドウ内部に向かう時の中央のマウスボタンを押す。ラインセグメントとして軌跡が現われる。ANIMATEボタンをONにすることにより、軌跡がどのように時間と場所にまたがり移動するかを観察することができる。概して立体カーソルの位置決めと軌跡のセットの作成のため、中央のマウスボタンをクリックする過程を繰り返す。

ANIMATEボタンがONの時に軌跡ができることによって、軌跡は唯一つではなく全てのタイムステップごとに作成される。これは全ての空気の包みの経路を表し、その包みは場所の中の単一地点を通過する。以下は様々な軌跡の機能の概要である。

- \* 立体カーソルの配置には、左マウスボタンでビューボックスを回転させると、右マウスボタンで立体カーソルをドラッグするのを取り混ぜて行うこと。
- \* タイムステップの選択にSTEPボタンかANIMATEオプションを使用する。
- \* 現在のカーソルの位置とタイムステップにおける軌跡を作成するには、中央のマウスボタンを押す。
- \* 軌跡の表示の入切を切り換えるには、左マウスボタンでセットボタンをクリックする。
- \* 中央のマウスボタンでセットボタンをクリックすることにより、現在の軌跡のセットを選択する。
- \* 軌跡ウィンドウ内のDelete Setボタンで軌跡セットは消去される。
- \* 軌跡ウィンドウ内のDelete Lastボタンをクリックすることによって作成された最後の軌跡を消去することができる。

速度ベクトルの軌跡はラインセグメントカリボンの2つの方法で描かれる。軌跡ウィンドウ内のRIBBONボタンをクリックすることによってリボンを選択することができる。RIBBONボタンの切り換えは、新しい軌跡が表示されるかを制御するのみであり、既に作成された軌跡には効果がない。軌跡ウィンドウは、更にSTEPとLENGTHとラベル表示された2つのタイプイン規格を持つ。STEP値は、軌跡を辿るアルゴリズム内で使用されている工程のサイズを制御するために使われる。LENGTH値は軌跡の長さを制御するために使われる。1.0は各々に対する省略値の値であり、それらは各々乗数として作動する。もしも省略値のサイズの1/2の過程で統合するために等高線にする必要があれば、0.5の

工程値を入力する。軌跡を省略値の長さと同じく2倍にしたければ、2.0の長さの値を入力する。

軌跡の色は同一表面に対するのと同じ方法で制御される。それは軌跡のセットが、他の物理的変数によって単一色か、または色付きのどちらかになる。そのカラーウィンドウを提示するために右マウスボタンで軌跡セットボタンをクリックする。色でマッピングされた軌跡が見える時、軌跡の色が時間依存型であることは知られている。現在のタイムステップに対する他の変数に対する値によって、軌跡の頭部のみが色付けされる。軌跡の尾部がその他の変数の色によって色付けされる時、頭部はその位置にある。

### 3. 10. 2. 8 速度の変数

省略値によって、速度の軌跡と速度ベクトルの一番目のセットは、U、V、Wと名付けられた変数によって計算され、その間に二番目の速度ベクトルがU2、V2、W2と名付けられた変数によって計算される。他の変数は、コントロールパネル上の"UVW VARS"ボタンを通して指定される。このボタンをクリックすると、ポップアップウィンドウが現われ、そこで軌跡、及び一番目と二番目の速度ベクトルのセットの計算のために使用される変数の名称を指定することができる。指定は単に新しい変数の名前をタイプするだけである。有効な名称を入力しないと、動画化シミュレーション・プログラムは選択した図形を計算しない。新しい速度の構成要素の変数名を入力した後は、新しい値を使用してAPPLYをクリックし、かつウィンドウが見えるようにしておかなくてはならない。新しい値を使用しウィンドウを閉じるには、OKをクリックする。変換を廃止しウィンドウを閉じるには、CANCELをクリックする。更に動画化シミュレーション・プログラムの開始時に、コマンドライン上の速度構成要素の変数も設定することができる。

### 3. 10. 2. 9 テキストラベル

テキストラベルは、立体ビューウィンドウ内でイメージの注釈を付けるのに使用される。これは図形表示の作成用に使用される。題目、名称、日付を追加することができ、日付、データの意味がウィンドウ内に見える文書の特殊な機能を強調する。テキストラベルを入力するには、モードをコントロールパネル上のLABELラジオボタンで選択する。テキストラベルを作成するには、立体ウィンドウ内のどこかにマウスのポインタを位置付け、左マウスボタンを押す。縦のバーカーソルがその位置に現われた時、テキストをタイプすることができる。<Backspace>キーは間違いを訂正するために使用される。終了後、<Return>を押す。

テキストラベルを新しい位置に移動させるには、マウスでそれを指し、中央のマウスボタンを押さえドラッグする。マウスの動きに合わせて、マウスボタンを離すまでテキストの輪郭がポインタに引きずられる。テキストラベルを消去するには、マウスでそれを指し、右マウスボタンを押す。ラベルを消去する場合、消去確認のプロンプトがないので注意を要する。消去してしまったら、再度タイプすることによって保管し直すしかない。コントロールパネル上のSAVEボタンは、作成されたテキストラベルのどのようなものでも保護する。異なったフォントを選択するには、'font'コマンドオプションを使用する。



### 3.10.2.10 データ検証

立体ボリューム内で様々な位置における個別のデータ値を検査するためデータ検証が役立つ。コントロールパネル上のPROBEラジオボタンをクリックすると、立体ボックス内に立体カーソルが現われ、右マウスボタンを使用して動かすことができる。各物理的変数に対する現行のタイムステップの値が立体ウィンドウの左端にそって表示される。物理的な単位が変数に対して設定されたら、その値の隣に表示される。単位は、データ変換プログラム内のv5dSetUnits()機能で割り当てられる。

GRID#'sボタンをONにすると、その検証は一体型の格子座標に強制される。それは、カーソルが最も近い個々の格子座標に'snap' (はまる) ことである。

### 3.10.2.11 新しい変数の作成

コントロールパネル上のNEW VARボタンは、ボタンマトリックスに対する新しい物理的変数を追加するのに使用される。3種類の追加可能な新しい変数は以下の通りである。

- \*複製された変数：これらは実在する変数のコピーである。例えば、同じ変数の異なった同一表面を2つ同時に作成するには、この複製された変数を使用する。
- \*外部機能の変数：実在する変数の機能として新しい変数を計算するには、外部機能（自分で書き込んだ）を呼び出すことができる。
- \*計算された変数：実在する変数の値を格納する方式でタイプすることによって、新しい変数を計算することができる。

NEW VARボタンをクリックするとウィンドウが現われ、複製された変数と呼び出した外部機能を列挙し、新しい変数を計算するための方式でタイプさせる。新しい変数が作成された後は、新しい縦の行のボタンが、新しい変数としてコントロールパネルに追加される。他の物に似た変数の、同一表面、軌跡、スライス等を使用し作成できる。

#### (1) 複製された変数

Uの速度素子の変数を複製する場合、その同一表面を+20と-20の両方にすることになる。最初は、NEW VARボタンをクリックし、それからポップアップウィンドウからUを選択する。複製された変数はU'と名付けられる。このU'は他のどの変数とも同じで、その同一表面を作成するために扱われる。

#### (2) タイプイン方式

タイプイン方式は、実在する変数の機能として新しい変数を計算する数学的な表現でタイプさせる。例えば、U、V、Wからの速度速度を計算するためには、以下の数式を入力する。

$$SPD3D=SQRT(U*U+V*V+W*W)$$

気温に対する露点(TD)の制限を計算するためには、以下の数式を入力する。

$$RATIO=TD/T$$

数式は実在する変数、番号、算術操作の+、-、\*、/、\*\*（パラメータ関数）等の名称、及びSQRT、EXP、LOG、SIN、COS、TAN、ATAN（弧の接線）、ABS（絶対値）、MIN、MAX等の関数の名称を使用する。MINとMAXは、他の関数が全て1つの引数を取る間に2つの引数を取る。OKボタンをクリックして新しい変数を計算するか、CANCELで数式を廃棄する。NEW VARポップアップウィンドウから再度それを選択することにより、後に数式を編集することができる。

### (3) 外部分析機能

外部分析機能は高度な機能である。外部分析機能はFORTRANで書き込まれており、実在する変数の機能として新しい変数を作成するために動画化シミュレーション・プログラムによって呼び出される。例で挙げたように、SPD3Dの機能を含んでおり、 $SPD3D = \sqrt{U*U + V*V + W*W}$ のような速度の速度を計算する。外部関数機能は動画化シミュレーション・プログラムを経験済みのユーザーで、更にFORTRANの熟練プログラマに向いているといえる。

全ての外部関数は"userfuncs"（これは動画化シミュレーション・プログラム.h fileで変換される）という名称のディレクトリ内にあり、動画化シミュレーション・プログラム実行時に現行のディレクトリに対応する。例えば、"/usr/jones/data"内で常に動画化シミュレーション・プログラムを実行させるとするならば、分析関数は"/usr/jones/data/userfuncs"内になくなくてはならなくなる。更にこのディレクトリはスクリプト"externf"を格納しており、それは機能の編集に使用される。

外部関数の書き込みを行うには、提供されたサンプルの1つをコピーし、それを修正するのが一番良い。内蔵されている"userfuncs/example.f"は、この目的に対して説明がなされている。後に動画化シミュレーション・プログラム内部からその関数を呼び出す時に、それは各タイムステップ毎に1度起動される。その関数に通じる引数は以下の事項を含む。

- \* データファイル内の物理的変数の番号
- \* 各変数の名称
- \* 立体格子のサイズ
- \* タイムステップの日付と時間
- \* マップ投射と縦の座標系の情報
- \* 各物理的変数に対するデータの実際の立体格子

その関数は計算を必要とするのに必要なものを見出す物理的変数の名称のリストを操作する役割を果たさなくてはならなくなる。それは動画化シミュレーション・プログラムへ戻るデータの新しい格子を発生させる実際の計算を行わなくてはならない。これをどのように行うかを実演するサンプルを内蔵してある。関数引数の文書化を詳術するsample.fを特に見るとよい。マップ投射と縦の座標系引数は、v5dCreateライブラリ呼び出しと全く同じ方法で作動する。

その関数を"delta"と名付けたとするとFORTRANプログラムの名称が"delta.f"になる。"externf delta"とタイプすることによって関数をコンパイルすることになる。エラーが無ければ、実行可能なファイル"delta"が書き込まれる。その後、動画化シミュレーション・プログラム内でNEW VARを選択する時に、ポップアップウィンドウ内の機能一覧に"delta"が現われる。

外部関数に通じる格子データを得るために、動画化シミュレーション・プログラムに対して独自の

圧縮されてないMcIDASファイルか、または圧縮されv5d/comp5dファイルの2つの位置がある。圧縮されてないMcIDASデータの方が、より精密であるが故に優れているといえる。McIDASファイルが見つからない場合は、動画化シミュレーション・プログラムがメモリ内に所持している圧縮されたデータが外部関数に通じることになる。関数内部からデータ証明の位置と値を取り戻すことができる。データ証明の位置を得るには以下の（コマンド）を使用する。

```
CALL PROBEPOS (ROW, COL, LEV, LAT, LON, HGT)
```

格子座標内の位置は行、列、標高へ戻る。地理的座標の位置は緯度、経度、高度へ戻る。

実際の試験位置とタイムステップにおけるどの物理的変数の値を得るにも、以下の（コマンド）を使用する。

```
VALUE = PROBEVAL(VAR)
```

VARが、どの物理的変数を要望するかを設定する地点である。

### 3. 1 0. 2. 1 2 画像ファイルの保管と印刷

コントロールパネル上のSAVE PICボタンは、立体ウィンドウ内の画像をファイルへ保管するために使用される。SAVE PICをクリックすると、ファイル書式とファイル名を選択できる所にポップアップウィンドウが現われる。ファイル形式の選択は使用している機種によって異なる。動画化シミュレーション・プログラムによって支援される書式は以下の通りである。

XWD	xwud、またはxvでは役割を果たさない、Xウィンドウダンプ
RGB	ipaste、またはxvでは役割を果たさない、SGI画像ファイル書式
GIF	xv、または他の多数のプログラムでは役割を果たさない、標準GIF書式
PostScript	ghostviewのようなプログラムで画面上に印刷、または写し出される
Color PostScript	ghostview状のプログラムで印刷、または写し出される

動画化シミュレーション・プログラム（GLを使用する）のirix4とirix5の機器構成は、直接RGBファイルの書き込みを行う。GIFファイルの作成には、togifプログラムが必須である。PostScriptファイルの作成には、topsプログラムが必要である。togif、tops、その他多くのRGBファイルの変換プログラムは、IRIXの標準的である。もしも、それらが/usr/sbinにおいて見つからない場合は、IRIX CD-ROMからインストールする。動画化シミュレーション・プログラム（OpenGLを使用する）の他の全ての機器構成は、直接XWDファイルの書き込みを行う。RGBファイルの作成には、fromxwdプログラムが使用される。gifファイルを作成するには、fromxwdとtogifの両方（SGIシステム上でのみ使用可能）が必要である。無彩色スケールのPostScriptファイルを作成するには、xprの有用性（X11で規格化された）が求められる。PostScriptファイルを作成するには、topsプログラムが必要とされる（SGIシステム上でのみ使用可能）。

前述のどの有用性も持ち合わせてなければ、画像ファイルの変換のためにxvを使用してみるべきである。xvはcontrib/内のexport.lcs.mit.eduと、pub/ディレクトリ内のftp.cis.upenn.eduからのftpによって使用可能である。

動画化シミュレーション・プログラム画像を印刷するには、立体ウィンドウ越しにマウスポインタを設定し、Pのキーを押す。その動作の確認のためにプロンプトが来る。動画化シミュレーション・プログラムは、Post Script画像ファイルを省略値プリンター、またはPRINTER環境の変数によって指定されるプリンターへ送るため、lprを使用する。

### 3. 10. 2. 13 テクスチャマッピング

テクスチャマッピングは、コンピューターグラフィック用語で、3Dの表面上に2次元の画像を表示することを指す。動画化シミュレーション・プログラムでは地形図（地形図が消えているときは、立体ボックスのボタン）を重ねて、衛星や地図といった画像を表示することができる。テクスチャマッピングは、SGIシステムかメサ・ライブラリを使用したシステムでのみ利用でき、テクスチャマッピング用の支援ハードウェアを使用することを勧める。Vis5dには3種類のテクスチャ/画像マッピングがあり、コマンドラインで指定できる。

#### -area N

NはMcIDASエリアファイルの数列の最初の記号である。ファイル読み出しの数は自分のデータファイルのタイムステップ数と同じになる。画像は全て同じ大きさにするのがよい。再マッピングする必要がある場合、Vis5dを使わなければならない。

例：仮に自分のデータファイルが4タイムステップで、エリア100と指定したとすると、AREA0100、AREA0101、AREA0102、AREA0103がロードされ、かつ表示される。

このオプションには、SGIシステムでのみ利用できるMcIDASライブラリが必要である。

#### -sequencefile

これは-areaオプションのような働きをする。ただしデータが、McIDASエリアファイルではなく、非常に単純なファイル形式から来る場合はこの限りではない。ファイルは3 int'sで始まり、この3 int'sにはシーケンス中の画像の数、一画像に対するラインの数（ラインに対する画素の数）が含まれる。srcディレクトリのimage.cファイルにある読み込みread\_texture\_sequence機能は、このシーケンスファイルを読み込み、画像シーケンスファイルを作成したい人のためのファイル形式レファレンスとしての役割を果たす。

#### -texturefile

このオプションは画像を一つ指定し、それをタイムステップの間中、地形図にかぶせて表示するためのものである。このファイルはSGI RGB ファイル形式になっている。XVプログラムを使用して、自分の画像をRGB形式にコンバートすることができる。テクスチャマップが利用できるときは、コントロールパネル上のTEXTUREボタンで画像のオン、オフ表示を切り換えられる。

### 3. 10. 2. 14 Tclスクリプティング

動画化シミュレーション・プログラムにはスクリプト機能がある。つまり、Tcl言語のコマンドのテ

キストファイルで動画化シミュレーション・プログラムをコントロールできる。

### 3.10.2.15 キーボードの機能

マウスのポインタが立体ビューウィンドウの内側に入り込んでいるとき、以下に述べるキーボードの機能呼び出すことができる。

キー	機能
F1	コントロールパネルウィンドウを上下に移動する。このキーは-fullオプションのとき役に立つ。
F2	システム情報（使用中のメモリーも含む）の表示と計算されるグラフィック表示を切り換える。
P	現在のウィンドウの画像を印刷する。ポストスクリプトプリンタが使用可能でなければならない。自分のシェルのプリンタ環境変数を設定し、どのプリンタを使うか指定する。
S	動画の動きを遅くする。フレーム間の最小時間を10ミリセカンド増やす。
F	動画の動きを速める。フレーム間の最小時間を10ミリセカンド減らす。

自分独自のキーボード機能をプログラムしたい場合は、ファイル `src/gui.c` 中の `func1()`、`func2()`、`func3()` 等の機能を探ること。これらは、対応するファンクションキーを押すと呼び出される。

### 3.10.2.16 制限事項

動画化シミュレーション・プログラムのSGI版で複数のCPUを使用できれば、裏でグラフィックスの計算をし、動画化シミュレーション・プログラムの速度を速められる。他のシステムでは、動画化シミュレーション・プログラムは、ユーザーとのインタラクションでグラフィックスの計算を割り込ませようとする。この方法では、グラフィックス計算の処理が終わるまで、ユーザーインターフェイスが遅くなる。動画化シミュレーション・プログラムのユーザーインターフェイスを言葉で記述するのは、複雑かもしれないが、実はこれでも簡単なものにしようと努力した結果である。サンプルデータを使って少々練習すれば、スムーズに感じるようになると思われる。

### 3.10.2.17 v5dimportのユーティリティ

v5dimportのユーティリティは格子ファイルをv5d形式にコンバートするための新プログラムで、複数のソースファイルを組合せ、新しい座標系へ再サンプリングし、変数やタイムステップを選別する。グラフィックス・ユーザーインターフェイスとラインコマンド・ユーザーインターフェイスの両方がある。

例えば、v5dimportを使うとMcIDASGR3DファイルとMcIDASGRIDファイルを読み込み、全データのランバート・コンフォーマル投射への再サンプリング、CWATやVORT変数の省略、`lambert1.v5d` という名前の動画化シミュレーション・プログラムファイルへのデータの書き込みができる。

v5dimport使用時の事象の基本的な順序は以下のとおりである。

- 1) 入力ファイルの読み込み。
- 2) タイムステップ、物理変数、射影、または垂直座標系に従って出力用の格子を選択する。
- 3) 出力ファイル用に射影と垂直座標系を設定する。
- 4) 出力ファイルを書き込む。このとき再サンプリングが行われる。
- 5) 出力ファイル上で動画化シミュレーション・プログラムを起動してもよい。

v5dimportは以下のファイル形式を読み込むことができる。

McIDAS GR3DとGRIDファイル  
v5dとcomp5dファイル  
GRADSファイル  
"UW vis"ファイル (ウイスコンシン大学で使用されている)  
EPA MM4とRADMファイル (Clays上でのみ)

### (1) v5dimportの操作

シェルからv5dimportを起動する。

```
v5dimport [-path pathname] [files]
```

ここでは、[files] は入力ファイルのオプションリストであり、[-path pathname] は、"pathname"というディレクトリが、現在のディレクトリの場所で、入力ファイルブラウザ用、あるいは出力ファイル作成用に、省略時値として使われるように指定する。v5dimportをスタートさせると、メインウィンドウが現われる。その構成は以下のようになっている。

- 1) 入力ファイルからスキャンされた全格子のリスト。このリストはスクロールする。
- 2) 変数名、タイムステップ、投射、あるいは垂直座標系に従って格子を選択 / 選別するためのボタン。
- 3) 出力ファイルを記述し、作成するためのボタンやタイプインフィールド。

### (2) インプット格子の読み込み

いつでも"Read file..."ボタンをクリックして、追加の格子ファイルをv5dimportに読み込ませることができる。ファイルセクタを使って、自分のファイルを位置付けし、OKまたはCANCELをクリックする。

最初に入力ファイルを全て読み込んでしまうのがよい。というのは、新たなファイルを読み込むたびに、全格子が出力用に選択され、その前に行った選択が無効になっていってしまうからである。

"Discard all grids"というボタンは文字通りの作業をするボタンである。v5dimportから出て再スタートさせることになる。各入力ファイルを読み込んだ後、ウィンドウの上半分に表示される格子のリストは時間順に再分類され、その中で変数名順に再分類される。リストの桁は以下の通りである。

Grid	格子ナンバー
YYDDD	格子の年月日
HHMMSS	格子の時間 (時、分、秒)
Variable	変数名
Nr	格子の行数
Nc	格子の桁数
Nl	格子のレベル
Proj#	投射ナンバー ("Select by projection..."のウィンドウを参照する)
VCS#	垂直座標系ナンバー ("Select by VCR..."を参照する)
Filename	その格子が入っているファイルの名前

### (3) 出力用の格子選択

入力ファイルから、ある物理変数やタイムステップを捨てて、出力ファイルに書き込まれないようにしたいというケースがよくある。省略値によって出力用に全変数が選択される。

変数を選択／選別するには、"Select by variable.."ボタンをクリックする。全変数が載っているポップアップウィンドウが現われる。マウスを使って、欲しいタイムステップを選択、あるいは省略したいタイムステップを解除する。このボタンをホールドしている間は、マウスをドラッグするだけで、幾つかのタイムステップを選択／解除できる。最後に、"Select by projection ...", "Select by VCS..."ボタンで、マップ投射もしくは垂直座標系に従って、格子を取捨する。タイムステップ、変数、投射、VCSを選択するたびに、その影響を被った格子が、メイン格子リストで強調表示／強調表示解除される。"Select All", "Select None"は文字どおり、全選択、無選択で実行する。

#### (4) 出力ファイルの定義

読み込まれた最初のファイルから、出力ファイル（格子サイズ、投射等）用の省略時媒介変数が取り出される。自分の出力ファイルをつくる前に、媒介変数の値を変えなくてはならないことがよくあるため、必ずこの媒介変数を見直したほうがよい。出力ファイル用の行、桁、レベルの数を指定するには、メインウィンドウにある"Rows", "Columns"及び"Max Levels"と表示されたタイプインフィールドを使う。省略値が正しくない場合、新たな値をタイプする。出力ファイル用のマップ投射を見たり、変更したりするには、"Map Projection..."ボタンをクリックする。このポップアップウィンドウで、マップ投射の種類を選んだり、特定の投射媒介変数を入力することができる。"Guess" ボタンもあり、現在選択されている格子リストに与えられた出力投射で、適当なものを探すことができる。"Select by Projection"ポップアップウィンドウを画面に表示しておいて、出力投射と入力投射を比較することができる。出力ファイル用の垂直座標系を見たり、変更するには、"Vertical Coodinate System..."ボタンをクリックする。このポップアップウィンドウで、垂直座標系の種類を選んだり、特定の媒介変数を入力することができる。このウィンドウには"Guess" ボタンもあり、適当な省略値を探すことができる。"Select by VCS"ポップアップウィンドウを画面に表示しておいて、出力VCSと入力VCSを比較することができる。

#### (5) 出力ファイルの作成

メインウィンドウの底辺部にあるタイプインフィールドに、出力ファイルのファイル名を入力し、"Make"をクリックする。ファイルの変換が行われ、メッセージが印刷される。エラーが発生した場合、処理を停止する。入力格子座標系から出力用の新しい座標系ヘデータを再サンプリングする必要がある場合は、出力ファイルの生成には時間がかかることに注意する。

"Visualize"をクリックすると、ファイルができ、次に自動的にそのファイルの上に動画化シミュレーション・プログラムが立ち上がる（最初に"Make"をクリックしなくてよい）。タイプインフィールドにファイル名を入力する場合は、その名前を使用する。そうでなければ、自分のログインネームの後に".v5d"をつけたものを使う。動画化シミュレーション・プログラムコマンドで、コマンドラインオプションが欲しい場合、"vis5\_option"というファイルに入れる（例 "-mbs 64"）。

#### (6) その他

オプションのウィンドウを利用するには、"Options..."ボタンをクリックする。

最初の項目は、"combining of co-located data"を統制する。幾つかの立体格子が出力用に選択され、そ

れが空間と時間に関して同じ場所に設置されることは少なくない。出力ファイルに入れる値を計算する場合、その位置において、より解像度の高い格子からデータの値を選んでもよいし、あるいはその格子位置での全格子の平均値を取ってもよい。

二番目の項目は、格子データの出力ファイルでの圧縮法を統制する。省略値によって、格子の値は整数で1-バイトに基準化される。そうでなければ、よりよい解像度を求めて整数で2-バイトに基準化できるし、あるいは4-バイトの浮動小数点値を選択し、圧縮も基準化も行わないということもできる。このオプションは、ファイルサイズや精度のトレードオフを表している。

#### (7) v5dimportのテキストインターフェース

Xが使用できないとき、あるいはスクリプトでv5dimportを作動させたいときに、v5dimportへのテキスト/タイプイン (text/type-in) インターフェイスを使う。以下のテキストモードを入力し、v5dimportを起動する。

```
v5dimport -t [-path pathname] [files]
```

ここでは、[files] は入力ファイルのオプションリストであり、[-path pathname] は"pathname" という名前のディレクトリが、現在のディレクトリの場所で、入力ファイルブラウザ用、あるいは出力ファイル作成用に省略時値として使われることを指定する。テキストインターフェイスを通せば、自分のシェルのインポートリダイレクション機能を使うことによりスクリプトでv5dimportを作動することができる。

```
v5dimport -t <script
```

-tオプションでv5dimportを呼び出すと、>>プロンプトが表示される。そのプロンプトでは以下のコマンドを出すことができる。

exit	v5dimportを終了する
help	オンラインヘルプ
list	格子、タイムステップ、変数、マップ投影、垂直座標系のリストを表示する
read	入力ファイルを読み込む
keep/omit	格子、タイムステップ、変数、マップ投影、垂直座標系を出力ファイルに入れたり、そこからはずしたりするか選択する。
info	出力ファイル用に媒介変数を表示する
rows	出力ファイル用に格子行の数を指定する
columns	出力ファイル用に格子桁の数を指定する
levels	出力ファイル用に格子レベルの最大数を指定する
projection	出力ファイルのマップ投影を指定する
vertical	出力ファイルの垂直座標系を指定する
make	出力ファイルを作成する
visualiz	出力ファイルを作成し、動画化シミュレーション・プログラムを起動させる。

v5dimportにテキストインターフェイスを使用するのは、戦略という点ではグラフィックインターフェイスを使用するのと同じである。



- 1) 入力ファイルを読み込む
- 2) タイムステップ、変数、投射、and/orはVCSによって格子を選択する。この選択は一般的には一連のリストによって行われ、コマンドを無視したり、続行したりする。
- 3) 出力ファイルの媒介変数を設定/調整する。一般的には、インフォメーション、行、けた、レベル、投射、垂直コマンドといった一連のものである。
- 4) 出力ファイルを作成する。もしくは出力ファイルを作成し、動画化シミュレーション・プログラムを起動させる。

v5dimportのスク립トは、v5dimportコマンドと引数だけのASCIIファイルである。最も簡潔なケースでは、たった2、3のコマンドだけで成りたっていることもある。

```
例：# read my file, omit two vars, write v5d file
      read mydata.dat
      omit var CW
      omit var RW
      make outdata.v5d
      exit
```

v5dimportがスク립トを実行する際、各コマンドとその結果が印刷される。#で始まるラインは注釈とみなされ無視される。

#### (8) 新ファイル形式のための追加支援

v5dimportは、新しいファイル形式を読み込むための追加コードが簡単なものになるように書かれている。v5dimportのソースコードはimport/サブディレクトリの中にある。comment/\*\*\*ADD NEW FORMATSHERE\*\*\*を探し、新しいファイル形式を支援するには、どこにコードを追加すればよいかを見る。基本的に、2つの新機能を書き込む必要がある。一つはgrid\_info structsのリストを構築するためのファイル形式を走査するもの。もう一つは、自分のファイルからgrid\_info structsに与えられた実際の格子データを読み込むものである。これらの機能は、read\_foo.cという新ファイルに入れるとよい。ここでfooというのは、自分のファイル形式の名前である。そして、自分の機能を使用するために、file.cファイルをアップデートする。ガイドとして在るread\_\*.cファイルを使う。

#### (9) 特定のファイル形式に関する注意

記号EPAは、EPAファイルを読み込むことができるシステム上のみ、-DEPAというccコマンドラインに規定されている。EPAが提供するファイル読み込み機能はクレイコンピュータでしか作動しないので、現在はクレイシステムだけがEPAファイルを読み込める。111記号MCIDASは、libmcidas.a.ファイルを使用できるシステムにのみ、-DMCIDASというコマンドラインに規定されている。現在支援されているのは32ビット形式のSGIだけである。

## 4 データベース技術研究

地層処分ヴァーチャル・エンジニアリング・システムの、将来における要求事項を定義付けするために、高レベル放射性廃棄物の処分に関するモデル化過程について、現在開発されている技術から包括的な考察をした。その結果、以下の機能が情報の管理サービスとして考えられた。

- ・蓄積されているコードから、数値計算コードを選択するための指針情報を提供すること。
- ・数値計算を行うのに必要な入力データを検索し、グラフィック処理により数値計算を開始すること。
- ・性能評価シミュレーション個別部品のために数値計算結果を視覚化すること。

### 4. 1 蓄積されたモデル（コード）の管理

性能評価シミュレーション等のコードを管理する際の主要目的は以下の通りである。

- ・地層処分の安全性評価に係わるコードの分類と整理と行うこと。
- ・地層処分の安全性評価に係わるコードの出力を視覚化するソフトウェアと関連情報を体系化して整理すること。
- ・地層処分ヴァーチャル・エンジニアリング・システムの操作を支援するためのソフトウェアと関連情報を体系化して整理すること。
- ・地層処分ヴァーチャル・エンジニアリング・システムのシミュレーション用に選択された一つ一つのコードの機能に関する情報を整理すること。また、名称、著者、使用可能性、移植性、実験履歴などの追加情報を整理すること。
- ・参考文献について情報概略を提供すること。

表4. 1-1に、地層処分ヴァーチャル・エンジニアリング・システムのシミュレーション・コードの現象別に分類している整理例を示した。実際の性能評価にあたっては、人工バリアやニアフィールド等の場所による区分、これらの現象による区分、非定型的な区分（そのつど区分条件が決められる）に対応することが望ましい。従って、モデルとその関連情報に関するデータベースは、基本的には非定型的な論理条件検索に対応するリレーショナル・データベースが現時点では有利であると考えられる。

表 4. 1-1 コード整理の整理例

A. MECHANISTIC SOFTWARE	B. VISUALISATION SOFTWARE/DATA PROCESSING
1 Inventory Evolution	1 Engineering systems
2 Corrosion	2 Geographical Information Systems
3 Leaching	3 Geological Structure in 3D
4 Geochemistry	4 Data Visualisation
5 Geomechanics	
6 Thermal Processes	
7 Groundwater Flow	
8 Gas Generation	
9 Gas Transport	
10 Chemical Transport	
11 Biosphere	
12 Site Evolution	
13 Surface Hydrology	
14 Tectonics	
15 Catastrophic Events	
16 Radiolysis	

性能評価研究には数多くの分野があり、今後のモデル開発に利用する新しいツールを開発することが必要と考えられる。例えば、地下水の流動や地球化学などのプロセスについては、シミュレーションするための各種のモデルが存在するが、それに比べ、ニアフィールドの熱・岩盤力学影響を評価するモデルは比較的少ない。このように、必ずしも各分野で一様にコードが研究開発されているわけではないため、コードを選択する際に参考となるような基準の確立が必要である。このような、追加情報をコードや関連データとともに管理することにより、体系化されたモデルのデータベースの構築が可能となると考えられる。

## 4. 2 コード選択の基準

### 4. 2. 1 機能的な仕様（関連範囲、次元性、結合性及び適用性）

コード選択の際、モデルの適用範囲、関連性、及び将来性が重要であることは明らかである。モデルは、コードによって確実に適切に記述されなければならない。モデル間の相互作用やプロセスが相互に依存し合う問題は（例えば、地下水の流動、ガスの移動及び熱プロセスなど）、他のプログラムへ容易に連結できるように考慮することが必要である。

### 4. 2. 2 適用性（公的領域/私的領域、情報源や参考文書へのアクセス）

一般に、情報源に容易にアクセス可能であれば、研究がより進展する可能性が高まるので、最初は、パブリック・ドメイン（公開財産）のコード（例えば、NEA/OECD原子力機関データバンクから入手可能な）が評価用に選択されると考えられる。どのコードを使えばシステムに要求される事項が満足されるのかを決定するために、他のソフトウェア（プライベート・ドメインのコードも含めて）と比較して評価することも重要である。例えば、並列化の研究開発における研究の見通しが最も高いと思われるような、より進歩したソフトウェアがプライベート・ドメイン（私的財産）である場合がある。

地層処分ヴァーチャル・エンジニアリング・システムのシミュレーション・コードに含められるようなソフトウェアは、研究開発団体以外の組織にも入手可能でなければならない。また、対外的にも適用されるように支援するべきである。研究の現場では、国際的に受け入れられているコードが最も好まれている。なぜならば、文書化が未完成なプログラムは、適用基準を満たすことはできないと考えられるからである。完全に文書化されたソフトウェアにおいては、コードの機能、変数、データ要求事項、及び出力オプションを正確に記述すべきであることは言うまでもないであろう。

#### 4. 2. 3 ソフトウェア・エンジニアリング

プログラムのモジュール化によって、軽量化が容易になり、エラーも容易に認識されるようになった。現存のコードの多くは、FORTRAN言語によって書き込まれており、異なるプラットフォームに容易にインストールされる。今後、C/C++や図式的プログラム言語などの他のプログラム言語が数多くのこれらのFORTRANアプリケーションに取って代わるかもしれない。それは、これから数年の間に起きるであろう。新しいバージョンのソフトウェアは、マルチプラットフォームで動作するのに軽量であることが重要とみなされている。

ソフトウェア工学の検証及び確認とは、安全性がソフトウェアの品質を決定することを保証するための方法である。この文書の第5章に言及されているように、地層処分ヴァーチャル・エンジニアリング・システムプログラムに適用するための適切な方法が考え出されている。

##### (1) 実験履歴 (受入テスト、分析的/数値的検証、実験に基づく確認)

放射性廃棄物処分の安全性の評価用ソフトウェアを用いる際には、ソフトウェアもそれに関連するデータも理論的に正当な手順をふまえていることを明白に示さなければならない。

一般に、コードの検証とモデルの確認は、地球科学と物理科学とでは多少意味が異なる。ここでは、受入テストは、「組織内」のプログラムに適用される最低レベルの品質保証を表わしており、予め定義付けされている一連の受入基準を満たしている事を意味する（このプロセスは、ソフトウェア・エンジニアリングのアプリケーションにおいては、一般に「検証」と称されている）。

検証とは本来、数学モデルのために書かれた計算をそのモデルが実行するテストのことである。このことは、大規模な計算によっても証明できるが、もっとも一般的な手法は性能評価試験である。それは、共通な問題に対して得られた複数の結果を比較することである。

確認とは、予測と自然の現象とがほとんど一致することを意味している。しかしながら、自然界に本来備わっている不確実性により、モデルを明白に定量化して確認することはほとんど不可能であり、多くの場合は現実的ではない。従って、確認は、研究所における実験結果と、観察結果をモデルが正確に反映していることを実証するような、定性的及び定量的な証拠とみなされることがある。

### 4. 3 解析コードの管理

一般に前述の基準に基づいて、数多くのコードが開発蓄積される。コードについてのこれらの予備的な推奨事項の概要を表4. 3-1に示す。

表4. 3-1 推奨コード概要

CODE CATEGORIES		CODE RECOMMENDATIONS		
		Fall Back Code (Supporting Tools)	Preferred Approach	Coupled Processes Considered
<b>A</b>	<b><u>MECHANISTIC CODES</u></b>			
1	INVENTORY EVOLUTION	ORIGEN2	Determined by PNC	
2.	CORROSION	WAPDEG	<i>Required</i>	
3.	LEACHING (glass dissolution)  (other waste forms)	Possibilities: GLASSL GLADIS  WAPPA	<i>Required</i>	Corrosion and Leaching
4.	GEOCHEMISTRY	PHREEQE Supporting Tools: EQ3/6, MTDATA, MT3D, SOLISOL, HSC	New code based on PHREEQE/ PHRQPITZ with improved functionality	Speciation and Mass Transfer
5.	GEOMECHANICS	GEOSIM  TOUGH2 with SPECTROM  UDEC	Required	GW flow-thermal-mechanical  (Thermal-GW flow)-(Thermal-mechanical)  GW flow-thermal-mechanical
6.	THERMAL PROCESSES	TOUGH 2  SWIFT II  CFEST	TOUGH 2	GW flow-thermal-gas transport  GW flow-m transport-thermal  GW flow-m transport-thermal
7.	GROUNDWATER FLOW	TOUGH2  CFEST (porous)  (Supporting Tools): 3D FEMWATER (porous) FRACFLOW (fracture) MODFLOW	TOUGH2 (fracture/porous)  SWIFT II (fracture/porous)  (NAMMU) - porous  (NAPSAC) - fractured	GW flow-thermal-gas transport  GW flow-m transport-thermal  GW flow-m transport-thermal  GW flow-m transport  Groundwater flow

表 4. 3 - 1 推奨コード概要 (続き)

CODE CATEGORIES		CODE RECOMMENDATIONS		
		Fall Back Code (Supporting Tools)	Preferred Approach	Coupled Processes Considered
<b>A</b>	<b><u>MECHANISTIC CODES</u></b>			
8.	GAS GENERATION	GAMMON GLEAM	<i>Required</i>	microbial degradation-metal corrosion
9.	GAS TRANSPORT	TOUGH2 PORES	<i>Required</i>	GW flow-thermal-gas transport
10.	CHEMICAL TRANSPORT	CHEMTARD	LEHGC SWIFT II	Chemical transport - (flow) Hydrology - (transport)
11.	BIOSPHERE	BIOPATH/ PRISM	JESS + HSPF	Speciation + near surface processes
12.	SITE EVOLUTION	FFSM Supporting Tools: TIME4	<i>Required</i>	
13.	SURFACE HYDROLOGY	HSPF	JESS + HSPF	Speciation + near surface processes
14.	TECTONICS	Not available	<i>Required</i>	
15.	CATASTROPHIC EVENTS	No code recommended	<i>Required</i>	
16.	RADIOLYSIS	Possibilities: FACSIMILE CHEMISIMUL	<i>Required</i>	

表 4. 3-1 推奨コード概要 (続き)

CODE CATEGORIES		CODE RECOMMENDATIONS	
		Fall Back Code (Supporting Tools)	Preferred Approach
<b>B</b>	<b>VISUALISATION SOFTWARE</b>		
1.	Engineering Systems	AVS/EVS	AUTOCAD
2.	Geographical Information Systems	ARC INFO / REGIS	Required
3.	Geological 3D Structure	STRATAMODEL / EARTH VISION	Required (VIS-AD?)
4.	Data Visualisation	Various	Influence Explorer

表 4. 3-1 では、各分野にそれぞれ 2 つのコードを選択することを試みた。短期的にソフトウェアを表わすのであれば、「Fall Back Code」が最適である。Fall Back Code の適切な選択は、コードが即座に適用可能であるかとか、信頼性のあるものであるかどうかとか、望ましい実験履歴を有しているかなどに基づいている。「Preferred Approach」には、地層処分ヴァーチャル・エンジニアリング・システムにおいて研究開発が進む可能性が最も高いと思われるコードを選択した。

表 4. 3-2 は、表 4. 3-1 に関するより細かな情報の例である。これらの情報は、地層処分ヴァーチャル・エンジニアリング・システムを利用して問題解決にあたる場合に有益な情報となると考えられる。

表 4. 3-2 推奨コード情報

CODE CATEGORIES	Databases	Data Requirements
1 INVENTORY EVOLUTION	Decay database <i>data associated with each disintegration of a nuclide</i> Fission database Cross section database	$\alpha$ , $\beta$ and $\gamma$ energy (MeV), the number of neutrons emitted by spontaneous fission or from ( $\alpha$ , n) reactions in oxygen (in oxide fuel), the $\gamma$ energy spectrum (MeV) and multiplicity (number of photons in a given $\gamma$ energy band or group), radioactive decay constant, modes of decay, branching fractions. Fission yield data (fission yields of the fission products) Irradiation dependent cross-section libraries categorise cross section data in terms of actinide, fission product and (optionally) structural material nuclide libraries.
2 CORROSION	Geometric database  Water / gas phase properties  Flow and transport properties  Chemical Properties	Repository depth determines hydrostatic pressure and temperature; cavern shapes, dimensions and extent of Engineered Damage Zone (EDZ) determines fracture density and principal stress orientations. The rock type and nature of backfill determines porosity and permeability.  The characteristics of the gas and water phases, especially permeability and capillary pressure, which are strong functions of liquid saturation.  Porosity, hydraulic head and permeability of the host rock, of the disturbed zone, and of the repository lining and backfill. Diffusivities in the groundwater and backfill and water flow rates.  Metal corrosion rates. Chemical data including solubilities of radionuclides and matrix and concentrations of corrosive agents, sorption coefficients, redox conditions, pH and complexing agents.
3 LEACHING	Geometric database Properties database	Waste package geometry, characteristics of the waste form, list of materials Radiation decay and shielding properties, thermal and mechanical material properties and empirical data for radiation damage, radiolysis, corrosion and leaching.

表 4. 3 - 2 推奨コード情報 (続き)

CODE CATEGORIES	Databases	Data Requirements
4 GEOCHEMISTRY	<p><u>Thermodynamic database</u> Speciation and solubility data</p> <p>Surface complexation data</p> <p>High salinity data</p> <p>Temperature data:</p>	<p>Stoichiometry, log K, and <math>K_{sp}</math> constants, standard enthalpies, heat capacity and redox potentials, element concentrations for the aqueous phases, any required reactant mineral or gas phases, temperature.</p> <p>Specific surface area of the sorbent, capacitances of the surface layer, concentration of sites and a set of equilibrium constants for the surface reactions.</p> <p>Specific interaction coefficients.</p> <p>Heat capacities, standard enthalpies or the entropies of species at 298.15K.</p>
5 GEOMECHANICS	<p>Geometric database</p> <p><u>Stresses database</u></p> <p>Geological characterisation</p> <p>Geophysical characterisation</p>	<p>Waste package geometry, characteristics of the waste form, list of materials</p> <p>In-situ stress measurements as a result of gravitational, tectonic, structural and residual effects.</p> <p>Data on rock types from historical site records. Fracture density, orientation, roughness amplitude, mineralogy and hardness of fillings etc from site testing.</p> <p>Rock stiffness, strength and failure, moduli of deformation, Poisson's ratio, coefficient of friction and temperature effects (thermal conductivity and coefficient of thermal expansion of the host rock)</p> <p>Effects of excavation and concrete / backfill characteristics</p>
6 THERMAL PROCESSES		<p>Material properties of the medium, including density, specific heat capacity and conductivities.</p> <p>Heat source and hydraulic parameters, initial temperature distribution.</p> <p>Parameters for relative permeability and capillary pressure functions.</p>

表 4. 3 - 2 推奨コード情報 (続き)

CODE CATEGORIES	Databases	Data Requirements
7 GROUNDWATER FLOW	<p>Geometric database</p> <p>Matrix properties</p> <p>Water properties</p>	<p>1D, 2D or 3D geometry, repository dimensions</p> <p>rock type, fracture density and orientation, porosity, permeability mineralogy, backfill properties.</p> <p>Quantity, density, viscosity, diffusion coefficient, compressibility, salinity, fluid velocity, temperature, pressure.</p> <p>Hydraulic properties of each geological layer (hydraulic conductivities, specific storage)</p>
8 GAS GENERATION	<p>Radiolysis database</p> <p>Corrosion database</p> <p>Microbial degradation</p>	<p>See Radiolysis.</p> <p>See Corrosion data specifically related to corrosion of metals (steel).</p> <p>In most models, the formulation of equations is treated empirically and requires many parameters. Data includes carbon sources, pathways and sinks in microbiological food chain; effects of pH, temperature and moisture content; hydrolysis rate of waste, growth rates of microbes and mass of biomass carbon formed per mass of substrate metabolised.</p>
9 GAS TRANSPORT	<p>Geometric database</p> <p>Gas properties</p> <p>Water properties</p> <p>Matrix properties</p>	<p>Repository depth determines hydrostatic pressure and temperature; cavern shapes, dimensions and extent of EDZ determines fracture density and principal stress orientations. The rock type and nature of backfill determines porosity and permeability.</p> <p>Density, viscosity, solubility in water, binary diffusion coefficient, diffusivity coefficient, saturation vapour pressure, and compressibility. Gas properties are a function of temperature and pressure</p> <p>Quantity, density, viscosity, diffusion coefficient, compressibility, conductivity, salinity.</p> <p>Stress state, rock type, density, fracture density and orientation, porosity, permeability, mineralogy, strength, modulus of deformation, Poissons ratio, coefficient of friction, thermal conductivity, coefficient of thermal expansion, specific surface area, cement/backfill properties.</p>



表 4. 3-2 推奨コード情報 (続き)

CODE CATEGORIES	Databases	Data Requirements
10 CHEMICAL TRANSPORT	Chemical data	Equilibrium constants for aqueous complexes and surface reactions, stoichiometry, solubility products, redox potentials, concentrations of aqueous species/ elements, constants for ion exchange and thermal models.
	Transport-decay data	Dispersivity and fluid velocity, species dependent molecular diffusion coefficient, charge, radioactive decay half-life / radioactive decay constant.
11 BIOSPHERE		Transfer coefficients (rate constants) between the specified compartments in the biosphere (first order); radionuclide(s) half-life; parent nuclides; source of activity; masses and volumes of compartments; dose function specifications; number of exposed groups for which doses shall be calculated; biological data (uptake factors); dose conversion factors; consumption data; irrigation data; population data. Speciation and solubility data - see Thermodynamic database.
12 SITE EVOLUTION	Site definition database	
	Point Data	Coordinates, topography, stratigraphy.
	Climatic Data	4 global climate states considered (interglacial, glacial-temperate interstadial, glacial-interstadial, glacial-stadial); mean annual precipitation, temperature pdfs and degree of seasonal variation required;
	Scalar Polynomials	Single variables, eg. Diapirism rate, repository depth. Describe relationship between interactive processes in the submodels (eg. Sea-level and global ice volume).
	Probability Density Functions Constant Data	Required for each temporally changing parameter. Time invariant parameters such as time of change of stress regimes.

表 4. 3-2 推奨コード情報 (続き)

CODE CATEGORIES	Databases	Data Requirements
13 SURFACE HYDROLOGY		Meteorological records of precipitation and estimates of potential evapotranspiration are required for watershed simulation. Air temperature, dewpoint temperature, wind, and solar radiation are required for snowmelt. Air temperature, wind, solar radiation, humidity, cloud cover, tillage practices, point sources, and (or) pesticide applications may be required for water-quality simulation. Physical measurements and related parameters are required to describe the land area, channels and reservoirs.
14 TECTONICS		Specific data unknown; will include seismic forecasting data (displacement (frequency and magnitude) and associated disturbance to solid geology and hydrogeology).
15 CATASTROPHIC EVENTS		Specific data unknown; will include forecasting of human intrusion (accidental - engineering; deliberate - terrorism/war), meteorite impact (probabilistic forecasting).
16 RADIOLYSIS		G values for $\alpha$ , $\beta$ and $\gamma$ radiation (rate of radiation production per energy input) - temperature dependent. Dose rate (energy input per unit time), radiation yields, radioactive decay constant, activity, decay heat of isotope / heat production per unit time. Chemical reactions and kinetic rate constants for each chemical reaction.

現在の時間スケール内では、すべての利用可能なソフトウェアを決定的に評価することは出来なかった。しかしながら、地下水の流動に関するコードや地球科学的なコードが、選択基準 (上記した) の適用方法を詳細に説明するものとして検討された。新規開発コードの機能仕様をより詳しく説明していくことが長期的に必要とされるであろう。選択された各コードの機能仕様は、以下の情報を提供

することが望ましい。

- Program Summary
- Modeling Capabilities
- Testing History and Applications
  - Verification and Validation
  - Types of Use
- Program Operation
  - Program Language
  - Input Requirement
  - Output Facilities including Graphic Display
  - Machines on which the program has been used
  - Coupling
- Program Availability
- Known Users
- References

### (1) 核種インベントリー

核種インベントリー・コードは、多種の核種（アクチニド、核分裂生成物）のモル（グラム分子）数の経時変化を算出するものである。例えば、次の2つコードがある。ORIGEN2（パブリック・ドメイン）及びFISPIN（プライベート・ドメイン）である。本質的には両コードとも同じ機能であるが、FISPIN（ウェブスター1995年）は、特定のアプリケーション用に開発されたコードで、英国における核種インベントリーに焦点を合わせたものである。これに対して、ORIGEN2（CROFF 1980年、1983年）は、より大規模なソフトウェアシステムの一部である。モデル化能力、適用性、及び順応性を有するORIGEN2の方が、放射性廃棄物処理に関する研究用に適したコードである。

### (2) 腐食

好気的環境では、ステンレス鋼も炭素鋼（軟鋼）も腐食し、水酸化（第三）鉄を生成する。嫌気的環境では、腐食の発生後に磁鉄鉱（ $Fe_3O_4$ ）もしくは水酸化鉄が生成すると考えられる。両方の環境において、それに対応する陰極が半減し、水素ガスが発生する。嫌気的環境における、鋼鉄の一般的な腐食に関するデータを入手することは可能であるが（Marsh 1988年a, b）、温度、湿気、塩分などの要素によって決定される信頼できる予測に関しては、さらに研究が必要である。このことは、地層処分ヴァーチャル・エンジニアリング・システムにそのまま応用できるような包括的なモデルが存在しないことを説明している。

### (3) 浸出

数多くの一般的な廃棄物パッケージの劣化コードが適用可能である。これらのコードには、WAPP A（インテラ社1983年）やWAPDEG（LEEとATKINS 1995年）が含まれ、両コードともさらに研究を進めていくことが必要とされている。より機械的なモデルは、ガラス溶解やセメント固化廃棄物体の劣化などのような、特定の処分概念に限定されている。これらのモデルは、定常的な平衡化学コードを修正したものに依存する傾向がある。

### (4) 地球化学

長期的には、ニアフィールドの化学は、人工バリアシステムの性能に大きな影響を及ぼすと考えら

れる。金属製容器の材質の腐食、粘土埋め戻し材の安定性、ガラス廃棄物体の溶解、溶解した放射性核種の可溶性/分種化などが、ニアフィールドの化学的条件の直接的な機能である。地球化学モデルの重要な構成要素には、平衡状態での分種化、溶解度の推定、及び反応経路の算出などが含まれている。表4. 3-2に、検討された地球化学コード概略が示されている。

PHREEQE (PARKHURST 1980年) は、機能性に優れ、適用性の高さから、地球化学のモデル化に最も広く使われているコードの一つである。そのコードのバリエーションには (例えば、BROWN など1991年; ARTHURとTYRER 1996年) 有機錯化、表面における錯化 (吸着)、塩水におけるシミュレーション、及び温度上昇などのプロセスも含まれている。PHREEQE以外の地球化学コードについては、鉱物 (無機化合物) の溶解もしくはバリア材の劣化などについて動力的に算出することが必要とされている。PHREEQE以外の地球化学コードで最も広く用いられているのが、EQ3/6である (WOLERY 1979年、1983年、1992年)。これらのコードや同様の機能を有した数多くのツールが世界的プロジェクトであるCHEMVALプロジェクトにおいて、広範囲に試行されている。

(READとBROYD 1989年; READ 1990年、1991年、1994年; READとFALCK 1996年)。

#### (5) 地球力学

安全性の分析での重要な側面として、岩体の安定性、地殻変動による応力、処分場における地下水の流動及び熱負荷などの連結状態の評価が挙げられる。廃棄熱が割れ目に及ぼす影響や、それが岩体の透水性にどのような影響を及ぼすのかを把握するために、熱・水理学的・機械力学的 (THM) 連成コードを適用することが推奨されている。国際的な共同研究プロジェクトである、DECOVALEX (熱・水理学・機械力学連成モデルの開発及び核廃棄物の隔離に関する実験の確証) (JINGなど1995年) が開始された。DECOVALEXの全体的な目的とは、放射性核種の処分場から生物圏への流出及び移行に関してTHM連成プロセスがいかに重要であるか、そしてこれらのプロセスが数学的モデルによってどのように記述されるのかについて、一層理解を深めることである。このプロジェクトによって開発される、ベンチマークテストやケース問題によって、地層処分ヴァーチャル・エンジニアリング・システムに使われるコンピュータ・コードを適切に選択することが出来るようになると思われる。

#### (6) 熱プロセス

廃棄物容器や周囲岩石の中の温度を予測するには、熱伝達の計算が必要となる。温度は、これらの閉じ込め性能及び流出性能に直接的かつ間接的な影響を及ぼす。熱伝達に左右される地下水のシステムを評価するために、いくつかの限定されたコードを適用することが出来るが、その中には、TOUGH2 (PRUESS 1991年)、CFEST (GUPTAなど1987年)、SWIFT II (REEVESなど1986年)、SPECTROM (SVALSTAD 1991年) などが含まれている。これらのコードは、応力、流れ、放射性核種の移行、及びガスの移動能力などを連成した利点を備えている。

#### (7) 地下水の流動

概念化した多孔質媒体を通るダルシー流動のモデルについては、多くの文献があり (ASHTONなど1993年)、今回のプロジェクトの初期的評価段階に使用可能な候補には事欠かない。亀裂における流

動ツールについては、まだ開発がそれほど進んではおらず、種々の能力が異なるコード（パブリック・ドメインにおいてもプライベート・ドメインにおいても）を検討した。その結果、選択されたコードの能力、制限事項、有用性をリストにまとめた。亀裂が生じている多孔性媒体における流動体の流れを説明する概念的なモデルは、近接した多孔性媒体の貯蔵力と流動能力に基づいて分類することが出来る。主な4つの概念モデルを以下に示す。

- ・ 明瞭で不連続な亀裂（例えば、SWIFT）により、亀裂と多孔質媒体間の流体の潜在的な流動や勾配を最小限の非物理的なパラメータで説明することが出来る。しかしながら、数多くの亀裂に関するデータを得る事は容易でない。
- ・ 二重連続体モデル（例えば、TOUGH2）では、亀裂の生じていない岩体は媒体の空隙率（貯蔵力）が高く、透水性（流動）は非常に低いとする観察に基づいている。反対に、亀裂の生じた岩体の空隙率はごくわずかではあるが、透水性は高い。多孔性媒体と亀裂は、それぞれ2つの別々のものであるが重複している連続体としてみなされる。この2流動体間での流体質量移転は、亀裂・多孔性媒体の境界で起きている。
- ・ 非連続的亀裂ネットワーク（例えば、NAPSAC HERBERTとSPLAWSKI 1990年）は、流動が亀裂に限定されている、二重連続体としてのモデル分類がなされている。
- ・ 単一等質連続体モデルの重要性は高いとみなされ、平均的には、透水性は亀裂と多孔性媒体の透水性の合計と考えられている。

流体流動を概念化することに共通事項として挙げられるのが、問題に関連した時間的空間的尺度を決定することである。ニアフィールドの問題において、亀裂における流動について個別のモデルが必要とされる一方で、ファーフィールドの問題には、単一の連続体モデルが適用される。

TOUGH2 (PRUESS 1991年) は、不飽和、もしくは部分的に飽和しているマルチフェーズの流動解析を扱うプロジェクトにおいて、処分対象地層の様々な特性を検討し、またその特性を高レベル放射性廃棄物処分場の性能評価に適用する場合に、適した解析コードとして使用されてきた。TOUGH2は、また並列化の点でも発展の可能性が高い。

また、亀裂の生じている多孔性媒体中の地下水飽和流動についての解析では、SWIFT97が、地層処分ヴァーチャル・エンジニアリング・システムやその他の一般的な廃棄物処分研究に推奨される。

## (8) ガスの発生

次の2つの理由から、放射性廃棄物処分場におけるガスの発生について考えることが必要である。まず第一に、高圧ガスが通風孔のない塞がっている廃棄物キャニスタの中で発生して人工バリアを破壊してしまうかもしれないことである。第二に、溶解ガスや分離ガス相によって、移行時間が短縮するため、核種移行能力が増大してしまうことである。ガスの発生の重要なプロセスには、金属の化学的腐蝕（好氣的及び嫌氣的）、微生物による劣化、熱による劣化及び放射性分解などが含まれる。ジョーンズ（1990年）の調査文献によると、ニアフィールドにおけるガスの発生をシミュレーションするためのコンピュータモデルを適切に文書化したものが一般的には不足していることがわかっている。一般的に、ガスを発生させるかもしれないプロセス毎にモデルは限定される。そして、場合によっては、有機分解のプロセスに関連したガスの発生について明確に考慮されていないこともある。最近、GAMMON (PURDOMとAGG 1993年) というコンピュータプログラムが開発されたが、それは微生物による劣化と腐蝕のモデルを組み合わせようとするものであった。

### (9) ガスの移動

処分場内部でのガスの発生と移動が地圏における放射性核種の流動性に影響を及ぼす恐れがあることが考えられる。処分場を經由したガスの移動や処分場からのガスの移動には、相互に関係し合う広範囲のメカニズムが寄与している。これらのメカニズムを扱う手法は、ガスの移動及びマルチフェーズ流動、地球化学的影響、生物地球化学的影響の3つの分析方法に分けられる。ガスが移動する方法は、処分場の環境における水圧に強く影響を受ける。

### (10) 地質学的影響

毛管流動が主要なガス移動メカニズムであるならば、毛管圧力が現存の亀裂をさらに拡大させることになるかどうかを知ることが必要である。さらに処分場の周囲の地下水とガスの流動経路は、岩体の応力分布によって決定され、また、処分場掘削時やその後の密封などに対して岩体の体積が時間の経過とともに反応することによっても、処分場の周囲の地下水とガスの流動経路は決定される。従って、岩体や埋め戻し材の性質について、詳しく理解することが必要になる。

生物化学的影響については、溶存した状態もしくは分離状態のガスの移動が、化学的吸着により遅れる可能性もある。ガス移動メカニズムを完全にモデル化することができれば、亀裂が生じたストレスの変化の影響を受けやすい代表的な岩体マトリックスを介して、溶存状態もしくは分離状態のガスが移動出来るようになる。その際、吸着のような遅延要因も考慮に入れられている。現在のところ、このタイプの概念モデルを適用することは出来ない。システム全体の個別の現象を解析するためのモデルは存在する。適用可能なコードのうち、TOUGH2とPORES (PONTINGなど1983年; SCOTT 1990年)の2つのコードがガス移動のマルチフェーズ的な流動をモデル化するためとして認識されている。両方のコードともパブリック・ドメインのコードであり、亀裂の生じている岩体をモデル化するのに、二重空隙率モデルか透水性モデルかどちらかを選択することができる。PORESは、油層をシミュレーションするものであるが、上記に関連した二相の流動に関する問題にも適用可能であろう。

### (11) 汚染物質の移行

放射性汚染物質の移行をシミュレートするためには、同時に作用している地球化学的プロセスと水理学的プロセスを正確に理解することが必要である。

現在のところ適用可能なツールは、CHEMVALプロジェクト (READ 1991年; READとFLACK 1996年)において評価されている。また、CHEMTARAD (BENNETTなど1991年)は、適切に文書化されており、広範囲に検証された化学連成移行コードである。また、このコードは、パブリック・ドメインで入手可能である。しかしながら、初歩的な数値解析アルゴリズムに基づいており、一次元的なシミュレーションに限定されている。LEHGC (YEH 1994年)は、飽和媒体及び不飽和媒体中の地下水化学的移行を解析する最新の有限要素法モデルである。CHEMTARADとLEHGCの両方のモデルを使用して検討可能なプロセスは、地下水の流動(移流、分散/拡散)、アクア錯化、イオン交換、吸着、沈殿、溶解などである。HYDROGEOCHEM (YEHとTRIPATHI 1991年)によって後者のコード(LEHGC)が適用されて以来なされた改訂・改良は適切に文書化されている。LEHGCは、並列化に適用しやすいコードであり、将来発展することが期待される適切なコードと思われる。

## (12) 生物圏

放射性物質に関する評価の全体的枠組みでの生物圏モデル化は、空洞や地圏の進展をモデル化することよりも関心が低い。一般的に、現在の方法論では、簡単な分画的なモデルが用いられ、多くの生物圏の経路やプロセスが削減されて、前もって決められている個体群への被曝量を概算するための線量換算要因となっている。

適用可能で柔軟性のあるツールのモデルの中から、BIOMOV5プロジェクト (BIOMOV5 1996年) において、開発及び実験が継続中のBIOPATH/PRISM (BERGSTROM他 1982年; GARDNER他 1983年) が評価用に選択された。しかし、複雑な生物圏の経路が緻密な構造の中で簡素化されているために、これと同様の分画モデル制限が多いものになっている。従って、生物圏に関する問題に分種化に基づいた手法を適用して、各分画 (例えば、土壌、堆積、入江など) における主要な放射性核種の移動性や有毒性を限定してはどうかと考えられている。

高度な分種化モデルシステムである、JESS (分種化についての共同専門システム) (MURRAYとMAY 1995年) は、生物圏における化学作用をモデル化するための強力で、柔軟性のある予測ツールとして推奨されている。このシステムは、非常に包括的なデータベースによりサポートされている。気候、植物、土壌及び水理学との複雑な相互作用について検討する必要性があるので、機械論的方法で近地表環境に適用できる統合化モデルを構築するために、JESSと適切な地表水理学パッケージを組み合わせる提案が出されている。

## (13) サイトの変化

サイトの変化モデルの目的は、廃棄物隔離システムにあるひとつ以上の自然バリアの有効性に影響する一定の事象とプロセス (自然及び人間によって引き起こされる) を定量的に表わすことである。2つのサイト変化コード、すなわちTIME4 (DAMESとMOORE 1991年) とFFSM (インテラ社1983年b) を検討した。

TIME4は、気候によって引き起こされる環境の変化については検討しているが、地殻変動や生物発生のプロセスによる影響及び人間によって引き起こされる環境の変化などについては検討されていない。TIME4のパラメータの大部分は、生物圏の生物学的特性よりはむしろ物理的特性について多く記述している。

FFSM (ファーフィールドにおける状態モデル) は、確率論的枠組みの自然バリアにおいて自然、及び人間によって引き起こされる変化に適用される比較的古いタイプのモデルである。将来、多種の時間枠 (例えば、腐蝕プロセスについては0年から100年、再飽和については100年から1000年、氷河作用については1000年から10000年) における重要なプロセスを表記するために、規模を拡張した変化モデルを適用することが望ましいと考えられる。

## (14) 地表水理

地表付近の水理コードを選択する際、一般的なコード選択基準 (有用性、実験履歴、次元性など) を越える多数の追加機能を検討した。特に一連の気候条件下での地表水のプロセスをシミュレーションするために、モデルが使用可能かどうかを検討した。合計12個のよく知られているコードから、詳細に検討するために2つのコードとは、SHETRAN (EWEN 1990年) とHSPF (BICKNELLなど19

93年)を選び出した。SHE(欧州水理学システム)はヨーロッパにおいて広範囲に適用されているが、その最も高度な形態のコードはパブリックドメインにて公開されていない。HSPFは、USGSとUSEPAによって承認されたパブリック・ドメインにて公開されている地表水理コードであり、現段階においては地層処分ヴァーチャル・エンジニアリング・システム用のコードに利用可能と考えられる。

#### (15) 地殻変動

現在までのところ、地質時代における地殻変動による圧縮が土壌に及ぼす影響をモデル化する数値コードは確認されていない。もし、その必要性があると考えられるのであれば、この分野における研究及びコードの開発をさらに進めていく必要がある。

#### (16) 天災事象

地震や火山の噴火などの天災事象が引き起こす地圏への影響をモデル化するのに適しているコードは、地層処分ヴァーチャル・エンジニアリング・システムプロジェクトの第一段階では検討していない。この目的のために「既製品」コードがあるならば、あるいは開発が必要とされているならば、さらに調査研究を進めていく必要があると考えられる。

#### (17) 放射性分解

高レベル放射性廃棄物処分場において、放射線分解は、廃棄物容器の近辺の地下水の化学的性質を変化させることがある。その結果、容器の腐蝕率や廃棄物の溶解度、及び放射性核種の可溶性に影響を及ぼすこともある。放射線分解により劣化しやすい材質は、イオン交換樹脂、アスファルト、プラスチック、コンクリートなどである。コンクリートは空隙水が劣化しやすい。高温高湿下の一連の酸化還元条件下では、放射線分解によるガスが発生しやすく、特にコンクリートについては、水の劣化によりガスが発生しやすい。水の放射性分解によって、基やイオンが結合して反応することにより、数多くの末端元素(過酸化水素、水素、酸素などを含む)が形成される。これらの新しい元素の発生は、吸収されたエネルギーとの関係で、G値を使用して定量化される。これらの値は、100Evのイオン化した放射線を吸収した結果、形成されるある一定の種の分子の数として定義付けされる。

このアプリケーションには、標準的な機械的モデルはない。モデルによって一般的に検討されるのは、簡単な同種のものから構成されるシステム(純粋な水の場合が多い)で、拡散については検討されていない。適用される一般的な手法によって、重要な反応(反応の再結合及び劣化)や適切なG値が限定される。これらの反応は、割合を表わす方程式によって表わされ、形成及び(時間の経過と共に)消費される分種(ガスを含んだ)の量がその等式によって算出される。一般に、このタイプのモデルにおいては、検討されるのは溶液だけで固体については検討されていない。放射線分解を表わす微分方程式を解くために推奨されるコードが、FACSIMILE(非公開)である。この場合、リアクタントや生成物の化学的性質の他にその動きについても検討されている。

#### 4. 4 視覚化情報管理

本項では視覚化ソフトウェアの実例を挙げて検討し、視覚化情報の管理に関する技術について示す。

##### (1) ARC/INFO

ARC/INFOは、本質的に異なるデータを統合し、新しい機能を受け入れることができることから、地理学的な情報システム (ESRI 1995年a) のおそらくリーダー的存在といえる。ARC/INFOによって、視覚化及び図式化されたデータを単一の地質処理システムを用いて統合し、地理学的な背景知識と対比させて、すべての空間的な参考データを考察することが出来る。GISによって、マルチメディアの相互作用が可能になり、それによってグラフィック化されていないタイプのデータ (例えば、ベクトルデータ、地表、ビデオ、走査イメージ及び文書など) を見る事が出来る。ユーザーは、また、CADシステム、アースビジョン、文書管理システムなどの他のアプリケーションを用いて、分析を行うことができる。また、出力されるデータが空間的な性質をもっているのであれば、マップと対照させて結果を考察することも出来る。

ARC/INFOの中核となるのが、別々に限定された一連の層や適用範囲に地理学的情報の概要を取り入れた地理学上のデータモデルである。その一つ一つがそれに密接に関連した地理学的な特性を表わしている。この手法は、主に、関連したデータベースの管理システム (RDBMSs) の概念から取り入れられたもので、その手法はそのような管理システムによって厳密に統合されたものになっている。

いくつかの選択自由なソフトウェアの拡張機能により、ARC/INFOにアプリケーション上の特定の機能が加えられる。例えば、ArcStorm (Arc Storage Manager:アーク記憶処理) は、空間的な情報の記憶及び処理装置である。この特徴指向の連続的な地理学的なデータベースは、ORACLE、INGRES、INFORMIXやSYBASEなどの関連データベースシステムに統合される。ArcCADは、データ構造が根本的に異なるARC/INFOとAutoCADを結び付ける。AutoCADデータモデルの拡張機能として、ArcCADは、GISとCADを統合する際に以前生じていたいくつかの制限条件を取り除くことができる。ArcViewデスクトップ作図及びGISバージョン3.0 (ESRI白書シリーズ1996年11月) は、マップの表示及びそれについての質問、データ分析、編集、作図などに関する強力なツールとしての特徴を有する。



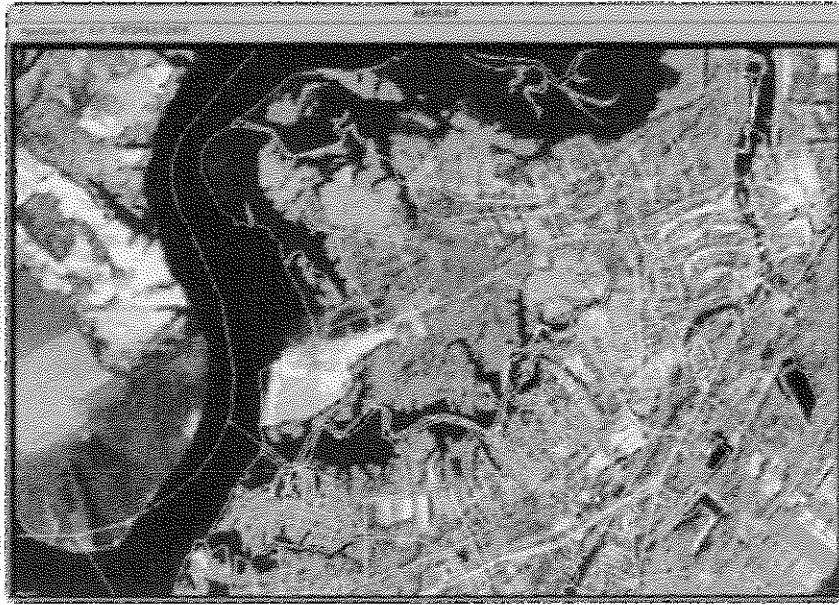


図 4. 4-1 ARC/INFO表示例

プロジェクトの進行に伴い、自動作図、地理学的な情報システム、イメージ処理などの様々な技術を用いることによって、膨大な量の空間に関するデジタル化されたデータが、基礎的な研究やサイト調査プログラムなどから蓄積される。クライアント/サーバー計算やデータベース管理技術の発展により、空間に関するデータベースエンジンを使って、そして地理学的及び地質学的データを管理し、空間分析に関する質問に基づいて選択されたあらゆるアプリケーションに対して情報を普及させていくことがユーザーにとって可能になる。

データ管理に加えて、エンジンによってクライアント/サーバー共同処理アーキテクチャが構築されなければならない。その際には、アプリケーションの効率的な性能は保証されていなければならない。環境システム研究機関の空間に関するデータベースエンジン (ESRI 1995年b) が、論理上の起点である。相互に作用する視覚化技術により、ユーザーと数値計算の間に重要なインターフェイスが形成される。アルゴリズムの内部作用を明白にすることによって、画面上で実験をすることが可能になり、post priori感度分析の手法を適用するよりもより効果的な方法で計算結果を得ることが出来る。

## (2) Vis-5D

Vis-5Dシステム (HIBBARDとSANTEK 1990年) は、このシステムのモデルの履歴ファイルが5D矩形のデータ (2Dの配列と3Dの空間的グリッド) であるという事実からその名称がつけられている。2Dの配列は、時間とモデルフィールド (例えば、温度、圧力、塩分、風など) によって表わされる。これらのデータは、ユーザーが相互制御する形で3Dの動画で表わす事も出来る。システムはデータグリッドを変換して、3Dベクトルと多角形から構成される原線として図式化される。原線を図式化して表わすことによって、「仮想上の地球環境」が作り出され、データグリッドをスライスが移動することによって、あるいは図式化されたイメージを回転させたり、拡大縮小させたりして調

査することができる。

Vis-5Dシステムは、特に、天気予報のモデル化や大気及び海洋の視覚的なシミュレーションのために開発されたものである。それに関連したシステムである、Vis-AD（アルゴリズム展開のための視覚化）システム（HIBBARDなど1992年）は、より一般的なアプリケーション用に開発されたシステムである。Vis-5Dが単にポストプロセッサとして働くのに対して、Vis-ADは多種多様な視覚的実験を支える実行環境としての役割を果たす。例えば、Vis-5Dにおいては、データは5Dの矩形から構成されるとされ、3Dの図式化されたスペースはいつも3Dの物理的スペースを表わしているとされる。一方で、Vis-ADにより、より幅広い分野のアルゴリズムをサポートするために、ユーザーは前述のデータ構造を定義付けし、図式化されたスペースを抽象化することができる。Vis-ADシステムは、データモデル、コンピュータモデル、ディスプレイモデル、及び図式的なユーザーインターフェイスを組み合わせるものである。（HIBBARDなど1994年）

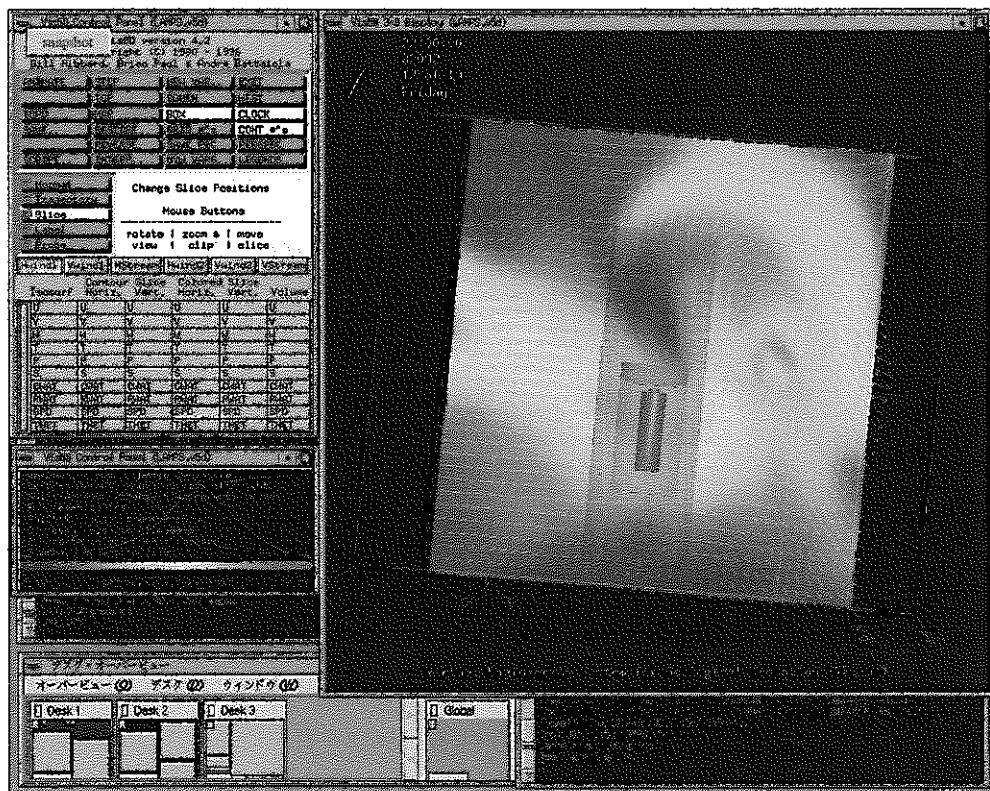


図4. 4-2 Vis-5D表示例

### (3) 影響ダイアグラム

データ視覚化の見地からは、現在適用されている影響ダイアグラムや相互作用マトリックス手法の中には、本来非常に便利なものもあるが、そうではないものもある。例えば、岩盤工学的手法（HUDSON 1992年）の個々のループは多くの情報を伝えることが可能である。しかしながら、ループの定量的及び定性的解釈が簡単ではないので、ユーザーが理解するにはかなり難しいように思われる。「ループゲイン」をある程度まで視覚的に表示することは、価値のあることである。もし、ループゲイン

が相互に作用し合うものであったならば、表示方法はかなり改善されたと考えられる。次に、図4.4-3に示されているように、ダイアグラムの中の経路によって表わされる多くの相互作用には、強力な数学的枠組みがすでに存在している。より多くの明白な情報を得るためにこれらの相互作用を利用することができる。

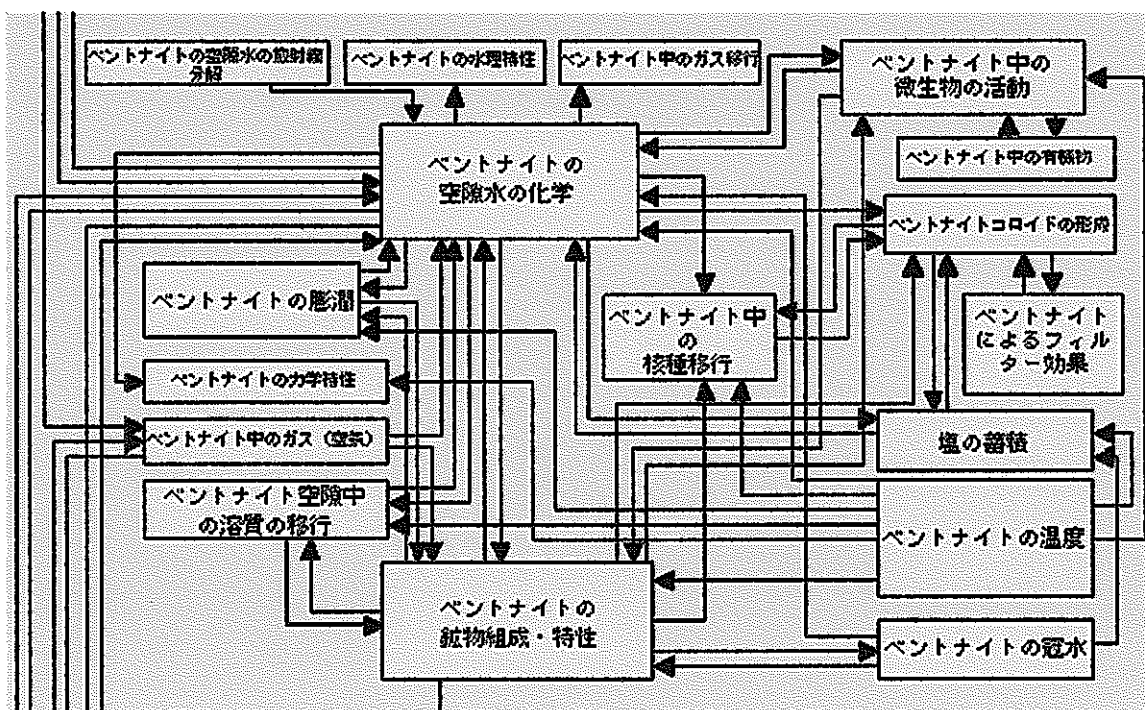


図4.4-3 影響ダイアグラム表示例

#### (4) 影響探索

「インフルエンス・エクスプローラ」(TWEEDIEなど1994年)は、ユーザーが多くの複雑なデータを相互的に調査可能になるように開発されたツールである。データベースにおけるそれぞれの属性は、ヒストグラムの形で表わされる。例えば、図4.4-4は、2つの異なる属性であるX1とX2のデータの分布を表わしている。バーの高さは、ヒストグラムのスケール上の各間隔の範囲内に値が入っているデータベースの項目数を表示している。このようにして、データ項目の値を示すために、ユーザーはバー上のひとつの点を選択することが出来る。そして、データ項目の値は、ディスプレイ上で強調される。

ユーザーは、ヒストグラム軸にスライダーを用いる事によって、選択することもできる。このことによって、選択されたヒストグラムだけでなく他のすべての関連ヒストグラム上のスライダーの限定範囲にあるすべての項目が強調される。例えば、図4.4-5においては、低い値のX1を有するデータ項目はX2の値も低くなる傾向があることがわかる。スライダーをドラッグすることによって、ユーザーは、一つの属性における変化が他の属性にどのように影響を及ぼすのかについて理解することが出来る。図4.4-6は、選択したスライダー(X1)がスケール上を移動した場合の結果を表

している。このシステムに特有の反応フィードバックによって、ユーザーはX1とX2が相互に関連しているということが分かる。同様に、異なる属性の間関係を調べることによって、ユーザーは大量の複雑なデータベースを迅速に調べることが出来るのである。

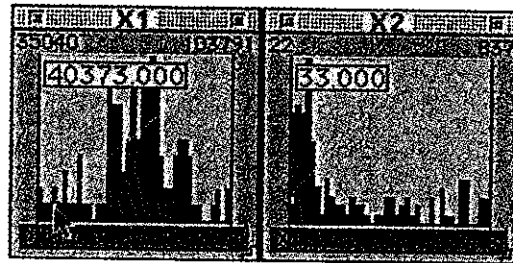


図4. 4-4 影響探索図表示例 (Selecting a single data item)

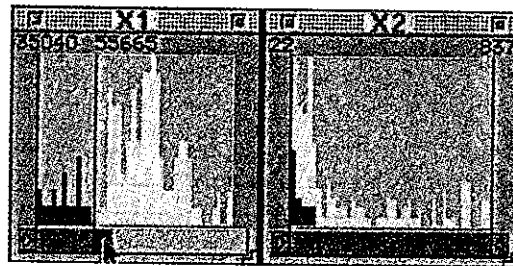


図4. 4-5 影響探索図表示例 (Selecting a range using a slider)



図4. 4-6 影響探索図表示例 (Dragging a slider range)

ユーザーは、2つ以上のスライダを組み合わせて選択したいと思うかもしれない。

例えば、図4. 4-7においては、ユーザーはX1、X2、X3のスライダの範囲を選択している。この場合、感度に関する情報を加えるためには色でコード化する手法が適用される。両方のスライダバーで設定された限定範囲内のデータ項目は黒でコード化される。それに対して、どれか一つのスライダの限定範囲でも外れてしまったデータ項目はダークグレイの色にコード化される。これらの項目は、ユーザーの要求事項をほとんど満たしており、選択された属性のわずかな変化によって

選択されるようなシステムにおける潜在的な変化もこれらの項目により表示される。これらのダークグレイの点により、ユーザーが将来どのようにスライダーを調整したいと思っているのかについての有用な情報を得ることができる。2つ以上のスライダーの限界範囲を満たすことができないデータ項目が、ライトグレイの点によって表わされる。

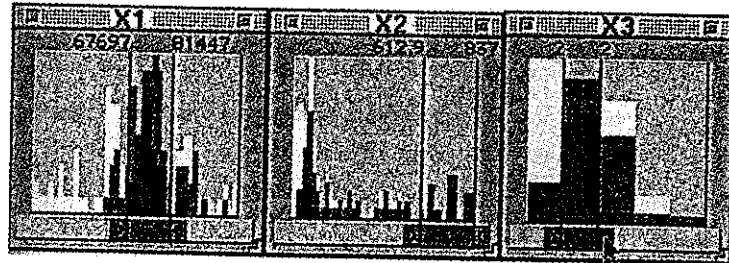


図4. 4-7 影響探索図表示例 (Combining a number of different slider limits)

視覚化データベースに加えて、相互作用ツールもまた、数学的モデルの研究に適用することができる。相互対応ヒストグラムは、それに関連したツールである「インフルエンス・エクスプローラ」(SP ENCE他1995年)に適用されている。この場合、入力パラメータ値のサンプルをとって、出力結果を算出することによって、データが得られる。前もって算出されたこれらのデータは保存され、その結果、相互対応ヒストグラム技術を用いて、データは考察される。このようにして、ユーザーは複雑な問題を理解するために、入力変数と出力変数の関係を質的に調べることが出来る。

出力パラメータの設計を最適化しようとする際、図4. 4-7に示したように感度をコード化することによって、ユーザーはその出力パラメータから仕様を選択することも出来る。生産や構築では設計の変更があるため、ユーザーは、要求事項を満たすような単一の設計を求めているだけではない。その代わりに、例えば廃棄物キャニスタや埋め戻し材などのような信頼性の高い製品を作製出来るような、既知の公差が含まれる設計仕様などを必要としている。このような目的を達成するために、入力パラメータヒストグラム上のスライダーを調整することによって、ユーザーは入力パラメータの公差を設定することが出来ると考えられる。生産高やコストなどの設計全体の目標を実現するために、このような公差を、最適化することが可能である。図4. 4-8では、最小のコストで最大の生産高を達成するための一連の公差がユーザーにより最適化されている。それは、公差範囲の幅によって表わされる。

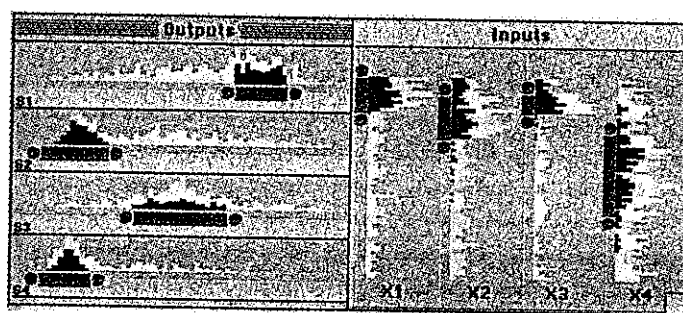


図4. 4-8 影響探索図表示例 (Setting range on input data and output)

特に相互対応の結果が直ちに現れる時（すなわち反動的である時）、ユーザーがダイアグラムと相互対応が可能であるならば、ディスプレイの背後に「隠されている」数学的な関係を洞察する能力は飛躍的に高まる。そうすれば、ユーザーは、「～したらどうなるか」という質問をすることが出来る。また、パラメータ値を自由に変更し、その結果を認識することが出来るのであれば、新しい事実を発見することも可能である。

#### 4. 5 数値計算のためのデータの扱い

処分場の挙動やその環境への影響を予測するためには、システムの物理的、化学的及び生物学的特性を説明する代表的なデータを用いて、機械的な方法でシステム全体をモデル化することが非常に重要であると考えられる。このようにして、予測シミュレーションコードに必要とされるデータベースを選択し、管理することは、コードそのものを選択するのと同じくらい重要である。コードもしくは専門的なシステムに関連しているすべてのデータにあらゆるフォーマットでオンラインでアクセスすることは困難である。すべてのデータを保存・管理するためには、様々なレベルにおけるサポートデータベースが必要となると考えられる。

モデル化調査に基づく数値的な予測は、適用されるデータの品質に大きく左右される。このため、データの標準の必要性が高くなり、特に性能評価計算の場合はデータベースを選択し易くするための一連の基準も必要となる。最も重要なことは、放射性物質の評価研究結果をまとめたデータベースは完全であり、内部的に整合性があり（データが体系的に標準化された）、追跡可能な情報源を参照したもので、最新のデータ処理技術に対応可能なこと、より現実的な予測が行われるように包括的に実験されていなければならないことである。上記のような要求事項をすべて満たすことはできないので、より実用的な手法が（そのデータの適用性やリソースが限られていることなどから）これまで適用されてきた。

##### (1) データベースの管理

膨大な量の数値データを保存・管理するための包括的な方法は非常に重要な要件である。表4. 3-2は、機械的なコードの各カテゴリーのデータに要求される事項、及びそれに関連するデータベースの一覧表である。コードの各カテゴリーに関連するデータベース間にはある程度重複する部分があ

り、その結果、対応するデータベースのデータも重複していることが認識できる。

理論的には、必要とされるすべてのデータは、標準的な参照の情報源から得られるか、もしくはそれらのデータは特定サイトのもので、マトリックス特性や水理学的特性などの処分場の設計特性に左右される。移行分種に特有のデータは、温度や圧力それに塩分、pH、Eh等のような空隙水に関するその他の特性などの主要変数に左右される。このような要因は、処分場の深さや地質的位置選定などに大きく影響される。情報源である共通のデータや依存パラメータを表示するために、ガスの移行に関連したガス、水及び岩石マトリックスデータをまとめる試みがなされている。従って、数値コードによって参照されるすべてのデータを整合性を持たせ確実に保存するために、データベースを統合して、適切なデータベース管理システムを開発する必要がある。データベースシステムによって、特定のタスクに関連したすべてのデータを見つけ出し、引き出すことが可能でなければならない。

最新版のデータが主要情報源である、特定のデータベースの各バージョンを電子的に保存することが提案されている。以前にまとめられたデータベースと、より新しいバージョンのデータベース（未検討である）が比較研究や研究開発のためにそれぞれ保持されると考えられる。例えば、固体やイオンの複合体の水溶液における安定性を説明する熱力学的データベースについては、可溶性や分種化を算出するためにただ一つの熱力学的モデルが用いられると考えられている。これまでは、他の研究者たちによる研究成果と直接比較するために、他のコードを使って熱力学的なデータを変換させる事が必要であった。

従って、地層処分ヴァーチャル・エンジニアリング・システムにおいて、熱力学的データを2つの形態で保持する方法が考えられている。第一の熱力学的データとは、選択されたコードと共に直接使用するのに適しているデータであり、決定論的あるいは確率論的なシミュレーションをする際、データに迅速にアクセスすることが可能になる。第二の熱力学的データとは、第一のデータと同じ熱力学的データを有していると同時に、特定の測定が行われた実験状態の概略などの、重要な情報や文献目録データを有しているフラットファイルのことである。これによって、文献目録的なデータベースの動的リンクが可能になり、情報源となる文書にアクセス可能となる。そして、二次的なデータベースを形成するような数値的なソースファイルもまた作製される。後者（第二の熱力学的コード）の重要性は二重構造になっており、同一の基本的なデータを用いて予測コードを比較することは可能であるが、すべてのコードの使用に適しているフォーマットですべてのデータを保存し管理する必要はないと考えられる。

このように、別個のモデルの成果を比較することは必要とされるべきであり、それによって適切なデータベースを形成することが出来る。データベースを一つのフォーマットから、他のコードとの使用に適した形態に直接変換するために数多くの小規模のコードが書き込まれてきているが、この方式を本プロジェクトに適用することが望ましいと考えられる。

## (2) データ選択の基準

特定の数値シミュレーションのためにデータを選択する際は、ソースデータの信頼性と互換性を評価するために一連の基準を適用することが必要とされる。すべてのデータ型に関連してこのような基準を認定することが本来のプロジェクトである。例えば、熱力学的なデータを決定選択する際に考慮しなければならない重要な基準について以下の項に取り上げている。同様のデータ評価は、地層処分

ヴァーチャル・エンジニアリング・システムにおけるすべてのデータベースに必要とされる。システムに関連している化学的、熱力学的、及び動力学的データに関する現行のデータベースについての調査が、原子力エネルギー機関（NEA）によって1994年に開始された。NEAによる研究調査（NEA 1996年）においては、熱力学的なデータベースは次のように分類されている。

- ・「純粋な物質」（例えば、CODATA;COX他1989年）についてのデータを有しているデータベース。
- ・金属・金属、金属・酸素、酸素・酸素の各システムに関連している特定の「解法データベース」
- ・水システムおよび地球化学システムをシミュレーションするためのデータベース。

NEA（NEA1996年）によってリストされたデータベースの比較や、厳密な評価はなされていなかったが、表向きは同じ用途に開発されたツールに共通性が欠けていることは明白である。このことは、深刻な問題であるので、この研究分野に関してさらに検討する必要がある。

放射性物質を包括的に評価するために設計されたあらゆる組み合わせのデータの品質も変化しやすいことは、避けられない事実である。脱落のリスクよりは損害が少ないと思われているので、測定された定数は理論的もしくは化学的類推に基づいた推測によって補足される。ウラニウム及びアメリシウムの熱力学を包括的に考察したことによって、適用可能なすべての情報が厳密に評価され、特定の情報を選択した理由が明白にされた。（GRENTHE他1992年、SILVA他1995年）適用された実験手順を検討することや、標準的な状態の温度、圧力及びイオン強度に従って生データが転写もしくは再計算された方法を検討することに基づいて評価が進められた。

ある特定の分野のアプリケーションについては、科学的に厳密な選択を行ったり、データを削減したりすることができる。しかしながら、資源やデータに互換性があるにもかかわらず、地化学的な計算のためのデータベース範囲内では、そのように徹底した手法を適用する事は出来ない。CHEMVALプロジェクト（1987年から1994年）において、データベースの検討や拡張（FALCK他1996年）のために、下記事項に関する一連の手順が適用された。

- ・実験的な手法の妥当性／感度
- ・反復分析の精度
- ・データ推測の際に、関連しているすべての平衡状態を含有すること
- ・実験条件の選択及び範囲
- ・算出方法とデータの削減
- ・使用される補助データの信頼性及び互換性

補助データの中には、最初の実験で明示的に研究されていないが、実験データを引き続き評価する際には適用される定数も含まれている。データを選択する際は、想定モデルとそれに関連する適切なデータがデータベースの他の部分と矛盾しないようにしなくてはならない。研究開発において「凍結データセット」を確立させるため形式化した手順が適用され、その結果、CHEMVALの6つの管理バージョンが作成される。（FALCK他1996年）このような手順は、モデル化による計算をその後確認するのに必要不可欠なものである。

水、ガス及び固体の各相に分布されている多種の種を含んでいる自然システムをモデル化するため



には、水種と「純粋な物質」混合物のデータが両方必要とされる。従って、共通参照状態を適用する際には、誤った結果が得られることのないように注意を払わなければならない。異なるデータベースからのデータを混合すると、また、厳密に評価されたデータと推測データ（実験的な論拠となるものが限定されている）が混合してしまうと、地化学的な計算が不正確なものになると考えられる。

FALCKとREADは（1997年）、パブリック・ドメインにおける数多くのデータベースの内容をイオン強度、温度正法、種、相及び熱力学的な定数の観点から比較した。検討されたデータベースの中からCHEMVALの最新バージョンが、放射性物質の評価に適用できる独立型のフレキシブルな選択肢として推奨された。しかしながら、NEA（原子力機関）とCEC（欧州共同体委員会）に於いては、これも新しいデータベースにとって替わると考えられる。

### （3）地球座標システム

地層処分ヴァーチャル・エンジニアリング・システムによって生成されたり処理される大半のデータは、採取場所が表示されたタイプのものである。例えば、遠隔探査、土地の測量、試錘孔ログ、数値的モデル化及び処分場の設計などの多くのソース（情報源）から得られるデータは、適用されるデータベースあるいはソフトウェアパッケージに特有の局所的な座標システムに左右される。この空間的な情報（動的なシミュレーションのための情報や時間に関するデータ）を管理する際には、地層処分ヴァーチャル・エンジニアリング・システムの構成要素間をその情報が移動する間、この情報が確実に維持されるために厳密な品質管理を行うことが必要となる。

このために、地球座標システムを最初から適用する方法が考えられる。地球座標システムを適用するための方法とは、国際的な基準を参照することである。従って、すべての空間に関する情報を次のように表わすことが望ましいと考えられる。緯度と経度を実際の北極と南極、赤道及びグリニッジ子午線に従ってそれぞれ表わし、海拔の基準面は国際的な基準による平均海面によって表示する。

地球座標を適用することにより、システムの構成要素において適用されている局所的な座標に関わりなく、地層処分ヴァーチャル・エンジニアリング・システムにおける空間に関するすべてのデータの位置が明白になる。基準となる地球の枠内の空間情報をシステムが追跡することになるので、多くのコードに共通の自動メッシュ生成ルーチンを適用することが出来る。さらに、地層処分ヴァーチャル・エンジニアリング・システムによる予測出力は、マップや試錘孔ログなどのようなベースとなるデータに直接関連したものになり、広範囲にわたる後処理は必要なくなると考えられる。

## 4. 6 情報の処理

地層処分ヴァーチャル・エンジニアリング・システムの情報データベースの機能要件は広範囲なものである。技術的な専門家をサポートするために最初から科学的な情報を提供する必要があるが、システムの適用範囲を拡大しなければならない。

例えば、教育的活動を援助するような一般的な情報の提供や、合法的な調査研究及び公衆の合意を得るための研究などにおいては、複雑性を有する多様なタイプのデータが必要とされる。従って、情報データベースの研究開発の結果、一つのソフトウェアだけが形成されるということではなく、システムが発展して数多くの別々のモジュールから構成されるシステムとなる。このようにして、新しいデータ管理技術が発展するにつれて、システムがそれほど大きく破壊されることなく、旧式のモジュール

ルとは別のモジュールに取って替わられると考えられる。プロトタイプ（原型）システムの概要については後述の通りであるが、それはこの第一段階において重要と考えられているモジュールを説明するものである。

### （１）プロトタイプ・システム

地層処分ヴァーチャル・エンジニアリング・システム・プロジェクトにおいては数多くの様々なタイプのデータを管理することが必要とされるであろう。例えば、前述したように、熱力学的及び流体力学的なデータのような数値上の情報を必要とするシュミレーション・コードにそういった情報を送らなければならない。このようなデータはすでに発表されている参考文献から得られるであろう。従って、システムには強力な文献目録上のデータベースが必要となるであろう。地図、写真、遠隔探査などのような空間的な情報によって、文献目録上の及び管理上の問題が認識されるようになるであろう。そして、そうした問題に迅速にアクセスしたり、フィルムやオーディオ及びビデオでの記録などのように文献として保存されていないものに迅速にアクセスすることが可能になることによって、システムに対する需要が高まるであろう。

公衆の需要についての研究をサポートするために、業界に関連しているグループの政策や技術的な立場についてよく理解することが必要である。このために、プロジェクトが進行している間、重要とみなされている問題を記録するモジュールを考慮に入れてはどうかと提案されている。このことによって、ユーザーは例えば、ガスの発生によって処分場を保全することが脅かされることなどの特定の問題について、特定のプラントのオペレーターの意見と環境問題に関係する圧力団体の意見を比較することが出来るようになるであろう。

同様に、日本と海外における一つ一つの研究プログラムの進行の様子を監視する事も価値のあることである。一つ一つの研究プログラムの進行の様子を監視した結果の概要をまとめたものからプロジェクト全体の進行の様子を概略することにより、業界の発展の様子を特別に記録することができるであろう。そして、発展を推進させるような技術的な問題を監視することにより発展について先取りして論じることが出来るために、地層処分ヴァーチャル・エンジニアリング・システムの展開が可能になるであろう。

最後に、一般的な情報のデータベースに「索引」サービスを提供できるような一般的なツールが必要不可欠である。このモジュールは、本質的には特に情報を持っている必要はないが、派生モジュールに保持されている情報にユーザーを導いて、その中のデータに迅速にアクセス出来るようにしなくてはならない。

### （２）文献モジュール

文献のソフトウェアがさらに発展して、前述したような文書として保存されていない項目に迅速にアクセス出来るようになることが必要とされている。よく使われる製品を予備的に検討すると、地層処分ヴァーチャル・エンジニアリング・システムに適用可能な、価値がある競合製品が数多くあることがわかる。「EndNote」(Reference)は幅広く使われているマルチプラットフォーム型の製品で、非常に柔軟性に富んでいる。それは、主要オンライン及びCD-ROMのソースから参考データを直接取り入れるものである。それはまた特定の定期刊行物のために前もって規定されているひな型もしくはユ

ユーザーによって指定されているフォーマットで選択された参考データを決まったスタイルで提供するものでもある。「ProCite」(Reference)は、広くユーザーから支持されている、このクラスでは最も長く定着している製品の一つである。「EndNote」で可能な広範囲のプラットフォーム選択はないが、「ProCite」には同様の機能があるので、これを選択することも可能である。

上記のこれらの製品の適用の主な制限として挙げられることは、特定のデータ項目を検索するために非常に大きなデータベースを操作する際、テキスト列全体と比較して反応が遅いということである。

「CAIRS」(食品研究団協会、英国レザーヘッド)は、文献作業上及び関連作業で、英国環境庁が使用しているDOS(唯一の)製品である。その検索手法は迅速で、データエントリーの項目を検索するのにデータベースフラッグのシステムに依存している。それは、メーカーからの技術援助(商業ベースで)を受ける事が可能なので、非常に柔軟性に富んだ製品であると考えられる。その結果、このソフトウェアは(多くの点で多少時代遅れとも言えるが)、英国の放射性廃棄物の管理プログラムにおいて、他のアプリケーション用に開発が進められている。

上記の製品が参考文献や概要の登録を扱うのに適しているのに対して、文献の形では保存されていない資料に迅速にアクセスするには不向きである。これらのタイプのデータには、ユーザーが必要な情報を迅速に見つけ出すことが出来るように他のツールを開発することが必要不可欠になると考えられる。例えば、地図やフィルムを示している従来の文献データベースに概要を登録したとしても、ユーザーが必要とする情報をハイライトすることはない。概要を拡張して情報を1画面以上にすると、それが読み取られないというリスクが生じる。このために、代替的な技術をこれらのタイプのデータに適用する方法が考えられている。MuMMI(McCRINDLEとDOGGETT 1996年)の様なマルチデータを扱えるツールを適用するとこれらの作業を容易に対処することが出来る。例えば、マップの情報をデジタル保存することにより、ユーザーはウィンドウの中のマップを見ることによって必要な領域に導かれる。ハイパーメディアが他の情報につながることによって、ディスプレイの各「層」においてサイト調査報告、掘削孔あるいは遠隔探査に関するデータを得ることが出来るために、ユーザーはデータを参照したり、時には能動的にデータを呼び出すことが可能になる。同様に、オーディオ、ビデオ、新聞の切り抜きなどが文献のデータベースに機能的に結合されることにより、あるいは(そうすることが適切であるならば)それぞれがお互いに結合することによって、それらのデータを保存することも可能である。このような手法によって、基本的なデータを変更することなく数多くのタイプのデータを機能的に結合したり相互に参照させたり出来るので、この手法の潜在的価値は非常に高いと考えられる。

データ管理の主な改革的前進とは、関連データベース管理システムにおける空間、CAD及びイメージなどのデータを保存し、管理するための空間的データベースエンジンSDE(Spatial Database Engine: ESRI 1995年b)である。SDEは単一のデータベースに他のすべてのデータと共に空間データを保存し、(例えば、Oracle, INFORMIX, SYBASEなど)ユーザーによって限定される空間的ではないデータと目的を透過的にリンクする。SDEは、多数の空間的な特性を管理し、局所的及び広範囲のネットワークにおいて他のどの空間的な技術を適用した場合よりも迅速にデータにアクセスし検索することが出来る。

### (3) 論争問題データベース

データベースに原子力業界に関連する社会問題や技術的な問題等を記録することは、非常に有効な手段になり得ると考えられる。例えば、ユーザーは一つの組織における特定の問題についての見解がどのように発展しているかについて調べたり、2、3の組織の見解を比較することも可能になる。公衆の合意に関する研究においては、この方法によって、原子力に関して展開している討議について独特の洞察を行ったり、原子力業界において作成された数多くの文献を通して歴史的なつながりを見いだすことが可能になると考えられる。関連団体による報告書の中の重要な文献の概要を作成し、特定の問題に関連するキーワードを記録することによって、上記の要求機能が満たされることになる。このようにして調査すると、ある組織の政策について言及した特定の問題に関連するすべての文献が明らかになると考えられる。

### (4) データベースの管理及び索引付け

地層処分ヴァーチャル・エンジニアリング・システムのプロジェクトが進行している間、大量の有効・無効なデータが集められてくると予想される。そのような大量のデータからユーザーが必要なデータを検索することが出来るようにデータを注意深く管理し、適切に検討することが必要である。数値データベースの場合には、この作業は比較的簡単なものになる。それは、問題の記録及び特定のデータセットの検討や最終的な受容については、年代順に記録されることが多いからである。

従って、データベース間の相違点を強調して、どのデータベースが保持されているか（文献目録上の参考資料と共に）をユーザーに知らせたり、特別なことについて調べる時には、特定の現象もしくは重要な反応を示す特定のデータ・エントリをユーザーに知らせるために、データベースを管理するためのツールが必要となる。

しかしながら、データベースの中にはより複雑な特質を持つものもある。別々に発展し、再び結合するサブセット（部分集合）が生成されることによって、ファミリー・ツリーに複雑な再帰ループが取り入れられる。同じ事がモデル化コードについても当てはまる。モデル化コードにおいては、一人一人の研究者による様々な研究開発によって統合化がはかられ、その結果派生物の数が減少することにつながると考えられる。文献のモジュールから得られるキーワード（及び発行日）を再度利用することによって、研究開発を年代順に追っていくことができなければならない。コードやデータベースの展開についての記録がそれ自体文献として記録されているならば、特定のコードの展開はデータ管理システムを簡単に調べるだけで決定できると考えられる。あるいは、展開の歴史的な過程がこのように文書化されていることをユーザーがすでに知っていることと仮定すると、文献データベースに直接アクセスすれば、同じ記録を再生することができると考えられる。

データベース管理モジュールは、主要な骨格部分とみなされ、それによってユーザーは他のデータベースを調べることができる。このようにして、キーワードという形式で調べることによって各データベースにアクセスすることが可能となる。この最も高度なレベルにおいて簡素なブール論理機能を働かせることにより、規模の大きいデータベースを参照する際に必要不可欠とされる選択性を有したデータをユーザーは検索することが出来るであろう。

最後に、能率的で効果的なインターフェイスは、最低2つの理由から必要となるであろう。1つ目は、大量で多種多様のデータベースに容易にアクセスできるようにするため、2つ目は、より多くの

研究者、一般の人に、地層処分ヴァーチャル・エンジニアリング・システムを議論のための共通の土俵として提供するためである。インターフェイスの設計は、まず要求される機能によって表現が変わることは言うまでもないが、多くのデータを扱うため、視覚的に分かり易い構成になっている必要があるであろう。データを入力画面では、複雑にそして相互に絡み合うパラメータ群の値範囲の設定や計算範囲の設定等のインターフェイスを通じて行う機能が要求されるであろう。この事は、より範囲の大きな連成問題を扱う際に重要となることが予想される。また、出力画面では研究者や利用者の要求に合わせた結果の表現方法の選択機能が必要である。一般の人には、より分かり易い3次元的表现が要求され、また、正確な値(数字)の読みとりが必要な研究者には、従来のように数字のアウトプットを表示させる等の機能がそれに当たる。

#### (5) シナリオデータベース

シナリオは、主に、天然自然事象のような別々の事象の影響を分析する目的のために検討することが望ましいと考えられる。しかし、説明的な目的、「～したらどうなるか」という調査を行う際のシステム性能を決定するには、全体的なシミュレーション手法も適用されている。ある一定のシナリオを構築する際に行われた重要な仮定や決定は、連続した事象の複雑性に応じて事象/フォールトツリー、影響ダイアグラムもしくは相互作用マトリックスを用いて図式的に表わすことが出来る。より高い視覚的な効果が望まれたり、あるいは一連のシナリオを相互に調査する必要がある場合には、例えば「ポートフォリオ・エクスプローラ」のような適切なIVA(インタラクティブな視覚化手法)アプリケーションを適用することが出来ると考えられる。相互対応を図式的に表示することによって属性値(この場合にはFEPを構成する要素)や各属性の重要性に基づいて、数多くのシナリオから一群の(ポートフォリオ)シナリオを選択、選別して分析することができるようになると思われる。

### 4. 7 オブジェクト指向データベース管理技術

オブジェクト指向データベース管理システム(OODBMS)の分野は、いくつかの個別の研究の収斂の結果として登場してきたものである。オブジェクト指向プログラミング言語、人工知能、ソフトウェア・エンジニアリング、データ抽象化理論および複雑データ管理という分野の全てが、データベース領域でのオブジェクト指向技術の利用に貢献している。OODBMSを必要としてきた応用領域としては、大規模コンピュータ援用設計・エンジニアリング(CAD/CAE)アプリケーション、分散形協力計算、コンピュータ支援ソフトウェア・エンジニアリング(CASE)、オフィス情報システム(OIS)がある。OODBMSの開発は、コンピュータ援用設計を支援する高性能グラフィックス・ワークステーションおよびネットワーク環境が利用可能になった結果、促進されたのである。

OODBMSは、従来のデータベース・システムの全ての機能、つまり、以下の機能を提供するデータベース管理システムである。

#### (1) データモデル

情報をモデル化するために使用されるテンプレートまたは構造のセット、それら进行操作するためのオペレーションのセット、および、データの整合性状態を定義するための制約のセットを提供する。

## (2) 永続性

データは、非持久メモリではなく、永続記憶装置に常駐しており、複数のセッションを横断的に使用することができる。

## (3) 同時性

複数のユーザーが、同じデータベースへのアクセスおよび使用を同時に行ない、相互に対話することもできる。

## (4) トランザクション管理

データの整合性または正確さを保証するためにデータベース相互関係をモニターするプロセス。

## (5) 回復

クラッシュから定義された安定状態への回復。

## (6) 照会言語

情報に体系的にアクセスするための高レベルで、使いやすい言語。

## (7) パフォーマンス

大量の永続性データを二次記憶装置から検索するための効率的なアクセス構造およびアルゴリズム。

## (8) 機密保護

無許可アクセスからの情報の保護。しかし、OODBMSはオブジェクト指向システムでもある。

## (9) データ抽象化

これによって、抽象的で、論理的に複雑かつ柔軟なタイプのデータの開発および利用を行なうことができる。

## (10) 強力な情報モデル化能力

情報が、コンピュータ環境内での実ワールド・エンティティの構造、および挙動を表示かつ補足するクラスおよびオブジェクトの形で、モデル化され、シミュレーションおよび設計オブジェクトに理想的なものとなる。

## (11) オブジェクト見出し

このシステムは、オブジェクトに関する識別名を定義し、維持する。これによって、「等しい」オブジェクト（同じ属性および等しい属性値を有するもの）が共存することが可能になる。ユーザーは、エンティティ事例毎に単一のキーを定義する必要がなくなる。

### (12) カプセル化およびデータ隠し

オブジェクトは、そのタイプに関して定義されたオペレーションによって操作され、その中身は外部アクセスから隠されている（あるいは、特定の言語実現の場合のように、アクセス不能なこともある）。これらのオペレーションの実現は、その使用を無効にせずに変更することもできる。

### (13) 活動（インテリジェント）データ

データとともに手順をカプセル化することによって、その定義域、整合性、有効性、その他を判定する能力が与えられる。これは、インテリジェント・オペレーションのためのデータベースに関するトリガを定義する能力を強化することになる。

### (14) 持越し

これによって、既存のものとの関連における新しいクラス構造の再使用および／または増分再定義が可能になり、導出オブジェクトに対する情報の「継承」ができるようになる。以下の2種類の持越しがある。

特殊化：エンティティ構造および挙動は、クラス導出によって、継続的に特殊化することができる。

一般化：共通の属性および挙動を有する（つまり、類似の）エンティティは、共通のスーパークラスから共通の特性を持ち越すために、一般化することができ、一方、類似性のない属性は、各子クラス内で特殊化することができる。

### (15) 同質多形データおよび機能

データが、実行時間中に決定される様々な形態を動的に仮定するを可能にする特徴である。

### (16) 構成

オブジェクトは、他のオブジェクトの一部とすることもできる。これは、セットおよびアレイのような群オブジェクトを基準とする集合とは異なっている。

### (17) メッセージ・パッシング

相互の方法の呼出しによるオブジェクトの対話シミュレーションを強化する特徴。

### (18) 拡張性

アプリケーションに対して使用可能なオペレーション、構造および制約のセットは、伝統的なDBMSと違って、固定されていない。ユーザーは、アプリケーションに対する増分として、新しいオペレーションやタイプを定義することができる。要するに、DBMS機能と組み合わせられたオブジェクトデータ表示の柔軟性とパワーが、エンジニアリング情報のモデル化、調整、記憶、操作のための強力な手段となっている。

#### 4. 7. 1 データベース・システム要件

大規模コンピュータ支援用設計またはエンジニアリング (CAD/CAE) は、一般的には、相互間の緊密な対話や設計データ交換および情報の動的シェアリングによって複雑な設計を完成するために、分散形ワークステーションによって協力しながら作業を行なう設計技術者のグループに必要とされる。このような設計環境では、強力なデータ・モデル化、シェアリング、管理ツール、ならびに、高度な対話処理等の同時プロセス管理のため、柔軟性のあるトランザクション・ネットワークが必要になる。このようなアプリケーションの支援のために要求されるデータ管理システムの基本的特徴は、以下のようによ約することができる。

##### (1) 複雑な情報のモデル化能力

モデル化および設計しなければならない物理的システムが複雑であるために、エンジニアリング・データ表示は通常は複雑なものになる。設計エンティティも、(関係の構成要素または相補要素のような) 様々な機能的または構造的リンクによって、複雑な相互関係を有している。エンジニアリング・システムの設計には、シミュレーションも含まれ、これによって、設計技術者は、最終設計の前に、プロトタイプの挙動テストを行なう。このためには、エンティティ従属性の表示、メッセージ・パッシング、スケジューリングのような、様々なシミュレーション能力が必要である。オブジェクト指向データ・モデルによって、ユーザーは、あらゆるレベルの複雑なデータ構造を極めて柔軟に定義することができるようになる。オブジェクト間のメッセージ・パッシング (情報交換) によって、オブジェクト指向プログラミング (OOP) は、シミュレーションにとって理想的なものとなる。

##### (2) 意味論的スキーマ設計

エンジニアリング・データの複雑さのために、大きなデータベース・スキーマによって、設計方針および階層を反映しなければならない。そうでなければ、維持および拡張に問題が出て来ることになる。これが特に重要になるのが、複雑なオブジェクト相互関係やオブジェクト同士の依存性をスキーマに表示しなければならない場合である。

##### (3) 動的スキーマ展開

エンジニアリング設計アプリケーションの場合には、アプリケーションの停止なしで、データベース・スキーマを、つまり、属性および方法のクラス定義、持越し構造および仕様を動的に定義および修正するに際して、かなりの柔軟性が要求されることになる。これが必要であるのは、設計が増分プロセスであり、時間とともに展開するからである。

##### (4) 厳格な制約管理

エンジニアリング・データベースの規模および複雑さのために、データが展開するにつれて厳格な設計制約を課すことによって、データ状態の整合性を維持しなければならない。整合性制約は、数種類のデータのタイプ、有効範囲、設計条件、安全限界、などをとることができる。このような制約を課す最善の方法は、データ更新の都度余分な整合性のプログラム・チェックを行なうよりは、制約チェック・メカニズムをデータ構造それ自体に組み込んでおくことである場合が多い。オブジェクトの



形のデータが、このスキーマを実現するのに極めて便利な方法であるのは、オブジェクトの各属性を更新するためのクラス法によって、更新の都度、制約および整合性チェックを行なうこともできるからである。

#### (5) 大容量データの管理

エンジニアリング設計アプリケーションは、高度データ集約的であり、膨大な量のデータの効率的な管理が必要である。エンジニアリング計算の分散的性格のために、データベースを、ネットワーク全体にわたって遠隔接続された分散形超不均一プラットフォームとしなければならない場合がある。設計グループのニーズに応じてデータベースを区分することも可能でなければならない。これについては、簡単に検討を行なう。

#### (6) データに関するデータ管理

CADは高度データ集約的であり、増分的に展開していくために、DBMSの場合には、設計データ自体に関する入念なデータ記録が必要である。この記録は、所有権、作成の時間およびオブジェクト、更新記録、クライアント従属性（つまり、どのクライアントがデータにアクセスしたか）、ロック状況、バージョン、ならびに、トランザクション管理と関連したいくつかのその他のアドバンスト問題のような情報を含んだものである。さらに、設計データに関するこれらのデータを設計オブジェクト自体の中にカプセル化し、それによって、記録の更新および維持を自動化し、しかも、タスクを、外部メカニズムによって制御するのではなく、様々なオブジェクトに分散することも考えられる。

#### (7) データ・シェアリング

エンジニアリングにおける重要問題の1つが、各設計技術者間のデータ・シェアリングである。所有権、使用、作成の時間およびオブジェクト、データ相互関係、ネットワーク全体にわたる分散、または、その他の有意義なオブジェクトのような判定基準に基いて、データを区分およびグルーピングすることが可能でなければならない。クライアントが利用するデータは、対話も可能でなければならない。オブジェクト自体の中に判別基準がカプセル化されるために、オブジェクト指向データ・モデルによって、クライアント間のオブジェクトシェアリングが容易になる。メッセージ・パッシングによって、設計オブジェクトが、実ワールドで表示するエンティティとして対話を行なうことが可能になる。

#### (8) データ・バージョンング

異なるバージョンがデータベース内に共存し、データ更新の結果、古いデータへの重ね書きが起こることがないように、データは、バージョンングも可能でなければならない。オブジェクトが、データの効率的な記憶およびバージョン管理の便利な単位となる。データのバージョンングは、同時性を大きく推進する（というのは、様々な設計技術者が、お互いの作業の終了を待つ必要がなく、同じオブジェクトの異なるバージョンで作業を行なうことができるからである）。オブジェクトがそのバージョン記録を記憶することができるために、設計の展開の軌跡をたどるのにも役立つことになる。いずれの段階であれ、設計に欠陥があると思われた場合には、設計を有効なデータ状態までロールバック

くすることが可能でなければならない。

### (9) クライアント間通信

CADデータベース環境の最も重要な要件の1つが、現在のDBMS環境には明らかに欠けている特徴である、設計技術者間の効果的な通信プロトコルである。設計技術者は、相互の開発状況を知らないことが多く、それが、協調の欠如、比較の減少、相当の時間と資源の浪費、さらに、データの解釈の間違いによる欠陥設計を招くことがある。クライアント間通信は、活発な情報交換、データ修正またはデータ状況の更新、設計インターフェイス横断的な作業の同期化の改善、同時性の向上を促進し、これらはどれも協力開発には不可欠なものである。以下の各種の通信モードを含むことができる。

#### ・ロック通信モード

ユーザーに対してオブジェクトのロック状況を知らせ、これによって、もし要求したオブジェクトがたまたま別のクライアントによってロックされている場合には、ユーザーは、何も知らずに待つ必要はなく、それ以外の有益な作業を予定することができる。

#### ・更新通信モード

該当するユーザーに対して、オブジェクトの変更および更新を知らせる。

#### ・コンフリクト通信モード

2つ以上のトランザクション間のコンフリクト（例えば、デッドロックまたはコミット故障）の場合に、ユーザーに対して通知を行なう。

#### ・ネゴシエーション・モード

相互に合意可能な設計に関するクライアント間の交渉のための、より高度な通信プロトコル。

### (10) 柔軟なトランザクション・フレームワーク

コエンジニアリングとは、結果を共同利用することができるように、作業の各単位で対話を行わなければならない、という概念に基づくものである。これは、同時トランザクションが、そのプロセス中に、結果を相互に伝達可能でなければならないことを意味している。CADのトランザクションは通常は期間が長いために、ロッキング機構は、同時性の向上を可能にするだけの柔軟性を有するものでなければならない。そうであれば、あるトランザクションが終了するまで、処理が相互に待つ必要がなくなるわけである。それに加えて、サブトランザクションを（設計タスクの細目を表示する）別のサブトランザクション内にネストさせ、活動データ・シェアリングのオブジェクトのためにグルーピングする機能が存在しなければならない。全てのこれらのアドバンスト・スキーマは、トランザクションの性格とは無関係にデータベースの整合性を維持する、逐次性という伝統的概念に抵触するものとなっている。しかしながら、同時性を大きく低下させ、CAD環境には明らかに不適當である。従って、トランザクションおよびデータ・タイプの意味論を考慮に入れたその他の手段によって、データの「正確さ」を保証しなければならない。オブジェクト指向データベースは、このスキーマを使用して、より高い同時性をもたらす機会を提供するが、それは、タイプに関するデータベース・オペレーションがその定義域に関して判定を下すからであり、さらに、単なる読み書きよりは意味論的な内容を有している。

#### (1 1) 高速データ・アクセスおよび検索用の効果的な記憶機構

対話形グラフィックス・ベースCADトランザクションは、永続性データに対する高速検索およびアクセスを必要とする。関連データは、様々な判定基準（例えば、機械部分の全てのコンポーネント、または同じデータの全てのバージョン、その他）に応じて、二次記憶装置に隣接集合させることが必要であると考えられる。多くのCADアプリケーションには、グラフィックスおよびマルチメディア情報の利用が含まれ、それには、特別なデータ圧縮、記憶、マッピングおよびアクセス機構が必要である。（オブジェクト見出しおよびオブジェクト間参照に基づく）関連検索が、もう1つの特徴であり、これによって、オブジェクトベースのデータベース内での照会のスピードアップを助けることができる。分散形計算も、異質プラットフォーム横断的なネットワークへのデータの分散を必要とすることがある。

#### (1 2) 計算面で完璧なデータベース・プログラミング言語

大部分のエンジニアリング・アプリケーションは、データの複雑な数学的操作、または、計算面で完璧または十分に低レベルなデータベース定義およびプログラミング言語を必要とするシステム・レベル・プログラミングを含んでいる。例えば、SQLを使用してエンジニアリング設計を行なうのは困難である。

#### (1 3) 互換性、拡張性および統合

エンジニアリング用OODBMSは、より大きな別のCAD環境に統合されるツールとして使用することをオブジェクトとしている。従って、これらのコンポーネントを相互に簡単にインターフェイスさせることが非常に重要である。共通のデータ表示またはデータ操作言語を持つということが、互換性を実現する1つの方法である。OODBMSシステムは、アプリケーション開発者のニーズに合わせるために、カスタマイジングも可能でなければならないと考えられる。

#### (1 4) グラフィック環境

最重要要件ではないが、大部分のCADアプリケーションの場合には、グラフィック環境に対する支援が必要である。これは、データ構造および従属性の操作検索と修正、エンジニアリング図面およびチャートの表示、さらには、アプリケーション開発の全般的な簡素化に有益である。

### 4. 7. 2 RDBMSとOODBMS

伝統的なリレーショナル・データベース・システムの設計は、典型的なビジネス・ニーズに合わせて、多くが決定されてきた。これらのシステムは、テーブル作成が可能で（従って、固定構造を有し）、非常に高度の照会言語によるアクセスや修正が可能なフラットなデータを記憶や操作するには優れた装備を有する。しかしながら、リレーショナル・データ内の構造、オペレーションおよび制約のセットは限定され、固定されている。その結果、アプリケーションが必要とするあらゆる構造やデータを、この限定されたセットの中にマッピングしなければならない。アプリケーションが次第に複雑になるにつれて、このマッピングは面倒で、非現実的なものに成ってしまうのが一般的である。

コンピュータ支援用設計アプリケーションは、高度データ集約的であり、設計しなければならない

複雑なエンティティの構造や挙動を模倣するために、複雑なデータ表示を含むものとなっている。エンティティ間の関係は、複雑または微妙であり、リレーショナル・モデルによってモデル化することはできない。それに加えて、相当な量の推論知能が、実現性、整合性、または、設計プロセスが課すその他の制約の形で、設計データと結び付けられている。これらの制約は煩雑であり、リレーショナル・システムで使用される制約のシステムを使用した場合には、実現が不可能なこともある。要するに、フラットな「ダンプ」関係およびRDBMSの関連制約の形で表示するには、エンジニアリング情報がそれほど簡単ではないことが多いのである。従って、エンジニアリング・データが複雑であることから、アプリケーション専用データ意味論、関連システムが貧弱なツールであるにすぎないデータベース構造の増分開発のメカニズムを補足するために、適正なレベルの拡張性が必要となる。以下に、RDBMSに対するOODBMSに利点を示す。

#### (1) よりな現実的データ・モデル

OODBMSにおける設計クラスやオブジェクトは、実世界の設計エンティティを表示し、RDBMSの場合のようにフラットなテーブルを作成するよりは、問題の技術的側面をはるかに明瞭に実感させる。さらに、リレーショナル・モデルには、挙動（機能）をデータと関連付けるメカニズムがないが、これが、オブジェクトによって容易に実現される。

#### (2) より強力なデータ・モデル

オブジェクト・データ表示は、柔軟性が高く、ほとんど制約なしで、ユーザーがカスタマイズすることができる。データを同質多形で高度化することが可能になる。持越しやスキーマ展開の機能によって、設計を増分発展させることが可能である。データの正規化や不要な重複のような、RDBMSで通常発生するような問題は存在しない。オブジェクトモデルが取扱うことのできるデータの複雑さや関係の複雑さの方が、リレーショナル・モデルの場合よりもはるかに優れている。

#### (3) より容易なスキーマ開発

OODBMSにおいては、一般化や持越しによって、スキーマをより構造の優れた、より直感的なものにし、スキーマがアプリケーションの意味論を補足することが可能である。さらに、共通の属性を除外することができるために（一般化）、持越しによって、スキーマもより小さくなる。設計環境は連続的な変化に特徴があるので、設計技術者が問題に対する理解を深めるにつれて、タイプやスキーマ定義を修正するのに適している。従来のRDBMSシステムは、変更をタイプのレベルで適応させるという限定的な機能だけしか有していない。OODBMSはもっと柔軟性があり、いくつかのシステム（例えば、ORION）は広範なスキーマ展開機能を提供する。

#### (4) RDBMSにおけるインピーダンス不整合

RDBMSを使用して複雑なアプリケーションを開発する際の問題の1つが、データベース操作言語（DML）と、アプリケーションの残りの記述に使用される汎用プログラミング言語との間のインピーダンス不整合である。この不整合には、以下の2つの局面がある。

- ・例えば、SQLのような宣言言語とCのような命令プログラミング言語との間のような、プログラミング・パラダイムの相違。
- ・タイプ・システムの不整合で、これによって、インターフェイスで情報の損失が起こる。大部分の関連システムにおいては、DMLは、エンジニアリング設計に共通の複雑な数学的操作を表現するための計算完全性を欠いている。一方、大部分のOODBMSは、有限要素解析のような大規模プログラムに典型的な複雑さを取扱うことが可能なプログラミング言語（例えば、C++、Clos、Smalltalk）を計算的に完成するための拡張データベースを提供する。

#### （５）オブジェクト見出し

関連データベース・モデルは数値ベースであり、同じ（または等しい）数値を2つ以上の関連オブジェクトに組み込むことによって、2つのオブジェクトの関係を表現する。ユーザーは、類似のエンティティを区別するための単一キーを定義しなければならないことが多く、その結果、データが重複することになる。オブジェクト指向システムは見出しベースであり、2つ以上のオブジェクトをその組み込み数値とは無関係に関係付けることができる（もし必要があれば、OODBMSには、数値ベース・フレームワークも組み込むことができる）。見出しは、いたるところにコピーを保持するのではなく、様々なポイントから同じオブジェクトを参照することによって、シェアリングを容易にする。

#### （６）強力な統一的知識表示

情報のオブジェクト指向表示は、データ（属性）と知識（手順）を統合するための均一な枠組みを提供する。OODBMSには、いくつかの強力な人工知能技術が登場する。実例としては、階層別の分類や特殊化（持越しが、各関係を正確かつ簡潔に補足し、知識の伝播を可能にする）、挙動の委任、方法へのメッセージの同質多形や実行時間結合、自動不要情報収集やメモリ管理戦略、ならびに、エンティティのインテリジェント挙動がある。大部分のエンジニアリング・データベースは、独立のものとしてではなく、例えば、グラフィックス・パッケージやエキスパート・システムを備えた、より大きなCAD環境に統合されるツールとして使用することをオブジェクトとしたものである。データ構造としてオブジェクトを利用することによって、エンティティ・定義域知識がオブジェクト自体の中に組み込まれているために、「知識」と「データ」の区別がファジーになり、別々の表示が不要になる。RDBMSにはこれに匹敵する機能はない。OODBMSの方が、エキスパート、またはインテリジェント・データベースのような、次世代統合ツールの開発に際して決定的に重要な役割を演じるための装備が優れている。

#### （７）より優れたトランザクション管理および同時性

現在のRDBMSは、データベースの整合性を維持するために、同時トランザクションの逐次性を強制する。OODBMSは、伝統的なアプローチによって可能であるよりも高い同時性をもたらす機会を提供する。これは、オブジェクト指向アプローチの場合の方が、データベース・システムが、実行中のオペレーションについてより多くのことを知っているからである。単なる読み込みまたは書き込みではなく、むしろ、より大きな意味論を有している。従って、読み書きコンフリクト・アプローチを使用するのではなく、アプリケーション意味論に基づいて、「正確さ」を強制することができる。

#### (8) より表現度の高い照会言語

大部分のRDBMSは、素晴らしい標準化高レベル照会言語を支援する。現在のOODBMSは、高レベル標準照会言語を欠いている。しかしながら、様々なオブジェクト指向照会形式が現在開発中である。オブジェクトSQL (またはOSQL) は、SQLに対するオブジェクト指向アプローチである。OSQLは、オブジェクト参照によって、照会ステートメントの形での機能呼出し、連想検索を可能にする。これによって、単一照会は (例えば、呼出し機能によって複雑な計算を行なうことによって)、限定されたセットの複雑なオペレーションしか行なうことができない標準SQL照会よりも、はるかに多くのことができるようになる。オブジェクト識別名や照会の機能的構成は、SQL照会の場合には必要なことがある不経済なジョインも不要にすることがある。それに加えて、オブジェクト間参照 (「ポインター・チェーシング」という呼び方に人気がある) によるデータへのアクセスは、関連リンクによるナビゲーションよりもはるかに有効である。

#### (9) より優れた協力作業支援

OODBMSは、協力作業用の環境支援の候補としては、RDBMSよりも優れている。OODBMSは、より複雑なデータ取扱いを可能にし、多数の細分性レベルで人工品のシェアリングの支援を行ない、バージョンや代替案の支援をよりうまく行ない、さらには、多人数プロジェクトに伴う大量の管理情報—例えば、スケジュール、タスク従属性、注釈、記録、設計経歴、設計決定、さらには、プロジェクト固有方針—を補足することができる。RDBMSおよびOODBMSの有効性の比較解析の事例研究が、簡単なコンピュータ統合製造システムについて行なわれている。その結果によれば、オブジェクト指向プロトタイプの方が優れたスキーマであり、情報への便利なアクセスを提供することが可能であり、拡張および維持もより容易である。

## 5 ソフトウェア品質管理研究

本項では、ソフトウェアの品質保証に関する背景となる情報を検討し、地層処分性能評価のための地下水モデル開発における品質保証と品質管理の役割を示す。作業にあたっては、先行していると判断される米国環境保護局の研究報告を参考とした。ここでは、機能的で実用的な品質保証の方法論が紹介され、品質保証にかかわる判断の論拠となる技術・手法を必要とするモデル開発ユーザー等の立場から検討されている。ソフトウェア品質保証の重要な部分は、コード試験と性能評価である。文字どおりに言えば品質保証とは、コード（モデル）または関連している作業（モデル開発）の品質を保証することである。品質管理（モデル開発において）は、モデル基盤の解析やアドバイス（意思決定者に対して）の品質が定量的品質規準や測定を満たすことを保証することである。品質管理に隠された主要な考えは責務であり、主要なメカニズムは全ての作業や結果の記録の維持（ハードコピーと電子ファイル、レポート）であり、より適切な用語は品質文書である。それは、広義で、品質管理は我々が自然に利用できる技術に対する現在の理解の範囲内において最善を尽くすための方法論的で管理上の構造を与える。

品質管理が、コード開発プロジェクトまたはモデル開発研究の受け入れられる品質を常に確証するということは無駄な期待である。しかしながら、品質管理は不完全なコードや不適当なモデル開発に対する予防手段を与えることはできる。モデル開発に基づいたアドバイスが完全に正しいということを保証する方法は何も無く、使用されたシミュレーションコード（または、それについてはどのような科学的モデルでも）が、これらの用語の最も正確な意味においても、証明、検証、妥当にできないということを調整者や意思決定者は理解しなければならない。むしろ、モデルは本物のシステムに関する独立して得られた予測の不一致によってのみ不当にできる。品質管理の主要な役割は、モデル開発者とその利用者との正しい意志疎通を提供することであり、モデル開発者のアドバイスの正確性、不確実性、そして信頼性の意味を与えるということを注目しなければならない。従って、品質管理は初歩のモデル開発者の仕事にのみ適用してはならず、熟練したモデラーにも執着しなければならない様々な注意されるべき点がある。

### 5.1 地下水モデル開発とソフトウェア品質管理

地下水管理や廃棄物処分等の環境評価作業において、地下水モデルの開発は重要な手法である。モデル開発問題において、コンピュータ・シミュレーション・コード（以下、地下水シミュレーション・コードと記す）を開発することは効果的手段として続けられてきた。それは、定性的な一つの科学的手段である。成功したモデルを扱うコンピュータ・シミュレーション・コードは、科学原理、数学的方式、そしてモデル開発過程における、数理解析とプログラミング技術にかかわる専門的洞察力、そして対象問題としてのサイト特性記述の知識が統合化されたものである。従って、それは多方面の専門家、研究者の協調から生まれてくる。

地下水モデル開発は、地下水システムのメカニズムと管理、そしてそれらの品質に影響を与える過程と要因、特に地下水システムにおける人工介入物（汚染物質等、社会地球化学的影響）が原因による特性変化等、地下水を対象とした現象理解を得るための解析手法を組み立てることである。水資源の管理者には、地下水モデルは地下水の水質特性にかかわる管理作業の基本的方針を立案するための有効な判断論拠を与えることができる。これは、特に地下水資源開発、地下水保護、そして帯水層復

元等、現実の地下水管理プロジェクトで利用されている。

地下水管理の成功は、技術的・科学的に確かなデータ収集、情報処理、そして評価方法の使用に基づいて決定が行われ、これらの方法が整然と統合されていることを必要とする。コンピュータ・シミュレーション・コードは基本的なモデル管理の手段であるため、それらのコードが計画や意思決定の道具として用いられる前に、コンピュータ・シミュレーション・コード特性の体系的試験や評価を通して、開発者から独立し、信頼性を確立（検証）することが極めて重大である。地下水管理のためのコンピュータ・シミュレーション・コードは、その開発初期段階においては非公式な開発環境で開発利用されたが、現在ではそれによる判断があまりにも社会的に重要になりすぎてしまった。さらに、サイト特定問題における地下水モデルのアプリケーションは、体系的で包括的な手順や詳細な報告が、結果の持つ社会的異議から求められるようになってきた。そのような背景から、現在地下水モデル開発について論じる際には、モデル開発とコンピュータ・シミュレーション・コードの間の区別がなされなければならない。モデル開発は、以下の3つの構造要素から成り立っていると考えられる。

- ・ 研究対象である地下水システムの定量的理解を得るための研究作業
- ・ ソフトウェア開発と管理作業
- ・ モデル開発と評価手法の開発作業

モデル開発、特にコンピュータ・シミュレーション・コード開発は、地下水資源管理業務において、即時に、かつ長期的予測問題として実行される。コンピュータ・シミュレーション・コードの開発は、サイト、または特定の問題を解決することを目標とした、より大きい作業の一部であり、データ収集、評価と蓄積、システム概念モデルデザイン、代替の問題解決シナリオ、設計、数値解析などの作業を含むものである。地下水モデル開発は、何を必要とするかという多数の意見が存在するにもかかわらず、ソフトウェア工学による「モデル」の定義は曖昧である。水文地質学における「地下水モデル」という用語は、概念的地下水モデル、数学的地下水モデル（解析モデルと数値モデルを含む）と同意語になった。

地下水モデルは、簡易化された数学的説明であり、境界条件、パラメータの形でコンピュータプログラミング言語にコーディングされると考えられる。様々なサイト、または特定問題のシミュレーションに使用可能な汎用コンピュータ・シミュレーション・コードは、（コンピュータ）シミュレーションコード、または総称シミュレーションモデル等と呼ばれる。地下水モデル研究は、ある特定の地下水管理問題を解決するための地下水モデルの開発と使用（すなわち、コンピュータ・シミュレーション・コードとデータ）として定義付けられる。時には、そのような地下水モデルは、一般化された地下水管理問題への一つ、またはそれ以上のコンピュータ・シミュレーション・コードの適用であると考えられる。また時には、モデルは用いられた数学的解析技術の見地から述べられる。最も一般的に使用される用語は、「解析モデル」、「半解析モデル」、そして「数値モデル」である（vander Heijde et al., 1988）。解析モデルは、問題の解（と境界条件）が解析式として与えられる数学モデルである。数値モデルは、一般に、与えられた基礎式から解析式としての結果を導くことが出来ない問題（解析解が導けない式）に関し、時間と空間に離散近似を行い、適当な条件の基に数値計算により解くモデルである。これは、有限差分法、有限要素法等の離散方程式により与えられる。半解析モデルはその両者を混同させた手法である。



効果的で信頼できるソフトウェアを開発し、そのような道具を地下水管理に適用するには多数の段階があり、各段階は念入りに行い、注意深くレビューされなければならない。作業は、モデル開発とコンピュータ・シミュレーション・コード開発過程の全段階において、体系的に管理・統制されたアプローチが必須である。そのためには、同作業に関する適当な品質保証手法の開発と適用が重要である。本項では、地下水モデルと関連コンピュータ・シミュレーション・コードの品質保証に関する背景となる考え方等を検討し、地下水モデル開発における品質管理の役割を定義付けている。

#### (1) 地下水モデル開発における品質保証

地下水モデル開発における品質保証とは、全てのモデル開発に基づいた解析が検証可能で正当と認められ、研究に含まれた全てのプロジェクト作業の適当な実行を技術的・科学的に証明するために、モデル開発研究を管理する協会が手順と操作において構成したものである。地下水モデル開発における品質管理は、モデル開発とモデル使用の両方において極めて重大であり、モデル開発過程の全面に適用されるプロジェクト計画の必須の部分である。品質保証の主要な2つの要素は品質管理(QC)と品質評価である。品質管理とは最終生産品の品質を保証する手順のことである。これらの手順は、コンピュータ・シミュレーション・コードの開発と利用のための適切な方法論の使用、十分な検証や妥当性検査の手順、そして選択された方法やコードの正しい使用を含むものである。品質管理手順を監視し、研究の品質を評価するために品質評価が適用される。各プロジェクトには、品質保証計画(品質管理計画)がなければならない、そのプロジェクトの品質目標を達成するための基準を挙げなければならない。

「品質保証」は様々な分野や環境で使用されている用語である。その意味や実現は分野によって異なってくる。例えば、ソフトウェア工学における品質管理、ソフトウェア品質保証(S品質管理)、そして工業生産品における品質管理の間では大きな違いがある。また、データ品質管理とソフトウェア品質管理手順の間にも大きな違いがある。多くのモデル開発研究は、適切な品質管理の準備が欠如したまま実行されている。正式な品質管理計画、すなわちプロジェクト計画全体のある部分が欠如していることがある。品質管理アセスメントは、プロジェクトが最終段階に入るまでたびたび延期されている。これはサイト特定問題を解くために、あるモデルが適用された研究においては特に重要なことである。しかしながら、大きな支持者層に影響を及ぼす政策決定がモデル開発評価に基づく場合、使用されたモデルと、そのコンピュータ・シミュレーション・コードの両方は、研究結果が受け入れられる前に批判的評論を受けさせられることがある。従って、現在の地下水汚染問題における財政上・刑事上の責任等に影響の及ぶモデル開発作業においては、モデル開発者がモデル開発プロジェクトの全段階に厳密な品質保証手順を履行する事が基本要件とされている。

## 5. 2 モデル開発過程の留意事項

### 5. 2. 1 モデル開発過程

モデル開発は、観察、仮説作り、そして推論された関係の検証を通して、自然界の新しい定量的知識を習得する科学的過程と密接に関連しており、観察された現象の確かな理論的構造の立証という結果になる。従って、地下水システムの基本的理解とは、理論によって統合された研究の産物である。

そのような研究の目的は、現実世界の多要素地下水システムにおける選択された要素を含む原型自

然システムである。研究のための特定原型システムの選択は、管理側の必要性や研究者の個人的興味によって動かされ、現実世界の通り一遍の標本抽出に基づいた統合されたシステムの初期概念化に影響されている（図5. 2-1）。

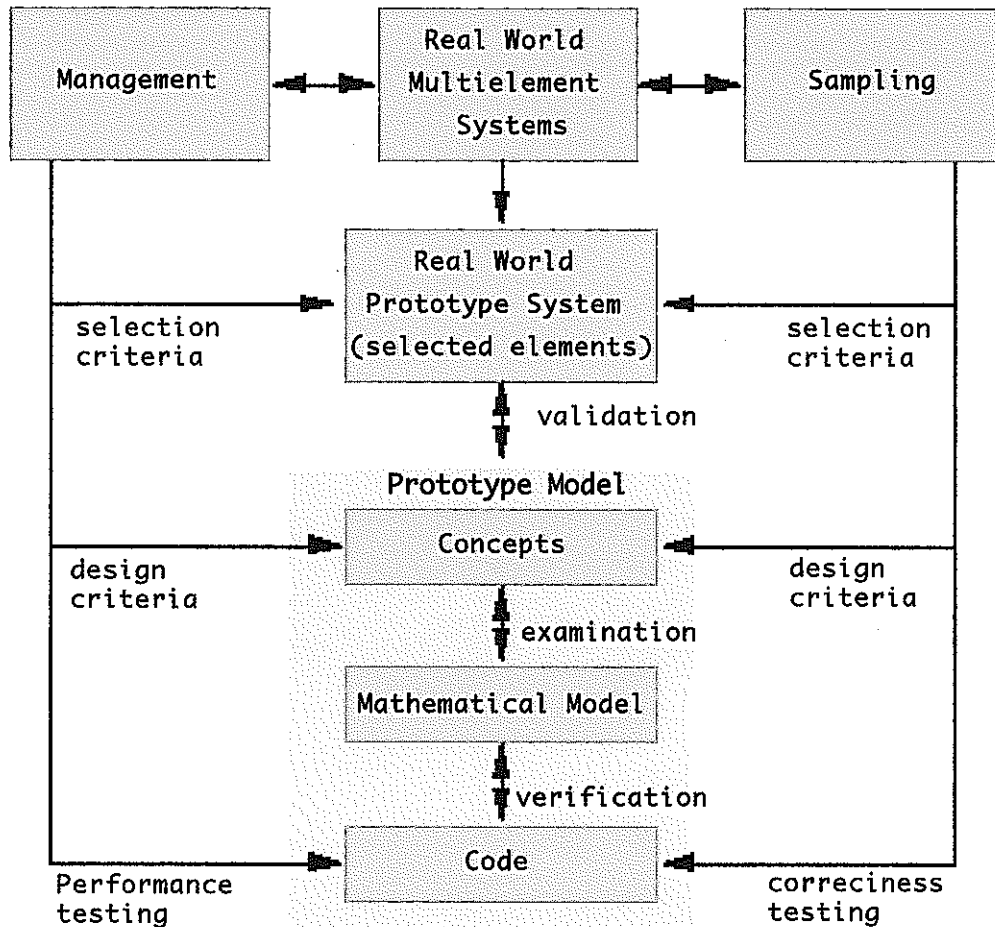


図5. 2-1 モデル開発の概念 (van der Heijde et al., 1988)

選択された地下水システムのプロトタイプモデルは、システムとその環境の様々な構成要素の間の不定の関係を把握するための基礎をつくる。これらの関係は数学的に定義され、数学的モデルになる。数学方程式の解法が複雑であったり、繰り返しの多い計算が必要な場合はコンピュータの使用が必須である。これは数学問題の解のコーディングをプログラミング言語で行うことを要求しコードになる。概念公式化、数学的解説、そしてコーディングがプロトタイプ・モデルを組成する（図5. 2-1）。モデル文書の準備は数学的モデルの公式化の段階で行われ、コード設計やプログラムコーディングの段階で続けられ、他の全てのモデル開発段階が完了した後で終了しなければならない。

地下水モデルが計画や意思決定の道具として使用される前に、モデルの正確さの体系的検査やモデルの性能特性の評価によって、その信用証明が立証されなければならない。利用できる2つの主要なアプローチのうち、評価またはレビューの過程は現実的に定性的であり、コード検査結果は定量的性能の測定によって表わすことができる。例えば、性能特性は信頼性（解アルゴリズムの収束や安定性、

そして端末不足が無いこと等)、コーディングされたアルゴリズムの効率(達成された数値の正確さ対メモリー要求とコード実行時間の見地から)、そしてモデルのステップアップのために必要な資源(例えば、入力準備時間)によって表わすことができる。

コードがシミュレートするために設計された全範囲のパラメータや応力の性能特性が決定される必要がある。地下水資源管理者は、そのようなコードが本来の設計規準を超えて、またはすでに試験された適用範囲を超えて使用された場合の結果を決定するために、更に進んでコードの試験を行うことに興味を持つかもしれない。包括的で体系的なコード検査やモデル評価を通して、コードの適用性への確信が増大する。このように、コード検査(またはコード検証)は、注意深く選択された実例試験問題や試験データセットの実行を通して、プログラミングエラーの発見、組み込まれたアルゴリズムの検査、そしてコードのオペレーション特性を評価することを目標とし、コード検査とモデル検査の違いを区別することが重要である。コード検査は、コンピュータコードに設計された規準や、要求に関連したその正確さを立証することに限定されている(例えば、数学的モデルを表わすため)。モデル検査または(原型)モデル妥当性検査は、そのモデルがシミュレートするために設計された現実世界によって得られた定量的関係の妥当性を把握するためのコード検査よりもより包括的である。

## 5. 2. 2 ソフトウェア開発と保守

ソフトウェア品質保証は、ソフトウェアのライフサイクル全体にわたる手順、技術、そして道具から成る。これはソフトウェア開発プロジェクトの初期段階において、適当な手順(例えば、品質管理計画の開発、記録、プロジェクトの品質管理組織の確立)、技術(例えば、監査、設計検査、コード検査、エラー傾向の解析、機能検査、理論検査、方向検査、レビュー、リハーサル)、そして道具(例えば、テキストエディタ、ソフトウェアデバッガ、ソースコードコンパレータ、言語処理プログラム)が確認されなければならない。ソフトウェア設計規準が決定されなければならない。現在、多くの地下水モデル開発コードがそのような厳密な品質管理アプローチを受けていない。理想的には、研究または資源管理のいずれにおいても、品質管理は現在使用中のコードとこれから開発されるコードの全てに適用されるべきである。

ソフトウェア・ライフサイクル概念の使用は、ソフトウェアの開発、使用、そしてオペレーションの品質管理要求を決定する。このソフトウェア開発ライフサイクルは、主に3つの段階から成り立っている。それらは、開始の段階、開発の段階、そして利用の段階である。この概念によると、開始段階はソフトウェアのための目的や要求を定義し、事業化前調査や費用便益分析を行うためである。開発段階では、ソフトウェアと文書要求が決定され、プログラム設計が公式化され、コーディングは履行されテストされる。利用段階においては、ソフトウェアはそれが設計された目的のために使用される。この段階においては、ソフトウェアは維持され、定期的に評価され、追加の要求の確認に応じて改良・変更される。コード開発と保守における品質管理手順は凡そ以下の通りである。

- ・コードデザインとコード開発のドキュメンテーション
- ・プログラムの構造やコーディングの検証
- ・完全なソフトウェア製品の妥当性検査(モデル妥当性検査)
- ・コード特性、可能性、そして使用についてのドキュメンテーション
- ・科学的で技術的なレビュー

## ・管理上の監査

この手順の一部として次のコード検査手順が考えられる。コードに何らかの修正が行われた場合、例えばある特定のアプリケーションのためにコードを適合するため、そのコードは再度試験されなければならない。また記録、検査と報告を含むコード開発のための全ての品質管理手順が適用されなければならない。修正されたコードは、適用可能なところで、前バージョンで試験された試験データで試験実行されなければならない。その前バージョンの試験結果と比較されなければならない。全ての新しい入出力ファイルは、検査や起こりうる再利用のために保存されなければならない。

### 5. 2. 3 コード設計とコード開発の文書

コード開発や保守の品質管理は、コード設計目的の公式化に始まり、規準や標準、実現戦略に続く体系的なアプローチを意味する。実現戦略は、コード構造の設計やコードに適用されるソフトウェア工学原理の方法の説明を含む。この計画段階においては、コード設計と実現、コーディング手順の記録、各コード区分の目的や構造の記述（機能、サブルーチン）、そしてコード検証過程の記録、の完全な文書を確実にするための測定を行う。コード開発手順の必須の部分はユーザー向き文書の準備である。モデル開発における品質管理の紙トレーラは、モデル開発における報告やファイルから成り、以下の事項を含まなければならない。

- ・（標準化され承認された）プログラマーのノートを含むコードの開発に関する報告。
- ・使用された検証（または妥当性検査）シナリオ、パラメータ値、境界条件と初期条件、そしてそれらのソース、検証結果、そして適応可能であればベンチマーク（データ、コーディング、ソース）を含む検証報告。
- ・使用された検証グリッドの特質とグリッド設計の調整。
- ・ベースライニング後のコードにおける変更との変更の検証。
- ・妥当性検査の場合は、もし適用されたならば、検証手順の詳細。

それに加えて、要求された品質管理のレベルによっては、あらゆるファイルが保持され（ハードコピーと、より高いレベルの場合にはデジタル形式）、以下を含まなければならない。

- ・コードのベースラインバージョンの実行できるイメージとソースコード
- ・各検証実行の入出力
- ・妥当性検査の入出力

### 5. 2. 4 プログラム構造とコードの検証

ソフトウェア工学における検証とは、ソフトウェアの一貫性、完全性、そして正確さを表わす手順である。検証はコンピュータコードにおける数値技術の調査であり、それが本当に概念モデルを表わし、解を得るために先天的問題が無いことを確定するためのものである、と定義している。例えば、地下水モデル開発においては、コード検証には、以下の2つの観点がある

- ・方程式を解くために使用された計算アルゴリズムの正確さや確実さを確かめるため

- ・コンピュータコードが完全に操作可能だということを確実にするため

理論的に正しいコーディング、アルゴリズムや主なプログラミングのエラー（「バグ」）のコードを調査するためには、コードは特別に設計された問題を使用して実行されている。このコード検査へのアプローチは、信頼性の評価（例えば、解アルゴリズムの安定性、端末不足の無いこと）、コーディングされたアルゴリズムの効率と内外部のデータ転送（例えば、数値の正確さ対計算時間によるコード性能、メモリ使用と貯蔵要求）、そしてコード実行準備資源の需要（例えば、入力準備時間）を考慮する。首尾よく構造されたコードは、全体的なプログラム構造に加えて独立したコード区分、サブルーチン、機能やモジュールの効率的な検査を考慮する。この段階のコード検査は、グリッド設計、様々な主要手順、そしてパラメータ値の広い選択に対するコードの感度を評価するためにも使用される。

この時点で、左右する方程式の解析解法に基づいたシミュレーションモデルと数値解法を含むモデルとの間の区別をしておくことが必要である。コーディングされた解析解法の検証は、同じ数学的表現を用いて独立して計算された結果の比較、例えば手動計算、第三者のプログラマーの団体によって独立してコーディングされたコンピュータプログラムの結果からの比較、または、一般的な数値計算用コンピュータソフトウェアシステムの使用に限定されている。数理モデルを表わすコードの検証には、以下の手法が考えられる。

- ・解析解法との比較
- ・同じ総称シミュレーションモデルを表わすコード間のコード相互比較。
- ・統合的データのセットを使用する。

数理モデルのために頻繁に使用される検証アプローチでは、試験問題は空間と時間の領域の各点において応力やパラメータが決定されるシステムを表わし、そのようなシステムの反応は左右する微分方程式に対する閉じた解によって表わされる。試験される数理モデルは、空間や時間における離散した点の限定された数で同じ方程式に解を与える。コーディングが正しいと仮定して、解析解法と数理モデルによって表わされたシステム反応の違いは、主に関連した数値方法の概算的性質とコンピュータの正確さの限界のためであり、一般的にはランダムに散布されていない。

多くの場合、これらの違いの絶対値は、計算方式に使用された離散における解像と関係がある。理論的には、時空の段階の大きさがゼロに接近するほどに解像が増大した場合、数値解法と閉じた解法の差は無くなるはずである。このような種類の検査において、解析解法のコンピュータコード実現が使用された場合、結果コードは数理モデルと同じ厳密な品質管理が必要であるということを注目しなければならない。解析結果と簡易化した状態の結果を比較することによって、数理モデルを試験しても完全にデバッグされたコードは保証されないにもかかわらず、首尾よく選択された問題のセットはコードのメインプログラムとそのほとんどのサブルーチンが試験で使用されているということを確実にし、使用されるプログラムや、あるパラメータフィールドにおいては、コードの計算性能に対して批判的になるかもしれないものを含む。検証練習の効率は、検証問題のデータを使用したコンピュータ・プログラム動作の着実な解析である。

コード検証手順の一部として、簡単な解析モデルには表わされない特別な計算特性を試験するため

に、仮説的なかなり簡易化された問題を使用することができる。明確にすると、このようなタイプの検査は、変則的境界、時間変動応力（ソース、シンク）、帯水層特性の不均質や異方性、そしてグリッドオリエンテーションと幾何学を扱ったコードの性能を評価することを目標とすることができる。このような仮説的問題は、単一フィールドのサイトの特性または様々なサイトの特性の総称的混合に基づくかもしれない。結果として生じるデータセットは統合的試験データセットと呼ばれる。

このような試験に使用される統合的または仮説的システムは、統合的システムパラメータやシステム応力によって定義され、単独的に観察されたシステム反応は無い。従って、検査は数値一貫性や安定性に関するコード反応の評価、または他と得られたものと試験された数理モデルとともに得られた離散的予測の比較によって試験は行われ、それは高い精度の離散方式を使用した立証された数理モデルが好ましい。これらの仮説的問題の絶対的「事実」は不明のため、あるモデルの比較の検証だけが得られる。このアプローチを使用することによって、相対的ベンチマークを得られる。このようなタイプのコード検証は時々「ベンチマーキング」と呼ばれる。

### 5. 2. 5 モデル妥当性検査

モデル妥当性検査とコード妥当性検査や、コード検証を等式化することによって多くの混乱が招かれた。「妥当性検査」という用語は、それが使用されているフィールドによって定義される。ソフトウェア工学においては、全米標準局（the National Bureau of Standards）によって採用されているように、コード妥当性検査は「ユーザーの必要性や要求に関する最終的ソフトウェア商品の正確さの決定」と定義されている。化学製品の環境の結果モデルの評価をするための標準実行について論じるため、米国検査と物質協会（the American Society for Testing and Materials）はモデル妥当性検査を「実験室の実験または、環境の観察から独立して得られた、数値モデルによる計算結果結果と数値データの比較」と定義づけている。

Vander Heijde (1987) は、地下水モデル妥当性検査の目的を、モデルが開発されたプロトタイプ世界の地下水システムの計算されたものと、独立的に観察された原因結果反応の間の「相関の度合い」に関して、モデルの理論的設立とコンピュータ実現が実際のシステム反応をどの程度表わすか決定することと定義している（図 5. 2-1）。この定義は、サイト特定設計モデルは妥当性検査を達成するために開発されてはいるが、サイト特定設計地下水モデルには当てはまらず、総称的シミュレーション・モデルに当てはまると考えられる。

「モデル妥当性検査」という用語は、サイト特定モデル開発にも使用される。さらに、サイト特定モデルがその予測において正確であると監査に証明された場合、そのモデルはそのサイトにおいては「妥当化された」とみなされる。「各サイト特性が固有であるためには、モデルは理想的には各サイト特定問題のアプリケーションごとに妥当化されなければならない」。

相関の度合を量化または定性的に表わすには様々な方法が存在する。モデル入力、システムパラメータとシステム反応の実際に測定されたデータは、本物のシステムと先天的に混合されたエラーのサンプルであるということを注意しなければならない。従って、このように立証されたモデル妥当性検査は常に主観的である。さらにモデルの複雑性は、しばしばフィールド妥当性検査のために使用されたデータがそのフィールドから直接収集されず、それ以前の研究で処理されるということにある。その結果、それらは不正確で、情報に欠け、解説上の先入観、精密さの欠如、そして転送や処理のエラーに陥りやすく、妥当性検査過程において使用されるデータの基本的な劣化の結果に終わる。

妥当性検査過程における検証の役割に関していくつかの議論が行われてきた。物理的パラメータの慎重な調整後にフィールドデータを当てはめるために、検証がコード（と総称モデル）の力量を試験する限りにおいては、著者の何人かはそれを妥当性検査のより弱い形とみなす。この見地に対する弁明は、利用できるシステムパラメータや観察された従属変数の不確実さによって本当の妥当性検査は可能ではないということである。妥当性検査の厳密な形は、検証を除いたモデル予測に基づいている。システム応力やシステムパラメータは、現存する応力へのシステム反応とともに分かっているとみなされる。このアプローチは、パラメータの独立したモデル開発推定を使用して実験データに当てはめるモデル力量の試験である。原則的には、これが妥当性検査への正しいアプローチである。しかしながら、フィールドデータの無効と不正確さは、そのような硬化した妥当性検査アプローチを実際のフィールドシステムに適用することを防止することがある。典型的には、フィールドデータの一部が検証データとして表示され、パラメータ値の適度な調整によって検証されたサイトモデルが得られる。フィールドデータのもうひとつの部分は妥当性検査データとして表示される。

比較のための類似したデータをシミュレートするために、検証されたサイトモデルは予測モードで使用される。従って、このような試験の品質は、その基づかれている検証データを「超えた応力」のサイトモデルの程度によって決定される。ユーザーの見地からすると、総称的地下水モデルの妥当性を決定することは次のような問題への解答と関連する。

- ・ 概念モデルはそれが表わす原型システムに対して妥当か？
- ・ 数学的モデルは概念モデル、関連している過程、そして様々な設計条件における応力を本当に表わしているか？
- ・ コードは本当にモデルの数学的構造を表わしているか？
- ・ モデルは、ユーザーが調査しようとするサイト特定設計の応力シナリオの原型システム反応を本当に表すことができるか？

従って、完全なモデル妥当性検査は、そのモデルが設計された全範囲の条件における検査を必要とする。総称モデルの開発は、新しい研究結果、技術開発、そしてユーザー要求の変化に反応する進化的過程である。モデルレビューと妥当性検査はこのダイナミックな過程に従う必要があり、モデルが修正される度に適用されなければならない。多くの地下水モデルのタイプにおいて、それらの妥当性検査のために要求される広範囲な高品質のデータのセットが欠如している。さらに、利用できるデータセットは、システムのサンプルにおいて起こる様々な条件や応力に関連して限定されることがある。従って、モデルの検査は一般的に現存する解析や体系的ベンチマークを使用した妥当性検査、コードの相互比較、そしてサイト特定設計モデルの詳細なポスト監査に限定されている。例えば、妥当性検査過程における第一段階は、妥当性検査シナリオの定義づけである。モデルの妥当性検査の計画と実行は次の段階をふまなければならない。

- ・ 妥当性検査のデータ必要性を定義し、利用できるデータのセットを選択、または研究するサイトを計画する
- ・ データ品質を正確さ（測定エラー）、精密さ、そして完全さの点から評定する
- ・ モデル性能または受け入れ規準を定義する
- ・ 感度解析の戦略を開発

- ・妥当性検査実行を行ない、モデル性能と立証された受け入れ規準を比較する
- ・妥当性検査実例を詳細な文書にする

様々なタイプの地下水モデルの妥当性検査のため、データベースのさらなる開発が必要である。これらの研究データベースは、広範囲の水文地質学的状況や地下水保護問題を説明し、そのフィールドに現存する様々なタイプの流動、運搬、そして変形メカニズムを反映しなければならない。データベースは、水文地質学、土壌、地球化学、そして気候学の特性に關する広範囲にわたる情報を含まなければならない。そのようなデータベースの開発や標準モデル検査と妥当性検査過程の採用によって、アプリケーションに使用されるモデルはかなり改善される。

また、モデルの絶対的な妥当性は存在しないということに注意しなければならない。ある特定のアプリケーションにとってモデルが妥当であるかどうかは、事前に決定された性能規準を適用することによって評定され、これは妥当性検査または受け入れ規準と呼ばれることがある。計画や意思決定の様々な使用が予測できるならば、様々な性能規準が定義づけできるかもしれない。そしてユーザーはモデルが使用されようとしている目的の妥当性を注意深く確認しなければならない。

#### 5. 2. 6 検証と妥当性検査規準

コード検証と妥当性検査の重要な見解は、評価または性能規準として使用される有益で有効な測定の定義づけである。そのような測定は、モデルに適切な全空間と時間の領域のなかの数理モデルと、モデルが開発されたシステムに直面するかも知れないパラメータの全範囲から得られた解の正確性と安定性を特性づけなければならない。これまで、検証や妥当性検査の受け入れは、主に試験されたコードや使用されたベンチマークに対する予測変数のグラフ説明の視覚検査に基づくものであった (図 5. 2-2)。

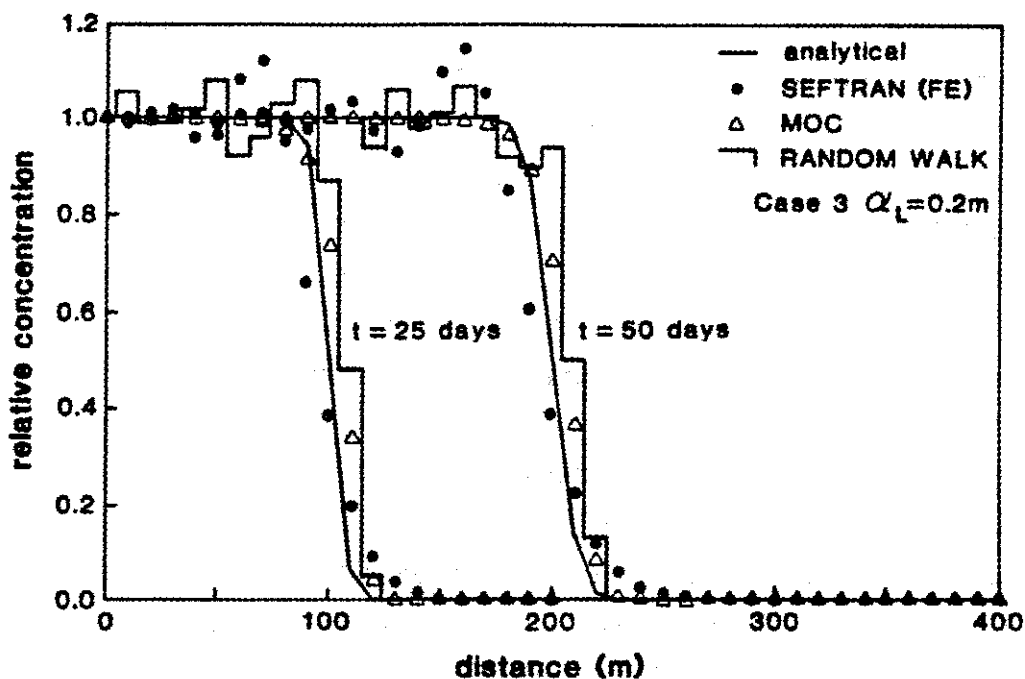


図 5. 2-2 解析解法と様々な数値解法における比較の例 (Beljinより 1988)



ほとんどのグラフの最良の測定は1次元グラフ、例えばヘッド対時間またはヘッド対距離を使用する。これまで、複数次元のグラフは主に妥当性検査、例えば輪郭化された偏差の形で使用されてきた。これら複数次元グラフの技術を検証過程に拡張することによって、適合度を判断する能力がかなり高められるかもしれない(図5. 2-4)。以下は、考えられるコードの性能の定量的測定手順である。

- ・ひと組になったデータ性能 - 時間や空間の正確な位置におけるシミュレートされたデータと観察されたデータの比較
- ・時間と空間の統合された、ひと組のデータ性能 - 時空の平均されたシミュレートされたデータと観察されたデータの比較
- ・振動数領域性能 - シミュレートされたものと観察された振動数配列の比較

彼等は、レビュー中の地下水コード性能の定量的測定を生成するため、2つの統計モデルを選択した。

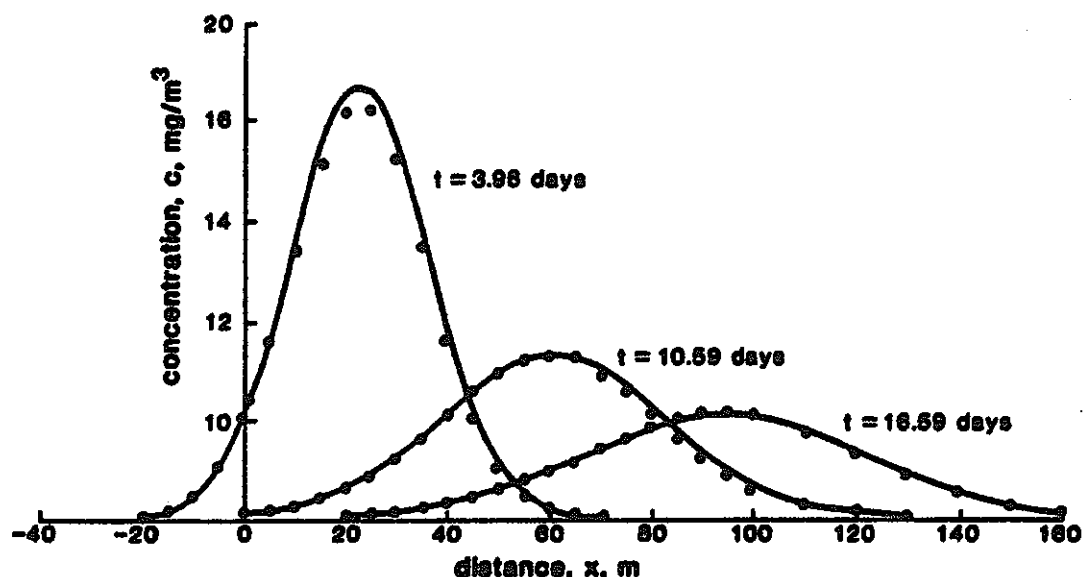


図5. 2-3 空間領域における、解析解法と数値解法の接近した一致の例  
(from Huyakorn et al., 1984a)

これらの3つの検査の各段階は、これらの測定の定義づけに対して別々のアプローチを必要としている。検査の最初の2段階においては、検証の段階と統計測定は不適當であるが、統計に類似した測定は有益と証明されるかもしれない。

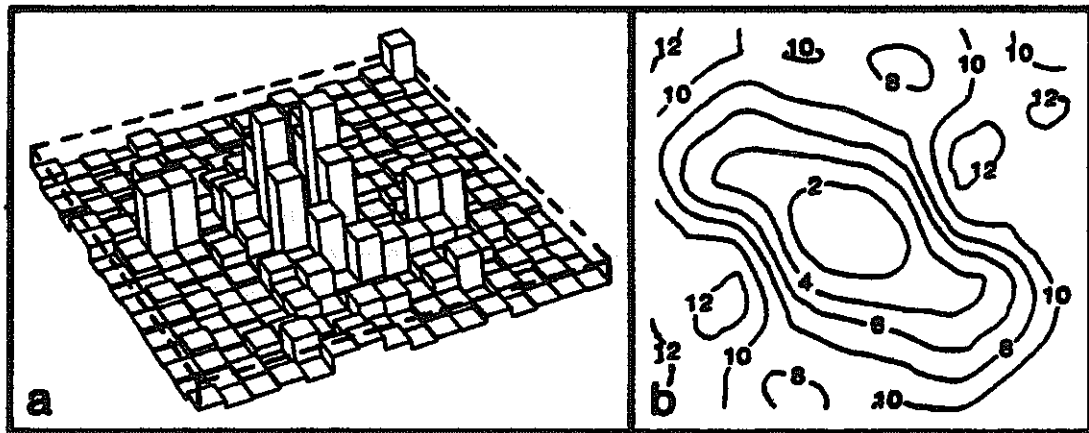


図5. 2-4 2次元検証偏差のグラフ説明。a) ブロック図表、b) 輪郭の偏差。

$$DC = [(x - y) / x] 100\%$$

$$DC = \sum_{x=1}^n [(x_1 - y_1) / x_1] 100\%$$

式5. 2-1

x= 予測された値、y= 測定された値、n= 計算された偏差の合計数。

モデル検査の第3段階であるフィールド妥当性検査は、統計分析される標準データ適合手順に類似している。

### 5. 2. 7 文書

コンピュータコードの文書とは、コンピュータアプリケーションの設計、開発、そして保守の最中に記録された情報であり、目的、方法、理論、関係、能力、そして限界を含むデータ処理システムの適切な面を説明するためのものであると定義づけられる。モデル開発の労作に関係している者、例えばコード開発者、コード保守スタッフ、コンピュータシステムオペレーターやコードユーザーにとって、それはソフトウェアの全ての面において伝達するための主要な道具である。ソフトウェア文書の主な目的は次のように考えることができる。

- ・ システムやプログラム変更をより早く効率的にすることを可能にする技術情報を記録するため
- ・ ソフトウェアオリジネータ以外にもプログラマーやシステムアナリストがプログラムを使用したり研究することを可能にするため
- ・ 何のプログラムで何ができるかということをユーザーが理解できるよう手助けするため
- ・ プログラムの共用ポテンシャルを増加させるため
- ・ プログラムオペレーションの監査や検証を促進するため、すなわち、コード評価
- ・ マネージャーたちに重大な開発についてレビューするための情報を与え、プロジェクトの要件はそろい、資源を使用し続けるべきだということを決定させるため

- ・ 人事異動による分裂の影響を減らすため
- ・ ソフトウェアのオペレーションにおける保守、トレーニング、そして変更についての情報をマネージャー、開発者、プログラマー、オペレーター、そしてユーザーの間に与えることによって理解を促進させるため。
- ・ 他のポテンシャルユーザーにとってソフトウェアが彼等の要求を満たすかどうか決定できるよう、その機能や可能性を知らせるため。

包括的なソフトウェア文書の準備のための詳細なガイドラインが、米国連邦コンピュータ性能評価とシミュレーション・センターによって示されている。この出版物は、マネージャー、ユーザー、アナリスト、そしてプログラマーのモデル情報を提供する4つのタイプのマニュアルのために推奨される構造を論じている。概略マニュアルはモデル説明、モデル開発履歴、実験報告、そして現在と未来のアプリケーションについての議論を含まなければならない。

ユーザーマニュアルは、広範囲なモデル記述、モデル入力データ記述とフォーマット、得られた出力データのタイプ、コード実行準備指示、サンプルモデルの実行、そして障害探索ガイドを含まなければならない。

プログラマーマニュアルは、モデル指定、モデル記述、フローチャート、ルーチン記述、データベース記述、ソースのリスト、そしてエラーメッセージを含まなければならない。コードそのものが上手く構造され、内部的に良く文書化されていなければならない。パラメータ、変数、サブルーチン、そして機能の名前が使用されなければならない。

例えば、地下水モデル開発コードの文書は、コードのモデルによる理論的構造、適用されたコード構造と標準言語、そしてコード実行パラメータに関するコード使用の指示の説明を含まなければならない。良質な文書は、総称モデルが基づかれている方程式、根本的な仮定、モデルに混合された境界条件、方程式を解くために使用された方法やアルゴリズム、そして選択されたアプローチから生じる限界条件の完全な処理法を含んでいる。文書は、コードの実現やオペレーション、そしてデータファイルを準備するためにユーザーの指示も含まなければならない。それは広範囲なコード検証と妥当性検査の報告を含まなければならない。最後に、プログラマー向けの文書はコード修正と保守のための指示を与えなければならない。

### 5. 2. 8 科学技術的レビュー

一般的に、完全な科学的で技術的なレビューの手順はその特質において定性的であり、以下から構成される。

- ・ 選択されたモデル概念、左右する方程式、そしてアルゴリズムの考察
- ・ 文書と基本的な使用の容易さの評価
- ・ プログラム構造とプログラム論理の検査
- ・ エラー傾向の解析
- ・ コンピュータコーディングの検査

検証または妥当性検査が行われた場合、レビューの過程はこれらの過程の評価も含まなければならない。総称モデルの徹底的なレビューを促進するためには、モデルの詳細な文書と開発履歴、同様に

検査のためのソースコードが利用可能であることが必要とされる。それに加えて、検証や妥当性検査結果の再現可能性の独立的評価を確実にするため、レビューする者にとって、コード検証や妥当性検査に使用されたオリジナルの試験データを含むファイルとともに、コードが利用できる、または少なくとも使用のためにアクセス可能でなければならない。

### (1) モデル考察

モデル考察は、原型システムにおける初期の概念化で、何か基本的なものが省略されなかったかどうかを決定する。そのような手順は、研究中のシステムの特徴をモデルの根本的な概念が十分に表わしているかどうかを決定し、モデルの意図された使用方法に適切な過程や動作を確認する。考察は、モデルの適応性の範囲において様々な手順を表わす方程式が妥当であるかどうか、モデルの使用に意図された範囲においてこれらの方程式が数学的に適合されるかどうか、そして選択された解法アプローチが最も適切であるかどうかも決定する。モデル考察は選択された初期境界条件の適切さを決定し、モデルの適用性範囲を確立する。

複雑なモデルには、モデルを表わすために適切な数値方式が採用されているかどうかを決定するために、実現されたアルゴリズムの詳細な考察が要求される。この段階は、数値解法の非独特さ、数値パラメータの不十分な定義、これらのパラメータに使用される誤った、または最適ではない値、数値散布、振動や異なる解法などの数値不安定さ、そして多量の滞留に関する問題などの先天的数値問題を公開しなければならない。それに加えて、モデルの適切なアプリケーションのための特定ルールは、その意図された使用方法の見地から解析されなければならない。そのようなルールの例は以下のようである

- ・有限差分法のノード中心またはブロック中心のグリッド構造によるデータの指定
- ・統合された有限差分法と有限要素法における要素の大きさや形
- ・グリッドサイズの変動
- ・井戸など特異なもの処理法
- ・2次元水平モデルまたは積み重ねた3次元モデルにおける垂直の平均化へのアプローチ
- ・解析要素法における概算解法の算入
- ・境界条件の処理法

数学的方程式、解法手順、及び最終結果が物理的に解釈できる容易さにも考慮する必要がある。

### (2) コード検査

モデルレビュー手順の一部がコンピュータコード検査である。この検査では、コード構造、プログラミング標準のコンプライアンス、プログラミング言語の効果的使用、そして内部文書に関して新式のプログラミング原理が適用された方法に注目される。この段階では、検証実行において検出されるのは難しい検出されていないプログラミングや理論エラーを示すかもしれない。

### (3) モデル文書の評価

モデルの文書は、視覚検査、存在する文書標準とガイドラインとの比較、そして検証や妥当性検査

実行を準備または実行するためのガイドとして使用することによって評価される。

#### (4) 使用の容易さの評価

モデル開発者によって与えられたデータファイルは、試験実行過程を通して、コードのオペレーションとユーザーガイドを評価するために使用されている。この段階においては、コードを操作するために必要なルールと制限やコードの使用の容易化の見解について注目されている。

### 5. 2. 9 ポスト監査

総称シミュレーションモデルの妥当性を決定するために重要な役割は、ポスト監査に残されているかもしれない。ポスト監査は主としてモデルアプリケーションの成功率を決定するために使用されるにもかかわらず、首尾よく実行されたポスト監査解析の明確な結果は、検証と関連して使用されている場合でもモデルの予測能力の受け入れに貢献する。ポスト監査の使用を成功させるためには、概念化、仮定、そしてシステムパラメータや応力を評価し、もし必要であればアップデートし、最近観察されたシステム反応の予測の比較を促進するためにモデルを再実行させなければならない。

## 5. 3 アプリケーション検討事項

### 5. 3. 1 要件

サイト特定問題等に関する総称シミュレーションモデルのアプリケーションは、「モデルアプリケーション」または「(コンピュータ、シミュレーション) コードアプリケーション」と呼ばれることがある。総称モデルのサイト特定設計条件へのアプリケーションは、上手く構成されたモデルアプリケーション原型を従わなければならない。このようなタイプの地下水研究における品質保証は、総称モデル開発プロジェクトで論じられたパターンと同じものに従い、適切なデータの使用、データ解析手順、モデル開発方法論と技術、管理手順と監査から成り立つ。大部分において、モデル開発研究の品質は、モデル開発や品質評価チームの専門的技術によって決定されるということに注目しなければならない。コードアプリケーションにおける品質保証は、モデル開発手順の全ての面を指し以下の論点を含んでいる。

- ・解かれる問題の正確で明確な方程式
- ・プロジェクトの説明と目的
- ・プロジェクトのモデル開発アプローチのタイプ
- ・モデル開発が利用できる最良のアプローチであるか、そしてそうであるならば、選択されたコードが適切で費用効果的であるか
- ・水文地質学的構造、境界条件、応力、そしてコントロールを含むシステムや手順の概念化
- ・明白なもの、無条件なもの両方の仮説や単純化の詳細な説明（批判的な仲間のレビューに従属する）
- ・データ獲得と解釈
- ・コード選択の考慮、または存在するコードの修正または新しいものの開発のための調整
- ・モデル準備（パラメータ選択、データ入力または再フォーマット、グリidding）
- ・モデルアプリケーションに使用されたパラメータ値の妥当性
- ・特に感度の高いパラメータのため、ガイダンスを与えるためのパラメータ概算とモデル検証の

## 原型

- ・コンピュータ出力の情報レベル（表示された変数やパラメータ、フォーマット、レイアウト）
- ・検証目標の確認やそれらがどの程度達成されたかの評価
- ・感度解析の役割
- ・ポストシミュレーション解析（輪郭を示すことに関しては、結果の合理性、結果の解釈、不確実性の解析、そして手動または自動データ処理技術の検証を含む）
- ・適切な性能目標の確立（例えば、6フィートヘッドエラーは、250フィート帯水層の厚さではなく、20フィートヘッドの傾斜または降下と比較されなければならない）。これらの目標はデータの限界を認知しなければならない
- ・結果の発表や文書
- ・管理者側から挙げられた質問に対して、モデル開発の結果がどの程度しっかりと解答しているかについての評価。

モデルアプリケーションにおける品質管理は、モデル開発過程の各段階の完全な記録を含まなければならない。品質管理のための紙トレーラは次の項目を挙げる報告やファイルを含まなければならない。

- ・仮定
- ・パラメータ値とソース
- ・境界と初期条件
- ・グリッドの特質とグリッド設計調整
- ・コードにおける変更と変更の検証
- ・実際に使用された入力
- ・モデル実行の出力と解釈
- ・モデルの妥当性検査（または少なくとも検証）

コード開発品質管理に関しては、監査やプロジェクト後の再利用のためにモデル開発研究で使用された以下を含む全てのデータファイル、ソースコード、そしてコンピュータソフトウェアの実行可能なバージョンが保持されなければならない（ハードコピーや高レベルにおいてはデジタルの形で）。

- ・ソースのバージョンと使用されたコードの実行可能なイメージ
- ・検証の入出力
- ・検証の入出力
- ・アプリケーションの入出力（例えば、研究された各シナリオのもの）

モデル開発研究に使用されたコードに何らかの修正がされた場合、そのコードは標準検査プロトコルによって再度試験されなければならない。コードは、正確な記録や報告を含むコード開発のための完全な品質管理手順を受けなければならない。全ての新入出力ファイルは、現存するファイル、記録、コード、そしてデータセットとともに検査や再利用のために保存されなければならない。

### 5. 3. 2 データ収集

モデル開発プロジェクトにおける重要な段階（時には準備的）は、データ収集である。プロジェクトのこの段階における品質保証は、モデル入力の準備のために収集された技術データの取得、レビュー、照合、変換、そして統合において行われるべき段階や準備的な技術結論を定義づける。

プロジェクトチームによって得られたデータは、利用できる様々な形の一つで、レビューや文書のあらゆるレベルにあるかもしれない（品質管理）。追跡可能性を保証するため、文書のソースや記述の情報を含む受信の記録は、オリジナルデータ書類やファイルとともにファイルされなければならない。追加のデータ処理の間、そのような動作やその後の発見の記録は保存されなければならない。そのようなデータ処理の例は、ユニット変換、輪郭作成、ノード割り当て、その他である。レビューする者は、変換が正当であるということを認められなければならない。記録は、実例計算や変換、そしてデータ処理のためのソフトウェア参照を含まなければならない（例えば、ソフトウェア名、プロバイダー、バージョンである）。

### 5. 3. 3 モデル化

モデル化または設計段階においては、システムや可能な解析手段の様々な面に注目が置かれている。それはシステム概念化（すなわち、水文地質学的構造、適切な過程、そして現存する境界条件の定義づけ）とモデル概念化（すなわち、モデル開発研究の目標と一致するようにシステムを単純化するために、プロジェクトチームは何を正しいとみなすか）を含む。この段階における品質管理は、以下のような全ての問題点における解析の文書を要求する。

- ・地下水と地上水との相互作用、状態の本質的關係や方程式の決定。
- ・時間の尺度の決定（安定状態対過渡状態）。
- ・システムが飽和されているか、または不飽和か。
- ・モデルの次元。
- ・境界条件。
- ・そしてシステムの初期状態。

モデルを使用すると決定された後、研究目的やモデル開発チームの人材と物質資源の限界を熟慮しながら、存在するコードのよく定義され品質保証された特性とモデル開発要求の詳細な記述を突き合わせることによって、適切なコードが選択される。モデル要件とコード特性の間に良い突き合わせが見つからない場合は、突き合わせ規準のいくつかの免除、存在するコードの修正、または新しいコードの開発を考慮してもよい。別のセクションにて詳細なモデル選択を説明し、品質管理コードの選択を強調する。どこか他から得られなければならないコードが選択された場合は、コード転送と実現において特別な扱いが適用される。ここにおいては、品質管理とは文書やコンピュータファイルの完全性の評価、転送メディアフォーマットの文書、そして受信通知や受け入れサインオフのファイリングを意味する。

### 5. 3. 4 モデルの検証と確認

検証とは、結果として生じる仮説が観察されているデータに適度な当てはまりを見せるまでモデル入力を調節する過程である。検証のほとんどは安定状況の条件で実行されるが、過渡なデータのセットに対して2度目の検証を含むこともある。普通は、検証は測定や結果として起こるデータ解析に基づいたモデル入力の最良の概算値から始められる。

初期のシミュレーション結果は、システムの概念を改善するため、またはパラメータ値を修正するために使用できる。モデル検証の成功は、基礎となるモデル公式化の妥当性に従属している。モデル構造が重要なソースを無視した場合、地質学的不均質、物質的過程、または化学反応、パラメータ概算、そしてモデル検証は、有効な入力に適切な公式化を補う（普通は不十分に）ことを強いる適合問題に減らされる。検証の過程においては、より多くのデータが必要になることがある。なかには、コードが初めにデータ収集プログラムを設計するために使用されることもある。新しく収集されたデータは、システムの概念化を改善し、予測的シミュレーションを準備する両方のために使用できる。検証の際に修正を受けやすいモデル入力は、以下を含む。

- ・本質的係数とパラメータ（例えば、水力伝導性、散布係数と分割係数）
- ・強制技術用語（例えば、水または汚染物質のソースやシンク）
- ・境界条件（特定されたヘッド、密集、流動）

検証の際に感度解析は、検証手順の結果にモデル入力のどの要素が最も決定的であるか確認するのを補助するために行われる。どのパラメータにおいても受け入れられる調整の度合いは、一般的に、その値や詳記の不確実性に対して直接比例し、予想された範囲の値または確信間隔に限定されている。普段は、満足できる正確さのレベル内において履歴データを再生した場合、モデルは検証されたことになり、検証問題に先立って決定される。特に履歴的適合が少量のデータに基づくならば、または予測をするために要求されているモデルの可能性を試験しないならば、履歴的データへの適合は良い予測を保証しない、ということを目ざしておかなければならない。検証は、モデル入力データの試行錯誤の調整によって達成されることがある。無関係な要因の多くがモデル出力に影響を与えるため、これはかなり主観的な手順となる。自動パラメータ確認手順がこの主観性のいくつかを削除するのを助けるかもしれない。モデラーの水分学的経験と判断力が、モデルを正確に効率的に検証するための主要な要因になる。検証の重要な品質管理面は検証目標である。検証値とその関連したエラーが検証目標をつくり、これはモデルを検証する前に決定されなければならない。関連したエラーは以下を含む可能性がある。

- ・測定エラー
- ・尺度のエラー（モデル変数の測定の典型）
- ・補間エラー（測定された情報からノード値への転送）

流動モデルのための追加の検証情報は、速度や解決した散布によって得ることができる。そのような検証情報の追加は「独特の」解を得る可能性を高くするかもしれない。通常使用される検証測定は特質上、定性的か定量的である。検証進歩の定性的測定は以下に基づくかもしれない。



- ・エラーの空間的散布の情報を与える、測定とシミュレートされた変数の輪郭図の比較
- ・検証エラーの輪郭化（剰余）
- ・各セルまたは要素における検証エラーのグラフ説明（3次元棒グラフと比較可能）
- ・測定されたもの対シミュレートされた変数の散布図表。直線からの点の偏差はランダムに散布されていなければならない（線形回帰の確信間隔を含むかもしれない）
- ・測定とシミュレートされた変数の製表化、そしてそれらの各ノードの偏差

定量的測定は、検証規準と呼ばれることがあるが、以下を含む。

- ・平均誤差（ME）は測定されたヘッドとシミュレートされたヘッドの平均差である。

$$ME = \frac{1}{n} \sum_{i=1}^n (h_m - h_s)_i$$

式 5. 3-1

nは検証値の数である。MEは検証されたパラメータの様々な値に対してグラフによって表わすことができる。負と正の差の両方がMEに合体しており、互いに削除できる。

- ・平均絶対誤差（MAE）は、測定されたヘッドとシミュレートされたヘッドの違いの平均絶対値である。

$$MAE = \frac{1}{n} \sum_{i=1}^n |(h_m - h_s)_i|$$

式 5. 3-2

- ・標準偏差の平均平方根（RMS）は、測定されたヘッドとシミュレートされたヘッドの二乗の差の平均である。

$$RMS = \left[ \frac{1}{n} \sum_{i=1}^n (h_m - h_s)_i^2 \right]^{0.5}$$

式 5. 3-3

検証成果の評価は、例えば以下の検証レベルによって表わされる（Anderson and Woessner 1992）。

- レベル1：シミュレートされた値は、目標範囲内に入る（検証の最高度合い）
- レベル2：シミュレートされた値は、検証目標と関連したエラーの2倍の範囲内に入る
- レベル3：シミュレートされた値は、検証目標と関連したエラーの3倍の範囲内に入る
- レベルn：シミュレートされた値は、検証目標と関連したエラーのn倍の範囲内に入る

検証されたモデルにおいて使用されたパラメータのセットがフィールド値を正確に表わさないかもしれないため、検証されたパラメータは別のセットの境界条件や水文学的応力のシステムを表わさないかも知れない。モデル検証が検証されたモデルの検証と予測の可能性において、より大きな確信をつくることを補助する。検証データから独立した試験によって、その正確性と予測可能性が受け入れられる範囲内のエラーにあると証明されたならば、サイト特定モデルは「検証」されたことになる。単一の時系列しか利用でない場合は、そのシリーズを2つに分割してもよく、ひとつは検証のため、そしてもう1つは検証（例えば、灌漑開発などの原因によって起こる大きく変化した応力状態のシステム反応の説明）のためである。データが入手できないならば、検証は第2の「独立した」安定状態のデータセットによって行うことができる（すなわち、以前には検証として使用されていない）。検証の最中にパラメータが変更されたならば、この問題は第2の検証となり、第1の検証は変更の説明をするため繰り返さなければならない。第2の問題はもはや「検証」として考慮されない。品質管理の見地から、首尾よく定義されたモデル検証手順は決定的である。レポートは、検証の目標、検証の測定、検証された変数、検証アセスメント（すなわち、検証された変数の適切さの評価と確立された分類による検証測定と検証目標の比較）、そして可能であれば、モデル検証結果の説明を含まなければならない。

### 5. 3. 5 感度解析

モデルの重要な特性は、変動への感度と入力パラメータにおける不確かさである。感度解析は、モデル予測の正確さと精密さを決定するための最も影響を与えるパラメータを確認するために使用される。この情報は、データの質と量の機能としてモデルにおける正確さと精密さを確立させなければならないため、ユーザーにとっては重要である。感度解析は、ある特定パラメータまたはパラメータのセットにおける選択されたモデル性能アセスメント測定（または中間変動）の従属状態を定量的または半定量的に定義づける。感度解析は、シミュレーションモデルの単純化の方法を決定するためにも（例えば、使用された手順、次元、時空の変動、境界条件等）、検証手順の効率性を改善するためにも使用可能である。モデル感度は、入力の変更によって起こる選択された出力の相対的变化として表わされる。入力における変更が出力に大きな変更をもたらす場合、モデルはその入力に敏感ということになる。感度解析方法は、ほとんどが非統計的で、その特質は直観的でさえある。

感度解析は、典型的には毎回1入力パラメータを変更し、従属変数の散布の結果を評価することによって行われる。例えば水力ヘッドである。選択された入力パラメータの公称値、最小値、そして最大値が特定されている。従属変数の散布はその公称値を決定されており、 $C_t^{nom}$ となる。そして与えられたパラメータの最小値と最大値は、その公称値の代理となり従属変数における限界をもたらす。すなわち、パラメータ $p$ の最小値に相当する $C_t^{pmh}$ と、パラメータ $p$ の最大値に相当する $C_t^{pm4}$ である。感度指数 $S_t$ は以下のように定義づけられる。

$$S_t = \frac{C_t - C_t^{nom}}{C_{nom}^{max}}$$

式 5. 3 - 4

これにおいてCtは、Ct<sub>pmh</sub>またはCt<sub>pm4</sub>である。C<sub>max nom</sub>はパラメータの公称値に基づいた従属変数の最大瞬間値である（すなわち、最大時間における従属変数の公称値である）。

汚染物質のマルチメディア的運搬のようにモデルが結合された場合、その後のシミュレーションによる伝播エラーや不正確さの増加が感度解析の一部として解析されなければならない、ということに注目しなければならない。詳細な感度解析は骨の折れる、多くの時間を要する作業ではあるが、それは通常、おそらく直感によって確認されたいくつかの決定的な入力に集中する小さい規模の探究的な解析が実行可能である。感度解析は、検証戦略を設計するために検証を始める初期に実行されなければならない。検証が完成された後で、帯水層パラメータ、応力、そして境界条件の不確かさによってもたらされた検証されたモデルの不確かさをはかるために、より精巧な感度解析が行われる。

上手く文書された感度解析は、シナリオ解析の段階において実行されたシミュレーションの様々な面を評価する最高の機会をレビューする者に与える。従って、感度解析の結果は定性的に説明されなければならない。

### 5. 3. 6 シナリオ解析

ほとんどのモデルアプリケーションの目標は、提案された管理者の意思決定を支持する代理シナリオのシナリオ解析とその後のスクリーニングである。従ってシナリオ解析段階は慎重に実行され、広範囲に文書化されなければならない。そのような文書の重要な要素は、使用されたコードの実行オプション、入力データの完全なセット（タイプとフォーマット）、そしてシミュレーションの直接的な結果の概略（操作されていない、または処理後）を含む。

### 5. 3. 7 結果の解釈

この段階における品質管理は、結果の形成や計算結果とポストシミュレーション解析の意味と合理性についての検査を含む。予測のコントロール計算やポストシミュレーションの妥当性検査（ポスト監査を参照）を行うことが品質管理構造における重要な手段である。確定的モデルによって行われた予測と確率的モデルによって得られた予測の妥当性検査の間には重大な違いがあることを注目しなければならない。不確定解析の数学的モデルの妥当性検査、例えば、性能アセスメントにおいて評価された各シナリオの確率散布計算を確証しなければならない。計算された確率は、シナリオの性能測定への反応を予測し、制御するパラメータが曖昧な値の範囲内にて変化することを認めるならば、結果が出る。曖昧な値の範囲から繰り返しランダムにパラメータ値を選択することによって、再サンプリング研究は行われなければならない。計算は新しい各パラメータのセットと行われなければならない。

### 5. 3. 8 監査

もうひとつの重要な品質管理手順は監査研究に見られる。モデル化されているシステムに関する情報をさらに入手する機会があるたび、モデルにおける工夫や改善を行い、前の解析を修正しなければならない。そのような機会が監査という形によって提供されることがある。監査とは、初期予測の正確さの評価であり、オリジナルのモデル開発研究が完成した数年後に行われる。モデルの予測が正確であったならば、そのモデルはあるサイト特定設計と実際の応力においては「妥当」とみなされる。監査は、予測された変数において新しいフィールド観察を要求し、それは予測された変化にシステム

が適応する機会が与えられた後に収集される。

#### 5. 4 品質管理組織

品質管理は、プロジェクトチーム（品質管理と内部監査）と契約、または監督する組織（品質評価）の両方における責任である。品質管理はプロジェクトの方向を推進または管理してはならず、品質管理は技術データの事後ファイリングを意図していない。ソフトウェア開発またはモデルアプリケーション研究を遂行する組織内の品質管理構造には以下の2つの段階がある。

- ・完全な品質管理方針、目標や目的を持つ常任組織があること
- ・一般的品質管理方針や研究課題がプロジェクト目的に向けて詳細であること

プロジェクト品質管理組織である管理の上のレベルにおいては、品質管理がソフトウェア開発やモデル開発研究の欠くことのできない部分であるということを認識しなければならない。上位の管理者におけるそのような認識は、品質目標を掲げる方針、品質管理機能の確立、そして仕事を実行するために必要な人材、資金、機材などの資源を通して公約に変えなければならない。

品質管理組織は、明らかにされた組織の各要素に特許が無ければならず、責任の大要が必要である。品質管理の責任者は、ソフトウェア開発またはモデルアプリケーションの責任者から独立していなければならない。品質管理組織はプロダクト開発や配信の何においても従属的であってはならない。品質管理プログラムを成功させる鍵は、スタッフの競合にある。品質管理のスタッフは一緒に働くプロジェクトスタッフに対して尊敬がなければならない。彼等は、彼等が品質を保証する仕事を実際にはどのように達成されているかを理解しなければならない。成功した品質管理プログラムの実現は、関係者全員が、品質管理とは何か、なぜそれが行われ、どのような利益があり、彼等に何が期待され、各個人の責任とは何か、そして良質な品質管理はモデル開発手順を妨げるよりもむしろ促進させるということを理解することを要する。彼等は品質管理の便利さと管理によって与えられた重要性を確信しなければならない。

プロジェクト完成の成功と同様に、成功した品質管理は、モデル開発チームの可能性に従属している。良いモデル開発チームは各専門分野協力のもので、かなり訓練された幅広い経験を持つ前任スタッフを含み、効果的な内部のコミュニケーションが見られる。このようなチームは、プロジェクトに関係した様々な学問分野の概略を知り、管理者の質問を技術的プロジェクト目標に直すことができ、管理者側にモデル開発結果を説明できるような人によって管理されている。

#### 5. 5 品質管理計画

モデル開発またはアプリケーションプロジェクトの初めには、完全な品質管理手順のセットを含むプロジェクト検証計画が立てられなければならない。これは品質管理検証計画と呼ばれている。これらの品質管理手順は、規定された品質目標を達成するための規準のリストを含む。品質管理検証計画は技術的作業の着手の前に承認が必要である。プロジェクト検証計画や品質管理検証計画のそのような品質管理部分の主要な要素は以下のとおりである。

- ・妥当性、不確かさ、正確さ、完全性、そして比較可能性に関する要求された品質レベルや品質管

#### 理目標の公式化

- ・適切なソフトウェア開発やモデルアプリケーション研究を行うためのオペレーション手順や標準の開発
- ・品質標準が維持されたということをドキュメントにするために品質管理作業の紙トレールを確立
- ・内外部の監査やレビュー手順のための品質管理上重要点の洗い出し

品質管理計画は、これらの目標達成のための個人的責任や、品質評価段階において万が一問題が確認された場合の補修または訂正の行動手順の概要の特定をしなければならない。

例えば、サンディア国立研究所 (Sandia National Laboratories) における高レベル廃棄物管理プログラムの検証が Harlan and Wilkinson (1988) によって説明されている。この検証計画は、ソフトウェア保守過程の詳細な記述と同時にソフトウェア保管と文書の要件を含む。サンディア国立研究所は、品質管理検証計画を実行するためのコンピュータ化されたソフトウェア管理システムを確立した。このシステムの目標は以下の通りである。

- ・現在、そして最近に使用されたソフトウェアのバージョンの貯蔵所 (追跡可能性と検索可能性のため) にする
- ・報告書、エラーの文書、修正や強化、検証シミュレーション結果等々を含むドキュメンテーションの貯蔵所にする

これに関連して、検索可能性とは、ある一定時間の経過によってハードウェアと、またはシステムソフトウェアに変化があり、そのため実行不可能かもしれないにもかかわらず、適当な時間において開発されたソフトウェアの以前のバージョンにアクセスできることであり、追跡可能性とは計算に使用されたソフトウェアの実際の構成を確認することができることである。

サンディア品質管理検証計画は、米国核規制委員会 (the U.S. Nuclear Regulatory Commission) による「ソフトウェアのための品質保証計画」に従って準備され、NRCのNUREG-0856に依拠することを目的としている。それは、ソフトウェア検査、原稿レビュー、ソフトウェア問題報告、コンピュータコード要約記述、アップデートログ (データとバージョンを含む)、そしてソフトウェアアップデート記述における標準フォームへの準備やファイリングの説明を含む。

典型的なアプリケーション品質管理検証計画は、プロジェクトにおける全ての主要な作業の標準的フォームへの準備、レビュー、ファイリングと、全ての適切なプロジェクトデータ、解析、そしてプロダクト (レポート、電子ファイル、地図、その他) の記述を提供する。それはプロジェクトにおける責任の連鎖と適応された品質管理組織を表わさなければならない。含まれているフォームの例は以下のとおりである。

- ・適応されたモデル開発手順の全要素のチェックリスト
- ・第三者からのデータやソフトウェア転送のフォーム
- ・標準化されたプロジェクトノートの使用 (概念化の仮定、データ評価、コード実現、モデルセットアップ、入力ファイルの準備、ポスト処理、その他に関連した手書きまたはワープロのノートを含む)
- ・データのフォーム、シミュレーションと報告のレビュー
- ・標準化されたソフトウェア問題報告やコンピュータコード要約記述 (データとバージョンを含む)。

全てのフォームは、ファイリング日時と品質管理承認のために余白を含まなければならない。

## 5.6 品質管理評価

品質評価は2つの要素から成り立つ。監査と技術的レビューである。監査とは、プロジェクトに規定された品質管理のレベルと同じ度合いで品質管理要件との承認の程度を評定するために設計された手順である。承認は記録の追跡可能性、有効性（責任者のスタッフからの承認）、そして品質管理検証計画における公約の実現から測定される。技術的レビューは、プロジェクトの技術的・化学的基礎の独立した評価と、プロジェクト結果の有益さから成り立つ。

品質管理評価は、手順が正確に適用されたかどうかをチェックするだけでなく、プロジェクトの当初の目的の全体的成功を定量的に確立することにも関係する。モデル開発とアプリケーションの両方において、品質評価の様々な面が存在する。まず最初に、ソフトウェア開発またはレビュー、そして責任のある研究者によるコントロール計算の実行におけるレビューと検査であり、時にはプロジェクトに参加していない他のスタッフ、または組織外の招かれた専門家が含まれる。プロジェクトが着手された組織による品質評価も考慮されなければならない。再び、プロジェクト、プロジェクトレビュー、または検査において例えば以下の3段階に分類される。

- 1) プロジェクト管理者またはプロジェクトモニターによるレビューまたは検査
- 2) 管理または資金を出している組織内のプロジェクトには関係していない技術専門家によるレビューまたは検査。
- 3) 外部の同僚のパネルによるレビューまたは検査

品質評価は一般的には技術と管理のレビューの形式を取る。論評はメモやレポートの形、またはレビューされたドキュメントの許可済みのコピーの余白に注釈として表わすことができる。万が一このレビューが追加の研究や訂正の作業のための重要な推薦に至った場合、プロジェクトチームによる追従と追加のレビューが取り計られ文書化にされなければならない。

### (1) モデル開発アプリケーション目標

- 01 管理側の決定目標とモデル開発目標は明確に特定されなければならない。
- 02 管理側の決定目標は、サイトの物質的特性（例えば、水文地質学的システム）や汚染のソースと場所についての存在する情報に基づかれなければならない。
- 03 モデルの機能（例えば、データ構造、システムの理解、代替りの補修の追加のフィールド特性記述の計画または評価）はモデル開発目標の開発中に明らかにされなければならない。
- 04 モデル開発着手の前に、評価されるべきポテンシャル解法（例えば、包含と補修解法）は確認されなければならない。
- 05 要求される解析のレベル（例えば、数値モデル、解析モデルまたは図解技術）は、モデル開発目標を明らかにする際に決定されなければならない。
- 06 管理者は、専門の地下水科学者との相談をもとに、モデル予測に必要な時間（例えば、1年、10年、または100年）を特定しなければならない。
- 07 モデル開発結果に要求される確信のレベル（定量的または定性的）が明確にされなければならない。

- 08 モデルアプリケーションの性能目標（例えば、許容ヘッドエラー）が明確にされなければならない。
- 09 これらの研究目標の発展と関連する増分コスト（例えば、研究領域の大きさ、モデル化された補修技術の数、またはモデルの性能目標の拡張）と、管理決定目標を支持する結果として起こる増分改善についての解析が行われなければならない。
- 10 管理側の決定目標は、モデル開発過程の全般にわたって再確認されなければならない。
- 11 モデル開発着手の前と概念モデルの開発後に、それらが管理側の決定目標を支持していることを確認するために、モデル開発目標はレビューされなければならない。
- 12 要求される解析レベルは、プロジェクトの経過の間にレビューされ、そして必要であれば修正されなければならない。
- 13 モデルの機能は、プロジェクトの経過の間にレビューされ、そして必要であれば修正されなければならない。

## (2) プロジェクト管理

- 14 実際にモデル開発を実行し、モデル開発労作の管理や同僚のレビューを行っている人達は、プロジェクトに要求される地下水モデル開発経験を持たなければならない。明確には、プロジェクトにおける彼等の役割のため、各自が以下のような適当なレベルを持たなければならない。
  - ・モデル開発と水文地質学における正式な訓練
  - ・物質的システムをモデル開発する職務経歴
  - ・サイト水文地質学の特性を表わすフィールド経験
  - ・モデル開発プロジェクトの管理経験
- 15 これらの各個人は、よく明確にされた役割、責任、そして参加レベルを持つ結合力のあるモデル開発チームとして編制されなければならない。
- 16 チーム編制はアプリケーションに適切でなければならない。
- 17 プロジェクトの最初に、独立した品質保証（品質管理）過程が確立されなければならない。
- 18 この品質管理手順は、以下に対する進行する同僚のレビューを含まなければならない。
  - ・モデル開発目標開発
  - ・概念モデル開発
  - ・モデルコード選択
  - ・モデルセットアップと検証
  - ・シナリオのシミュレーション
  - ・ポストシミュレーション解析
- 19 この品質管理過程は実現されなければならない。
- 20 モデルアプリケーションをドキュメントにするため、手順が確立されなければならない。
- 21 文書は以下に関する説明を含まなければならない。
  - ・一般的なサイトのセッティング
  - ・関心のある物質的システム
  - ・評価すべきであるポテンシャル解法
  - ・モデル予測のためのモデル開発目標と時間の骨組み
  - ・品質保証と同僚のレビュー過程
  - ・モデル開発チームの構造
  - ・データソースとデータ品質
  - ・概念モデル。水層序学、地下水流動システム、水文学的境界、水力学的特性、流動体ソースとシンク、汚染物質のソースの船荷と地域の広がり、そして汚染物質の輸送と変形過程
  - ・コンピュータコードの選択。コード、信頼性、使用性、輸送能力、性能、公衆領域対所有モデル、限界、そして関連したアプリケーションの記述
  - ・地下水モデル建設。コード修正、モデルグリッド、水力パラメータ、境界条件、そして単純化した仮定

- ・検証、感度解析、そして検証
  - ・予測シミュレーション。シナリオ、シナリオの実現、そして各実行結果の説明
  - ・不確実性の解析
  - ・決定目標に公式化されたような管理側の情報要求の結果に関する説明
  - ・実行上の要約（決定目標による）
  - ・参照
  - ・入出力ファイル
- 22 文書は、独立したレビューする者がポストアプリケーションアセスメントを完成するために必要な情報を提供しなければならない。

### (3) 概念モデル開発

- 23 どのようなコンピュータモデル開発の前にも、局地と広域の水文地質学的システムの初期概念モデルが開発されなければならない。
- 24 概念モデルは以下の情報を含む、他の定性的データと同様、フィールドデータの定量化に基づかなければならない。
- ・帯水層システム（帯水層と半透水層ユニットの散布と相関位置）。関連したユニットの厚さと連続性、地域の広がり、そしてユニット間の相互連結
  - ・水文学的境界。帯水層システムの物質的広がり、地下水システムをコントロールまたは衝撃を与える水文学的特性、地下水分割、そして地上水体
  - ・水力学特性（何に関連し、均質で等方性を含む）。伝達可能性、間隙率、水力学的伝導性、蓄積可能性、そして特定産出
  - ・ソースとシンク。帯水層へのかん養（例えば浸透）、蒸発散、排水溝、地下水流量（例えば、地上水体への流動）、そして井戸（例えば、給水量、注入または灌漑井戸、水平井戸）
  - ・流体ポテンシャル（例えば、電位差計の表面、各モデル層内の水力学的傾斜の振幅と方向）
  - ・汚染物質。ソース、流送、地域の広がり、物定性的特性、化学相互作用、そして生物変態
  - ・土壌
- 25 フィールドデータの量、質、そして完全性は概念モデルの開発の一部として解析されるべきである。
- 26 データにギャップがある場合（例えば、水位または水力学的伝導性の情報の欠如）、これらのギャップを埋めるための追加のフィールドワークや他の試みをドキュメントにする必要がある。
- 27 データにギャップがある場合、追加のデータを取得する際のコストと結果として起こる管理側の決定目標を満たす改善の間から解析されなければならない。
- 28 データソースはドキュメントにされなければならない。
- 29 データの品質は調査されドキュメントにされなければならない、プロジェクト結果における品質の影響は評定されなければならない。
- 30 他の物質的システムとの全ての可能性のある相互作用（例えば、地上水システムまたは農耕システム）はモデル開発の初めから、水予算、大量化学の見積り、または他の解析学的技術という手段で評価されなければならない。
- 31 存在する、または未来の工学（例えば、井戸、スラリー壁）が数値または解析モデルに表わされなければならないということは、概念モデルに明確に含まなければならない。
- 32 水柱中の汚染物質のかたまりを説明するため、十分な汚染物質ソースが確認されなければならない。
- 33 境界条件の場所、タイプ、そして状態の明確な報告書。それらの公式化の調整、そして境界条件の情報のソースが概念モデルの一部として含まれなければならない。
- 34 数理モデルの検証の始まる前に、全ての概念モデルパラメータと適度なパラメータ範囲が特定されなければならない。



#### (4) モデルコード選択

- 35 選択されたモデル（コード）は、その流動と輸送過程、数学、水文地質学的システム表示、境界条件と入力パラメータに関連して説明されなければならない。
- 36 モデル（コード）の信頼性は下記のレビューを含めて評定されなければならない。
- ・同僚によるモデル理論のレビュー（例えば、地下水の専門的知識または同僚にレビューされたジャーナルでの理論の出版によって一般に認められた個人または組織による正式なレビュー過程）
  - ・同僚によるモデルコードのレビュー（例えば、地下水コンピュータモデルを評定する専門的知識のため一般に認められている個人または組織による正式なレビュー過程を受けなければならない）
  - ・検証研究（例えば実験室試験、解析解法またはよく受け入れられたモデルに反対したモデル結果の評価）
  - ・関連したフィールド試験（すなわち、広範囲なデータのセットが利用できるサイト特定設計条件のモデルのアプリケーションと評価）
  - ・使用の量とタイプによって立証されたようなユーザー社会におけるモデル（コード）の受容性
- 37 モデル（コード）の使用性は以下の利用可能性を含めて評価されなければならない
- ・モデルの2進コード
  - ・モデルのソースコード
  - ・前後のプロセッサ
  - ・存在するデータ資源
  - ・標準化されたデータフォーマット
  - ・完全なユーザー指示マニュアル
  - ・サンプル問題
  - ・必要なハードウェア
  - ・プラットフォームを横断する輸送能力
  - ・ユーザーサポート
- 38 トレードオフは、モデル（コード）性能（例えば、正確性と処理速度）とモデル開発を実行するために要求されている人材とコンピュータ資源の間で解析されなければならない。
- 39 モデル（コード）は、公衆領域にあるか、少なくともすべての関心を持つ全ての団体に即座にアクセス可能でなければならない。もしそうでなければ、モデラーは、アクセス不可が研究目的や調整機能過程から品質を落とさないことを説明しなければならない。
- 40 モデル（コード）における仮定は、モデル開発目的の衝撃に関して解析されなければならない。
- 41 モデル開発要件の間の全ての不一致（すなわち、決定目標、概念モデル、そして利用可能データによって示される）と選択されたモデルの可能性が確認され調整されなければならない。モデラーは、何故モデル開発目標と、または概念モデルが修正される必要があったのか無かったのか、説明しなければならない。例えば、1、2、3次元モデル開発を支持する選択されたコードの実現。安定対不安定状態のモデル開発の提供。または概念モデルの単純化の要求が説明されなければならない。
- 42 そのような不一致によってモデル開発目標が修正された場合、それらの修正はドキュメントにされなければならない。
- 43 モデルソースコードが修正された場合は、試験を実行し、検査方法論と結果を調整しなければならない。
- ・信頼性の検査（#36参照）
  - ・使用性の評価（#37参照）
  - ・性能検査。

#### (5) 入力データ

- 44 全体的なグリッド解像（例えば、100フィート対1000フィートの間隔の平均）は管理側の目標

を満たすために要求されている従属変数の正確性に関連して解析されなければならない。例えば、グリッドは、水力学的傾斜が正確に表わされることを可能にするほど良質でなければならない。

- 45 有限要素または有限差分グリッド設計は、正確に井戸の場所を見つげたりモデル開発したりする必要性、存在するものと未来の工学、または汚染物質のソースと水柱などのモデル開発目標と関連して解析されなければならない。
- 46 グリッドは物定性的システムに関連して設計されなければならない。例えば以下の通りである。
  - ・主要グリッドオリエンテーションは、水力学的伝導性と、または伝達可能性の主要な方向と一直線に合わせられなければならない
  - ・結果が必要な場所（例えば、最も汚染されている場所または水位降下）または、伝達可能性の変化、水力学的ヘッドの変化、または密集した傾斜のある場所では、より良質なグリッドが使用されなければならない
  - ・データが乏しいところや、研究地域の特に関心のない部分においては、より粗いグリッドが使用されなければならない。
- 47 グリッドの間隔とタイムステップの大きさは数の正確さに従って解析されなければならない。例えば以下の通りである。
  - ・変数グリッドの間隔がおかれた有限差分モデルが利用された場合、そのグリッドは離れた境界に向かって2の因数に拡大されなければならない
  - ・有限要素モデルが使用された場合、モデルアプリケーションの数値の正確さの衝撃に関して次のものが解析されなければならない。各要素の長さとの幅の割合、隣接する要素の多きさの違い、そしてペクレ数 ( $Pe = v \cdot \Delta X / D$ )
  - ・クーラン数 ( $Cr = v \cdot \Delta t / \Delta X$ ) は $<1$ でなければならない。
    - $D$  = 散布係数 [ $L^2/T$ ]
    - $\Delta t$  = 時間の段階 [ $T$ ]
    - $v$  = 速度 [ $L/T$ ]
    - $\Delta X$  = グリッドの間隔 [ $L$ ]
  - ・もしランダムウォーク粒子のトラッキングが使用されたならば、グリッドの間隔と粒子の質量は汚染物質解像と関連して解析されなければならない。
- 48 グリッドにおける境界条件の場所の地図作成は評価されなければならない。例えば以下の通りである。
  - ・モデルにおける不安定さを弱らせるため、境界は対象となっている場所から十分離れたところに位置しなければならない
  - ・グリッドに表わされている境界のかたちは、グリッドの良質さ、幾何学の正確さと境界条件の正確さを検証しなければならない
  - ・有限要素グリッドにおいては、内外の境界が要素境界と一致しなければならない
- 49 井戸のノードは井戸の物質的場所の近くにななければならない。
- 50 モデル入力データのデータソース、データ収集手順、そしてデータ不確実性は、プロジェクトレポートまたはファイルに評価されドキュメント化されなければならない。
- 51 全てのモデル入力、それらが測定、概算または仮定のどれに当てはまるか明確にされなければならない以下を含む。
  - ・構造する係数とパラメータ（すなわち、一般的には観測することができないが他の変数の観測から推測されなければならないパラメータ、例えば伝達可能性や特定蓄積の散布）
  - ・強制用語（例えば、水や溶解した汚染物質のソースとシンク）。
  - ・境界条件
  - ・初期条件
- 52 データがモデル入力に変換される入力概算過程は（例えば、時空の補間と補外またはクリギング）説明されなければならない、補間を行うために使用された空間の場所とデータの連合した値は地図に示すか表にて提供されなければならない。
- 53 入力概算過程と連合した不確実さは明確にされ、説明されドキュメントにされなければならない

い。

- 54 モデルの検証をしなければならない。
- 55 モデルが検証されていない場合、モデルを検証しなかった理由を説明しなければならない。
- 56 検証過程を終結させるために使用されている規準は（すなわち、観測値とモデル化された値間の適切な組み合わせの定義づけ）モデル開発目標と関連して調整されなければならない。
- 57 検証は一般的に受け入れられる形で行わなければならない。明確には以下の通りである。
  - ・検証において調査されるべき主要パラメータと境界条件を決定するために、感度解析が行われなければならない
  - ・検証はシミュレートされた値と測定された値の間の誤差の計算を含まなければならない
  - ・検証は時空の誤差の評価を含まなければならない
  - ・検証は物質的特性に関連して行われなければならない（例えば、土手、陥没、または地下水流量およびかん養を含む地下水輪郭のパターンに関連して誤差は解析されたか）
- 58 水予算が展開された場合、モデルを検証する際のそれらの結果や使用が説明されなければならない。
- 59 検証による初期モデルパラメータ値の全ての変化はその合理性に対して調整されなければならない。
- 60 概念モデルにおける検証されたモデルパラメータと概算されたパラメータ範囲の間のあらゆる不一致は調整されなければならない。
- 61 モデル検証の結果、概念モデルが修正された場合、概念モデルにおける全ての変更は調整されなければならない。実行できるときはいつでも、検証されたモデルは独立したフィールド観察のセットによって検証されなければならない。

#### (6) シナリオのシミュレーション

- 62 各モデル開発シナリオにおいて、モデル入力とモデルグリッドにおける特性の場所は調整されなければならない。たとえば以下の通りである。
  - ・あるノードにおいてポンプ式井戸が位置してなかった場合、隣接するノードの井戸陥没の配置が調整されなければならない
  - ・スラリー壁が代わりの補修の場合、壁の幾何学的水力学的特性のモデルの表示が調整されなければならない
  - ・掃除の時間が計算されている場合、汚染物質の場所、量、状態に関する全ての仮定は調整されなければならない
  - ・井戸抽出などの補修行動が流動に影響を与える場合、各モデル化された抽出井戸の沈滞点への下降傾斜と側面範囲を含めてそのような影響は決定されなければならない。

#### (7) ポストシミュレーション解析

- 63 サイトシナリオをシミュレートする際のモデルアプリケーションの成功は評価されなければならない。
- 64 このアセスメントは以下の解析を含めなければならない。
  - ・モデル開発シミュレーションが現実的であったかどうか
  - ・シミュレーションが正確にシナリオに反映したか
  - ・水文地質学的システムが正確にシミュレートされたか
  - ・概念モデルのどの見解が成功してモデル化されたか
- 65 モデルの感度はサイト特定設計パラメータにおいて結果として不正確さが生じ、モデル検証におけるエラーのレベルは調査され量化されなければならない。例えば、モデル開発シナリオは、より感度の高い水文地質学的パラメータの可能な値の範囲にシミュレートされなければならない。さらに、モデル結果についての結論を出す際に、モデル検証におけるエラーの範囲

が考察されなければならない。

- 66 ポスト処理は、それがモデル開発結果を正確に表わしていることを確認するために解析されなければならない。適切などころで補間やスムージング方法はドキュメントにされなければならない。
- 67 ポスト処理結果はそれらがモデル開発目標を支持するということを確実にするために解析されなければならない。
- 68 最終発表はモデル開発結果を効果的に正確に伝達しなければならない。
- 69 実行できるときに、未来にモデルのポスト監査の実行または計画をしなければならない。

## (8) 全体的有効性

- 70 モデルアプリケーションにおいて遭遇したどのような困難もドキュメントにされなければならない。
- 71 モデルアプリケーションは管理側が意思決定のために捜し出した情報を提供しなければならない。
- 72 モデルアプリケーション結果は他の関連した団体にも受け入れられなければならない。
- 73 モデルアプリケーションはタイムリーで効果的な決定を支持しなければならない。
- 74 モデル開発労作の見解で、あと知恵に、別の方法で行われたかもしれない部分はドキュメントにされなければならない。

## 5. 7 品質管理とコード選択

### 5. 7. 1 コード選択過程

モデルアプリケーション過程において、コード選択はプロジェクト労力と結果の間の最善のトレードオフを確実にする決定的な段階である。この結果は、一般的に予測正確性に関するモデル開発労作の期待された有効性と表現される。労力は最後にプロジェクトのコストによって表わされる。そのようなコストは、フィールドデータ取得のそれと独立して考察されてはならない。モデル開発コストの適当な評定のため、新しいコードの開発または存在するコードの取得、実行、保守、そしてコードのアップデート、そしてデータベースの開発と保守の測定が考察される必要がある。

コード選択は本質的にはモデル開発必要性の詳細な説明と存在するコードの上手く定義づけられた特性との組み合わせのため、適切なコードを選択することは、存在するシミュレーションコードの体系的特性記述と同様に、モデル開発必要性の公式化を必要としている。モデル使用において、コード信頼性は主要な問題である。従って、十分なレビューと検査を受けた適確なシミュレーションコードの使用を確実にするため、選択過程においては特別な注意を必要とする。モデル開発プロジェクトの成功のためには、適切なモデルの選択が決定的である。

### 5. 7. 2 モデル開発必要性の公式化

能率的にシミュレーションコードを選択するためには、そのようなモデルの評価と受け入れのため、管理者向けの規準を開発する必要がある。そのような規準を得るためには、着手すべき管理側の目標と捜し出された解析のレベル（関連している間違っただ、または不正確な解答またはリスクに対するプロジェクトの感度に基づく）が把握されなければならない。さらに、研究されている物質的システムの適切な知識が存在しなければならない。研究に利用できる人材と物質的資源における制限の解析を完

成させなければならない。選ばれた選択規準のセットは以下に基づかなければならない。

- ・コードの実行コスト（要求された解析レベルのデータ取得を含む）と正確性  
の間のトレードオフ
- ・モデルユーザーのプロフィールと要求されるユーザーフレンドリネスの定義
- ・努力、コスト、制限に関するアクセス可能性
- ・受け入れられる時空の尺度と集合のレベル

これらの考察に基づいて、ユーザーは研究されている問題に関連したコード特性を公式化し、存在するコードの特性を地図に入れるためにこれらの特性を優先しなければならない。コード特性は、その選択過程に集中されることもあるが、モデルの目的、表わされた地下水システムの特質、そして使用された数学的方法を含む。さらに、ユーザーはモデルの正確さ、文書の完全性、そしてソフトウェアの品質、性能、そして使用性に関心があるかもしれない。ユーザーがコード選択において関連があると思うかもしれない他の情報は、それが実行するコンピュータシステムとその使用のための条件である。

### 5. 7. 3 コード特性の記述

地下水モデルには様々なタイプがあり、それらは様々なタイプの地下水システムをシミュレートするために設計され、多様なシステム変数を計算することができる。ある特定モデルの主要な特性を確認するためには、標準化され定義されたモデル記述子を使用した記述的システムが必要である。地下水モデル開発コード記述子のデータベース開発の一部として、国際地下水モデル開発センターは何百もの段階別に分類された記述子を含む体系的なリストを開発した。この記述的システムのために採用された詳細は、ユーザーに関心のあるシミュレーションコードを確認し、準備的な使用性アセスメントを行うことを可能にする。

シミュレーションコードの記述は、モデルの目的、表わされた地下水システムの特質、そして使用された数学的方法の目的を網羅する。さらに、それはモデルの正確さ、文書の完全性、そしてソフトウェアの品質、性能、使用性を指定する。最終選択が行われる前に、ユーザーはコード文書を手し、コード特性を詳細に評定する必要がある。また、ユーザーは入手できる様々な数値方法の一般的性能特性を認識しなければならない。各方法はそれぞれ長所と短所があり、それは左右する方程式、適用可能な境界条件、そして、関心のあるパラメータの散布の条件に含まれた種類と関連していることがある。

### 5. 7. 4 コード選択規準

コードの選択における主な規準は以下の通りである。

- ・コードが意図された使用に適確であること
- ・コードに信頼性があること
- ・コードが効率的に適用できること

様々な問題が解決されなければならない場合には、1つ以上のコードが必要になるかもしれないし、あるいは1つのコードがひとつ以上の能力に使用されるかもしれない。そのような場合、提出された各問題のモデル要件は選択過程の最初に明確に定義づけられなければならない。ある部分には、プロジェクトの様々な段階の同じシステムをモデル開発する場合においても事実である。しばしば、コードは問題を広げたりシステム概念化を補助するためにプロジェクトの初期段階に選択される。データ使用可能性における時間と資源の限定は、初期に公式化を「シンプル」なモデルにするかもしれない。システムとデータの使用可能性の理解の増加は、増大する複雑さや結果として生じる様々なコードの選択の成功を導くかもしれない。そのような場合、候補コードの柔軟性または様々なレベルの精巧さの統合されたコードセットの利用可能性が重要な選択規準になるかもしれない。

上記に述べられたように、コード選択における主な規準は、確認された作業の適確性、その信頼性、そして効率性である。コードの信頼性は、その開発中に適用された品質保証のレベル、その検証、フィールド妥当性検査、そして性能評価によって決定される。コードの信頼性は、広範囲に認められるレビューや検査手順を適用することによって確立されなければならない。様々な組織がそれぞれ独自の手順を確立したにもかかわらず、そのような手順はまだ採用されていない。コードの効率性は、ソースまたは実行時間バージョンそして文書の使用可能性、そしてオペレーションに要求される人材やコンピュータ資源と関連した使用性、移植性、修正可能性によって決定される。シミュレーションコードの受け入れは技術的・科学的確実性、効率性、法的そして管理上の考察によって基づかなければならない。モデルの効率性は、コードへの急速なアクセス、良質な文書があること、ユーザーサポートへのアクセス、そしてその信用性、真実性、使用性、移植性、修正可能性、実行時間資源の利用状況、及び全体的な節約によって決定される。

### (1) 利用可能性

総称的シミュレーションコードは、それに関連したプログラムコードがソースコードまたは実行可能なバージョンとして得られる場合、またはプログラムがポテンシャルユーザーによって簡単にアクセスできる場合に利用可能と定義づけられる。地下水ソフトウェアは、公衆領域もしくは所有権を持つステータスにある。米国においては、連邦や州の政府機関あるいはそのような機関からの資金による大学によって開発されたほとんどのソフトウェアは、その使用と配布において無制限に利用可能であり、従ってそれは公衆領域に入っているとみなされる。他の国々においては状況が異なり、たとえ政府援助によって開発されても殆どのソフトウェアは所有権を持つステータスにあるか、またはステータスが明確にされていない。そこでは、コードはその使用、複製、配布のある制限のもとで入手またはアクセスが可能になる。

また、コンサルタントや私営の企業によって開発されたコードはしばしば所有権を主張するものである。これはいくつかの大学や私営の研究機関によって開発されたソフトウェアにおいても同様だと言えるかもしれない。一般的に、所有権を主張されたコードは著作権の法律によって保護されている。いくつかの総称モデルのソースコードがテキストなどの出版物に発行され、出版社からのテープやフロッピーディスクによって入手可能ではあるが、その使用と配布は発行の著作権に制限されている。コードが数学的またはグラフィックのサブルーチンなど、所有権を主張する第3者ソフトウェアを含む場合、さらなる制限が生じる。公衆領域コードにおいては、そのようなルーチンは外部的であり、プ

プログラムを成功して実行させるためにホストコンピュータでの存在が要求される。公衆領域と所有権を主張するソフトウェアの間には、フリーウェアまたはユーザーサポートされたソフトウェアと呼ばれるものがある。フリーウェアは自由に複写し配布されてもよいが、ユーザーはこのようなソフトウェア開発を任意の寄付によって援助するよう奨励されている。地下水モデル開発においては、このようなタイプの配布方法は稀である。

公的資金によって開発されたいくつかのコードにおいては、例えばソフトウェアが輸出されたり、広範囲な保守とサポート設備が作成された場合、配布制限が実施されている。後に述べたケースでは、品質保証のされていないバージョンの使用防止、ソースコードの確認されていない修正を予防するため、そしてコントロールされたユーザーグループに効率的なコードのアップデートをサポートするために、制限が施行されている。

## (2) ユーザーサポート

サイト特定設計問題のために、あるコードが選択された場合、ユーザーは利用できるコンピュータシステムにおいてコードを実行する際に技術的問題に遭遇するかもしれない。そのような困難は以下の結果から生じる可能性がある。

- ・コードの開発されたコンピュータとユーザのコンピュータの間の互換性の問題
- ・オリジナルのプログラムにおけるコーディングエラー
- ・データ入力やコード実行におけるユーザーのエラー

コードにより精通することによって、ユーザーに関するエラーを減らすことができる。ここでユーザーに良質な文書が役立つのである。慎重なるコード選択の後に実現や実行に問題が起これば、文書から答えが見い出せなかった場合、ユーザーはそのコードのオペレーションと使用に関して知識のある人の助けが必要となる。ソフトウェア実現と実行サポートまたはソフトウェアユーザーサポートと呼ばれるそのようなアシスタンスは、一般的にはコードの使用やモデル開発の適切なトレーニングの必要性にとって代わることはできない。コード開発者からの補助の要求は、ユーザーサポートがコンサルタントサービスまたは職場内訓練になってしまうほどの広範囲な割合とみなされるかもしれない。この可能性は一般的にはコード開発者によって認知されているが、ソフトウェアのユーザーにはいつもそうとは限らない。

## (3) 真実性

地下水シミュレーションコードの使用における主要な問題は、その真実性にある。コードの真実性、そしてそれが表わす理論的構造は、その証明された信頼性とユーザーの受け入れに基づかれている（成功したアプリケーションの数の一部に基づいている）。モデラーや地下水管理者は、その頻繁に適用されるシミュレーションコードに最大の確信を持つことがある。この見解は、成功した同僚によくレビューされたアプリケーションが発行されたときに強化される。

プログラムの信頼性は、ソフトウェアエラーによって起こる可能性のあるローカライズされたエラーや端末失敗と関連しているため、そのような本来は広く使用されているプログラムにもあるエラーのほとんどは見抜かれ、訂正されているとみなされている。長い使用とアップデートの歴史にもかかわらず

わらず、依然としてプログラミングエラーの無いコンピュータプログラムは存在しない。いくつかのエラーは決して見抜かれることはなく、プログラムのユーティリティーに影響しないか、またはほんの少しの影響しか与えない。他のエラーは例外的状況または試験されていない状況においてのみ発生する。

シミュレーション結果に基づいた決定は、使用されたコードが適切なレビューと検査を行われているときのみ実行できる。しかしながら、フィールド妥当性検査（もし存在するならば）やコードアプリケーションの感度に頼りすぎるのは、いくつかの科学的に確かな、上手くドキュメントにされた、徹底的に試験されたコード、そして手元にある最も問題解決に効率的なものさえも除外することになるかもしれない。コードが多人数によって使用された場合、これが重要なユーザーの確信の証拠となるということが論じられるかもしれない。そのような広範囲な使用は様々なタイプの地下水システムへのコードの適用性や様々な管理側の問題に着手する能力を反映させるであろう。

#### （４）使用性

シミュレーションコードがユーザーのコンピュータシステムに実現された場合、様々な問題に直面する。そのような困難は、ハードウェアの不適合性や、コードのインストール、データ入力、またはプログラム実行などのユーザーのエラーのコーディングによって発生するかもしれない。オペレーション上の特徴の急速な理解と知識を実行するプログラムで使用が簡単なものは、ユーザーフレンドリーと呼ばれ、その使用性によって定義される。そのようなプログラムでは、一般的に広範囲によく編集された文書が強調されている。簡単な入力準備とコード実行で、オンライン補助システムのある（インテリジェントな）プレプロセッサの使用が望ましい。そしてよく構造化された有益な出力。適切なコードサポートと保守はコードの使用性も強める。

#### （５）移植可能性

ひとつの実行環境から他へ簡単に転送できるプログラムは移植可能プログラムと呼ばれる。プログラムの移植可能性を評価するためには、ソフトウェアとハードウェアの、両方の従属性が考慮されなければならない。新しいコンピュータ環境でプログラムを実行することが必要な場合、その修正可能性が重要である。

#### （６）修正可能性

コンピュータプログラムの有益な寿命のうちに、ユーザーの経験と管理側の変化する要件はソフトウェアの機能上の詳記の変化に導かれることがある。それに加えて、科学的開発、変化する演算環境、そしてエラーの持続がプログラムの修正を必要にする。ソフトウェアがある一定期間以上使用される場合は、そのような成り行きにペースを保てるよう、継続して修正できるよう設計されなければならない。修正が難しいコードは、壊れやすく維持能力が欠如しているものと呼ばれる。そのような困難は、地域的変化のグローバルな広範囲プログラムの実現によって生じる可能性がある。

#### （７）ランタイム資源効用性

ランタイム資源効用性とは、モデル開発計算を実行するために必要とされている計算資源を指す。



主な資源は、要求されるハードウェア装置やシステムソフトウェアの種類や数、コンピュータコアメモリ（RAM）使用、プログラム実行（CPU）時間、入出力処理（I/O）時間、そして質量蓄積要件を含んでいる。コンピュータコアメモリの使用は、プログラマーによってコードの中で前もって決定されることがある。他の場合においては、ユーザーはセットアップされた問題を特定することによって、コアの使用を間接的に決定する。質量貯蔵スペースの需要とともに、プログラム実行とI/O時間は、代表的ベンチマークのひとつ組みを実行させることによって測定が可能である。

#### 5. 7. 5 コード選択における品質管理

要求された特性と利用できる総称モデルに存在する特性の間には完璧な組み合わせは稀であるということを確認しなければならない。ある特定のアプリケーションに重要と判断された選択規準は主観的もしくは調整が弱いことがあり、それはモデル化されるべきシステムのある特性の重要性を確立するためのプロジェクト内の選択段階におけるデータが不十分であるからである。組み合わせの入手が困難な場合、これらの規準と選択過程におけるそれらの相対的比重の再アセスメントが必要である。このために、コード選択は巡回過程であり、成功は熟練した判断によるのである。

コード選択における品質保証は、コード開発やモデルアプリケーションの品質管理の見解において説明されたのと同様のパターンに従う。コード選択品質管理における主要な要素は以下の通りである。

- ・ 上記に説明されたように注意深く一步一步の選択手順に従うこと
- ・ この手順に執着した詳細な報告
- ・ 各段階に行われた考察、仮説、そして決定の文書である

明確には、研究されている問題のために選択されたコードがなぜ適切であるかということが詳細な調整として示されなければならない。研究されている問題に関連した選択コードの制限が説明されなければならない。

## 6 システムの試作

過去10年の情報テクノロジー分野における根本的な発展は、非常に急激であったと特徴づけられる。情報システム開発のための新しい支援方法が普及し、その支援方法による直接的な効果により、情報処理に関する問題は解決可能になりつつある。現在利用できる多くのシステム開発戦略、構造化開発手法及びソフトウェア・ツールは、情報処理問題の解決のための方法を選択する際に必須なものとなっている。

### 6.1 問題解決へのアプローチ

情報に関する問題が発生した場合、まず誰がその問題解決に最も適しているか、つまりエンドユーザーか情報システム部門かを定める必要がある。場合によっては、先進的で親和性のあるツールを使用することによって、エンドユーザーが自主的にアプリケーションを開発することも可能であるが、多くの場合、ユーザーが自分自身のシステムを開発することは不可能である。有効な情報システムを効率的に開発するため、今日では専門家であるシステム開発者は、あらゆる戦略、手法及びソフトウェア・ツールの使用が可能であり、それらの戦略・手法は、プロジェクトを計画し、統制するプロジェクト管理のためのマネージメント支援、及びアプリケーション・システム開発担当者のための開発支援と大別することができる。プロジェクトの初期段階で、プロジェクト・リーダーはプロジェクト・モデルと、開発方法、手法、技法及びツールを選択しなければならない。この選択は、問題とその内容の性質に左右されるが、すべての情報システムを同じ開発支援を使って開発する必要性はない。

開発戦略に対するのと同様に、プロジェクト管理に対する最良のアプローチは単純なものではない。プロジェクトの特質に基づいてプロジェクトを計画し統制する方法は、現在利用できるものの中から選択する必要があるが、その方法はプロジェクトのリスクを基に選択を行なうことが考えられる。プロジェクトのリスクは相対的な規模、採用するテクノロジーの経験量、それに“プロジェクト構造”（問題の構造化度合い）によって決まる。与えられた問題の状況に最も適した戦略、手法、ソフトウェア・ツールを選択するための具体的なガイドライン作りが様々な分野で行なわれている。

要求定義段階での1つのアプローチはプロトタイピングである。ユーザーと開発担当者にとって、プロトタイピングの概念は有益と考えられている。新しい問題が起こるたびに、その利点と欠点を考慮せずに、この方法を選択する傾向があるが、プロトタイピングは万能ではない。つまり、プロジェクトの初期段階で、伝統的なダイアグラム技法とプロトタイプとを組み合わせることが意味があるかどうかを検討すべきである。プロジェクト・リーダーは個々の状況に応じて、完全で、厳密で、確固たる要求定義を行なうため最良のアプローチを決めなければならない。

プロトタイピングはユーザー・インターフェイスのモデル化に限定されるかもしれない。従来の手法では、プロトタイプのユーザー・インターフェイスを作成しない場合、システムの一面が現実には無視されることになる。スクリーン・ペインタのようなツールを使用することにより、ユーザー・インターフェイスは紙の上でモデル化するよりさらに迅速に作成が可能になり、また修正を加えることが可能になる。したがって、ユーザー・インターフェイスが、開発中のアプリケーションにとって重要である場合や、ユーザーがユーザー・インターフェイスやシステムの最終的な形に関して見解を持っている場合に限られるが、ユーザー・インターフェイス・プロトタイピングは実用的なものとして推奨される。プロトタイプを採用すべきか否かは機能型プロトタイピングを考えた場合、選択が難し

くなる。機能型プロトタイプと従来の作図技法を使って作るモデルとは、システムの同じ側面を描くことになり、そのようなプロトタイプを作るのは費用的に問題が生じるからである。

### (1) プロトタイピング手法

ある特定の状況におけるプロトタイピングの使用は、経済的な観点から検証することが望ましい。つまり、予測される利益は予測される費用より大きくなってはならないが、容易に推測することができない。確かな計数的なデータを得るには、伝統的な方法とプロトタイピングを使って開発した方法の多くのシステムについて開発費用を測る必要がある。そのシステムは“ノイズ”を濾過するための統計手法を用い、管理が行き届いた状態で使用されることが必要である。

プロトタイピングの相対的費用、あるいは利益は、ライフサイクルの費用をそれぞれのカテゴリごとに比較した結果、推測することができる。しかし、このような実験には費用がかかり、長期間にわたって続ける必要があることから、この種の実験的なデータを得ることは困難である。この分野でなされたわずかな調査の結果では事実上比較できない。というのも“プロトタイピング”という用語の解釈はそれぞれ異なっているからである。

ソフトウェア開発の費用は使用する戦略だけでなく、他の多くの要素とも関係する。開発費用の見積りは、プロジェクトの初期に適用できるものが望ましい。この分野の研究では、ソフトウェア開発費用は開発担当者に大きく依存する。つまり問題となるのは、手法に関係なく、その手法を適用する担当者ということである。さらに、より先進的な言語は開発費用に多大な影響を与えるということが考えられる。プロトタイピングの費用と従来のアプローチによる費用との比較調査をする場合、他の重要な費用推進力の影響が小さくなることが考えられる。この種の研究は、どの方法によって最も生産性が向上するかを判定する代わりに、システム開発関数の生産性に対するプロトタイピングの影響を確立することを目指している。

### (2) プロトタイピング手法の恩恵

プロトタイピング・アプローチを採用するメリットは、主として情報問題の本質についての不確実性と、その情報問題を解決するためにユーザーが提示する要求の不確実性を減少させることにある。

“不確実性”とは、ある問題に関する既存の知識と、その問題解決に到達するために必要となる知識との差である。伝統的な技法も、この不確実性を排除することを目指しているが、成功するまでには至っていない。

保守費用の情報問題についての不確実性と、その問題を解決する情報システムの保守に要する費用とは直接関係する。つまり、不確実性を減らすことによって保守費用は下がることになる。このことから、情報システムの全ライフサイクル費用の50%は要求定義の質の悪さが直接の原因となっている。したがって、プロトタイピングを利用した代表的な例では相当な節約が可能となる。しかしながら、ユーザーの要求と要望について不確実性がない場合、つまり明確としていれば、要求定義に対するこのアプローチの利点はなくなる。この状況では、従来のアプローチもプロトタイピングも同じく（あまり）保守費用が減ることになる。個々の状況でプロトタイピングの潜在的な利益を査定しようとすると、どの程度の不確実性が存在するかを明らかにしなければならない。

不確実性はシステムの外的側面だけでなく概念的側面とも関係する。実際、概念レベルについての

不確実性は非常に深刻な影響をもたらす。この場合、システムの機能について、またデータベースの構造について明快さを欠くことになる。この分野における誤りや欠陥は、一般に根本的なそして費用を要する修正が必要となる。システムが取らねばならぬ外観についての不確実性は、費用面で重要な影響をもたらすことにはなりそうにはないが、満足感に対しては重大な影響を及ぼす。この分野で明快さを欠くと、ユーザーとの親和性には乏しくなるが、要求されたことを一応行なうシステムにはなる。ユーザー・インターフェイスを修正することは、システムの機能変更を行なうほど根本的な問題とはならない。以下に示す規準を基に、ある状況で概念の不確実性が存在するか否かを判定するのは可能である。

(a) 支援が必要な仕事から要求を演繹する。

支援を必要とする仕事を分析することから、情報システムへの要求が出てくる場合が多くある。そこでは以下の特徴が見受けられる。

(b) 新しいシステムが支援しなければならない工程は完全に定義される。

このことは仕事が決まりきった、あるいは繰返しが多いということと直接関係する。

(c) 支援が必要な組織の役割は、構造面においても運用面においても堅固である。

情報システムへの要求が支援を必要とする組織の役割を分析することによって作成できるとなると、つまり支援を必要とする仕事から直接要求を作り出せるとなると、不確実性の問題が深刻になるといことはほとんどない。またこのような場合、開発者とユーザーとの間のコミュニケーションに問題が起こることはなく、システムの仕様はかなりしっかりしたものとなる。

(d) ユーザーおよび分析者の知識と経験のレベル。

新しいシステムの要求定義がすべて支援を必要とする仕事から出てこないとなると、新しいシステムを使うことになる将来のユーザーやそのシステム開発に携わる情報分析者の見識に頼らざるをえなくなる。問題の場面にいかかわっている人々をよく理解すればするほど、“本当の”情報問題が明らかとなるチャンスが増え、またユーザーがそのシステムを使い出した後に要求定義の変更が起こるといったことは少なくなる。情報問題とその解決策に対するユーザーと情報分析者の見識は特に以下の点に左右される。

(e) 関係するアプリケーション分野に対する知識と経験及び同じような、情報システムに対する以前の経験。

一般的に自動化された情報システムについての経験支援を必要とする組織の役割の安定性や開発するシステムの複雑さ及びユーザーも情報分析者も、いずれも問題の状況について十分な知識をもっていないとすると、プロトタイプングは唯一の正しいアプローチとなる。しかし、少なくともどちらかが問題状況について、また開発するシステムの要求について十分な見識をもっていれば、要求を習得するためのモデルを利用する必要性はなくなる。情報問題が理解されると、ユーザーがそのシステムを使用し始めた後、システム要求定義が変わる危険はほとんどなくなる。

(f) コミュニケーション問題の可能性。

ユーザーと情報分析者が互いに徹底的にコミュニケーションする必要があるのは、情報のニーズについての不確実性を同じ言葉で話せない双方の当事者が原因となって起こる。この場合、不確実性は情報問題を理解できないからではなく、理解するためのコミュニケーション能力が低いからである。コミュニケーション問題は次の場合危険度が高まることになる。

(g) ユーザーが使用される分析用言語に経験がない。

ユーザーが情報分析者の使う言語に否定的な態度を示す。ユーザーが抽象的思考能力を前提とする言語にうまく対処できない。情報分析者が関係するアプリケーション分野とその分野の用語についての経験をもっていない。

概念レベルの不確実性はシステム要求が仕事の分析からでなく、ユーザーも情報分析者もアプリケーション分野の知識と経験がなく、そして彼らの間でのコミュニケーションが取れない場合に最大の問題となる。このような環境では、プロトタイピングは相当な時間的節約となるのはほぼ間違いない。

その他の利益として不確実性を減らし保守費用を削減できるほかに、プロトタイピングを使うことによって、より一層の節約が可能となる。従来の直線的に進められるプロジェクトと比較すると、テスト（機能）期間は短縮できること、また費用も要せず済む（これはプロトタイピングを用いることにより、このフェイズの費用がなくなるからではなく、要求定義フェイズにテストのほとんどが移ってしまうからである）。新しいシステムの組織への導入がより円滑に進むことから、プロトタイピングはさらに利益をもたらすことになる。

第1に、一般的にシステムは組織に適度に取り入れられ、次にプロトタイピング作業に係わったユーザーは、既にそのシステムを熟知していることになる。プロトタイプを本番のシステム（一部の）のベースとして使用できれば、保守費用の削減に加え、設計および実現フェイズにおいても相当な節約となる。この場合、情報システムの重要な部分はプロトタイピングの後には既に使えるようになっていることある。この状況はプロトタイプが本番システムで使われるのと同じ言語を用い、そして同じ（技術的）環境で構築される場合、特にその影響がある。ユーザーは初期の時点で自由にプロトタイプを使って十分有効なシステムを構築することができる。

### (3) プロトタイピング手法の経済性

プロトタイピングの費用は、使用するツールの品質と先進的な機能に大いに左右されることは明らかである。ユーザー・インターフェイスをスクリーン・ペインタとレポート・ライターで作成した場合、その費用は従来の開発ツールを使うよりも何倍も安価にできる。強力な手続き型言語および非手続き型言語を統合化された開発環境の一部として使用することにより、プロトタイプの開発および変更の両方の費用低減が可能となる。優れたリレーショナル（あるいはセミ・リレーショナル）データベースを使用できる場合は、データベース構造の変更やデータベースの再編成を相当簡単に、しかも安価に行なえる。ネットワーク型データベースを再構築するには何倍も余計に費用を要し、もし階層型のデータベース管理システム（DBMS）だけしか利用できないのであれば、この仕事に取り掛かる前に少なくとも2回はよく考えることが賢明である。

階層型DBMSは、データベース構造をプロトタイピングの助けを借りなくとも正確に厳密に指定できる開発プロジェクトには適している。さらにプロトタイピングの費用は、プロトタイプを受け入れることができるようになるまでに必要となる繰返し作業の回数にも関係してくる。この繰返しの回数は、プロトタイピング工程の開始時におけるシステム要求の確実性がどの程度かに左右される。優れたツールと完全に構築されたプロトタイプがあれば、変更作業の費用は高くなることはない。先進的なツールが使える場合ですら、プロトタイピングに関する費用は伝統的な分析費用よりも嵩むことになる。従来の方法であれば、不確実性の程度が非常に高いと、要求は不確実なまま残るという結果

になる。同じような状況でプロトタイピングを使った場合、やらねばならない仕事としては従来のプロジェクトの場合と同じである。しかしこの場合、この不確実性を減らし受け入れ可能なレベルまで到達するには、システムの側面、つまり部分について複数のプロトタイプを構築すること（そしてユーザーに評価させること）も必要となる。

#### (4) プロトタイピング手法の費用対効果

プロトタイピングによってもたらされる諸々の節約は、主として保守フェイズでなされ、プロトタイプが本番システムのベースとして機能する場合でも、設計および構築フェイズではそれほどではない。情報問題とその解決策について不確実性がある場合にのみ節約が可能となる。要求定義フェイズでは、プロトタイピング・アプローチのほうが余計に費用を要す。このようにプロトタイピングは新たな費用パターンを生み出すことになる。その原因を以下に示す。

- a. 要求定義フェイズでは従来のアプローチを用い、本番システムの構築には第3世代言語を用いる。
- b. 要求定義フェイズではプロトタイピングを、そして本番システムの構築には第3世代言語を用いる。  
(プロトタイピング作業の終りでプロトタイプは棄てられる)。
- c. 要求定義フェイズでは従来のアプローチを、そして本番システム構築は第4世代言語で行なう。
- d. 要求定義フェイズでプロトタイピングを、そして本番システム構築には第4世代言語を用いる。  
(プロトタイピング作業の終りにプロトタイプは棄てられる)。
- e. 要求定義フェイズではプロトタイピングを、そして本番システムのベースとしてプロトタイプを使用する。

プロトタイピングを正しく理解するには、シナリオ (a) はシナリオ (b) と比較し、シナリオ (c) はシナリオ (d) と、シナリオ (d) はシナリオ (e) と比較しなければならない。第4世代言語の利用によって変化した費用パターンは、シナリオ (a) と (c)、およびシナリオ (b) と (d) を比較すると明らかになってくる。図6. 1-1では、プロトタイピングが適用され、また第3世代言語による従来の開発 (シナリオ (a)) の代わりに本番システムのベースとして残る場合 (シナリオ (e))、いかに節約が最大となるかを示している。しかし、これらの節約がすべてプロトタイピングの利用によるものだけではない。

図6. 1-1を理解するには、この図が単に1つの徴候を示しているにすぎないことに留意しておかねばならない。絶対的な意味がヒストグラムの棒の最高位にあるということではない。この図は多くの仮定を基に作られており、経験的なデータによりその一部のみが認められているにすぎない。もちろん何らかの変更がこれらの仮定になされると図も変わることになる。これらの仮定は以下の通りである。

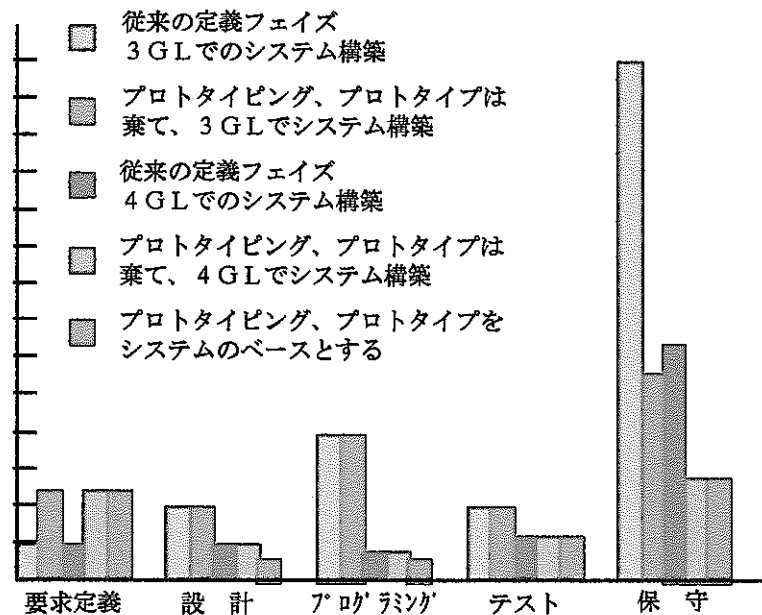


図6. 1-1 第4世代言語利用のプロトタイプ費用予想

- ・情報システムの全ライフサイクルの費用の中で、60%は保守の費用である（システムの出荷後の費用が発生する）。この数字はIBM社の最新の調査からのものである。
- ・第4世代言語を使用するが分析と設計が従来の方法で行なわれる場合（シナリオ（c））、保守費用は依然ライフサイクル費用の60%となる。システム要求定義の質も同程度であることから、この数字が下回ることはない。
- ・従来のプロジェクト工程の様々な開発フェイズにわたる費用の分布はZelkowitz他（1979）からのものである。しかし、これらの数字はそれぞれの研究者によって異なっている。というのも、研究者それぞれ各自異なるフェイズを定義しており、また費用のパターンは情報システムの種類と規模に大いに左右されるからである（Boehm, 1981）。
- ・第4世代言語を使った結果生まれる節約は、Rudolph（1983）によって調査されている。図の中で示されている節約は（シナリオ（a）と比較した（c）とシナリオ（b）と比較した（d））、Rudolphの研究結果と比較すると控え目に見える。
- ・要求定義について多くの不確実性があれば、この不確実性を除去するために複数のプロトタイプを作る必要がある。これらの費用は第4世代言語で本番システムを設計し実現化するために要する費用を越えてしまう。
- ・全保守費用の60%はシステム要求定義が不十分なことによる（IBM）。プロトタイプングを通して、これらの費用を除くことができる。

このことはプロトタイプングが終わると、システム要求定義はエンドユーザーの要求と希望を完全な形で表現でき、そして情報システムを長期間使ったとしても利用者のニーズは変わらないということを前提にしている。個々の問題でプロトタイプングの経済的評価が求められる場合、まずプロトタイプを本番システムのベースとして使用するのかどうか、あるいは捨て去られてしまうのかどうかを

判断しなければならない。もし前者であれば、プロトタイプはほとんど常に推奨するのに値するものとなる。何も不確実なことがないような状況だとすると、プロトタイプは単に本番システムを構築する、また別の1つの方法であるにすぎない。

要求定義フェイズでの余分な費用は、設計と実現フェイズの費用が安くなることで埋合せができる。このような場合、要求定義フェイズでの繰返し回数は多くなることはない。情報システムについて不確実性があると、プロトタイプはその不確実性を減らし、したがって保守フェイズにおける節約に非常に役に立つことになる。これに対して、要求定義フェイズの費用の増加を比較検討しなければならないが、多分繰返し作業が何度も必要になろう。しかし、この段階でもなお、情報システムにかなり容易に変更を加えることができる。

保守フェイズで変更することは何倍も費用を要することになる（システムが第4世代環境で実現されたとしてもである）。これは実際運用されているシステムが性能やセキュリティも含めてすべての要求を満たさねばならないからである。このような要求を満たすため取られる方法はシステムの複雑さを増し、したがって柔軟性が乏しくなる。反対にプロトタイプは一般に不完全であり、多くの制約を無視することができ、作成されたプロトタイプは複雑ではなく本番システムより変更がやさしい。本番システムの変更に伴う費用が高くなるもう1つの理由は、そのようなシステムは既に組織の中にがっちり立場を固めているという事実である。

出荷後のシステム変更は、しばしば組織上の手続きまで変えねばならないことがある。プロトタイプをプロトタイプ作業の終了後捨て去る場合、費用便益性が多くの注目を浴びることになる。この場合、要求定義フェイズでの費用の増加は、設計および実現化のフェイズの節約とバランスがとれていないことから、プロトタイプでの余分な費用は保守費用に見込まれる削減額を越えることはできない。情報問題の不確実性の度合いが大きいと、保守に予想される費用を相当節約できるとの観点から、プロトタイプのアプローチは評価されることになる。

もし不確実性が何もないければ、プロトタイプに従うことは必要ないし、また賢明なことでもない。したがって、要求定義フェイズでの費用の増加は保守費用の削減によって埋合せをすることはできない。このように、もしプロトタイプが本番システムのペースとすることができないならば、“厳しい”費用と利益の観点から、要求についての本質的な不確実性がある場合にのみプロトタイプの方法を利用することが賢明であろう。しかしながら、プロトタイプ・アプローチに従いたいという欲求にはさらに別の動機がある。

#### (5) 効果的問題解決

プロトタイプの有形の利益に加え、要求定義にこのアプローチを用いると、さらにいくつかの否定しがたい効果があるが、これらを測るのは特に難しく不可能ですらある。これらの無形の利益は、もちろん要求定義の方法を選択する場合も考慮に入れる必要がある。プロトタイプに関する多くの出版物の中で、このアプローチがエンドユーザーの開発工程と情報システム部門の両方に対する態度に実際及ぼす良い影響について取り上げている。

ユーザーの態度の肯定的な変化は、主としてプロトタイプ工程へユーザーが積極的に参加することと、システム開発担当者とエンドユーザー間のコミュニケーションの改善によるものである。ユーザーは高度なレベルでの参加により、開発工程の進展具合とその成果の両方に対して強い影響を及



ぼすことができる。ユーザーは、情報システムの形態と備えるべき機能をほとんど決めることができる。この結果、ユーザーは自分自身がシステムの“所有者”であるとわかる。理想的には、ユーザーと開発担当者が一緒にユーザーの要求と要望をできる限り最も正確に記述するために共同で作業を行なうことである。また、ユーザーの参加の度合いが高くなると、開発プロジェクトの進展に容易について行くことができるようになる。

プロトタイプが早い時点で利用でき、開発担当者とユーザー間の接触がプロトタイピングの過程で拡がると、プロジェクトはより見えやすくなる。これらの要因すべてによって、ユーザーは情報システム部門に大きな信頼を置くようになる。プロトタイピングについて書いてある本は、一般には論理的なものが多い。このアプローチについての実際的な経験は依然かなり限られており、経験的な研究についての出版物はまだ殆どない。しかし、プロトタイピングがユーザーの態度に肯定的な影響を与えるだろうとの予想は現実に確認されている。

表6. 1-1 開発工程へのユーザーの認識

	コミュニケーション	参加	衝突	理解
プロトタイピング・アプローチ	3.86	4.20	1.36	3.66
伝統的アプローチ	3.00	3.21	2.86	2.78
確率	0.13	0.0014	0.00149	0.10

出典：Alavi (1984a) を改作。(c)1984 Association for Computing Machinery, Inc.

表6. 1-1 がその結果である。見出しの欄は上に示された従属変数を指す。この表の最初の2つの欄は、これら4つの変数の平均得点である。統計上のテストを使うと (Mann - Whitney U Test)、様々な欄の得点の対が偶然によるものか、あるいは2つの戦略が関心のある点で本当に異なるのかをテストすることができた。表の最後の欄は測定誤差の確率である。つまり、0.05より小さければ、その差は統計的に有効となる。プロトタイピングに参加したユーザーは、伝統的な方法で固めたユーザーと比較して、開発工程をあらゆる面でより好ましいとみなしているのは、表6. 1-1 から明らかである。統計的に有効な結果としては、ユーザーの参加により高い満足度を示し、また開発担当者とユーザーとの間の摩擦には明らかにより低いレベルを示している。他の2つの点についての結果は統計的に有効ではないが、プロトタイピングにより開発担当者とユーザー間のコミュニケーションが容易になっており、また開発工程へのユーザーの理解が高まっていることを示唆している。この研究の結果は、比較的サンプルの規模が小さく、調査が実験室的な性格であることに影響を受けているかもしれないが、同じ著者が報告した現場でのインタビューから得られた結果と非常によく合致している。

このように、プロトタイピングは開発工程に対するユーザーの態度に肯定的な影響を与えることをこの調査は明らかにしている。情報システム利用者の満足度は、開発中のシステムの機能とユーザー・インターフェイスがユーザーに及ぼす影響が大きいことにより、プロトタイピングによりユーザーと親和性のあるシステムを作り出すことが期待されることになる。

- ・完成したシステムに対する全体的な満足度
- ・システムが作成するレポートの正確性
- ・情報システムのユーザーに対する親和性
- ・システムが作り出すレポートの有益性

表 6. 1 - 2 Mann - Whitney U Testの結果

	満足度	正確性	ユーザとの親和性	有益性
プロトタイプング・アプローチ	4.40	4.06	3.46	4.21
伝統的アプローチ	3.21	2.64	3.07	3.14
確 率	0.011	0.002	0.345	0.007

出典：Alavi(1984a)を改作。(c)1984 Association for Computing Machinery, Inc.

表 6. 1 - 2 は上記の点についての平均得点とMann-Whitney U Testの結果を示したものである。システムにより大きな満足を得ようとすれば、伝統的な方法よりプロトタイプングであることをこの結果は示している。システムの出力に対する正確性と有益性については両方とも非常に高い得点を示した。

プロトタイプングは、このように前の経験からの結果としてユーザーとシステム開発者との間で形成された不和を癒してくれる助けとなる。この要求定義のアプローチを使うことにより、開発工程に対するユーザーの態度の向上性が増し、長期的にみても同様な効果が期待できる。プロトタイプングにより、保守に必要な労力を大幅に下げることができる。保守の負荷は、新規アプリケーションを長時間待たす大きな原因となる。プロトタイプングをより広い範囲で使用すれば、最終的にはより多くの力を新しい開発へ向けることができる。そして情報システム部門は新規の開発要求に対してもっと柔軟に、もっとすばやく対応できるようになる。

## 6. 2 CASE

プロトタイプングが成功するには、RDBMS、スクリーン・ペインタ、能動型/統合化データ・ディクショナリ、VHLL（すべてが1つの開発環境に統合されていることが望ましい）といった強力なCASEツールを利用できることが必須である。このようなソフトウェアの支援がなければ、プロトタイプの開発は多大な費用を要し、またユーザーの要求や希望変更への対応が遅くて受け入れられないことになる。対応が迅速であることはプロトタイプング工程の有用性に著しい影響を与える。

さらに、構造化手法および技法を支援するためにCASEの機能が求められる。分析者用ワークベンチがなければ、作業現場で構造化手法・技法を正しく使うことは現実的に不可能である。分析者用ワークベンチは、システム要求定義を迅速に、しかも頻繁に変更できることが重要視されることから、プロトタイプング工程に適用される。プロトタイプングを使用するからとて、要求定義フェイズの最終結果を正しく文書化しなくてもよいということには決してならない。どのCASEツールを本当に必要とするかは、当然、システムのどの側面をプロトタイプングするかによって決まる。

もしプロトタイプがユーザー・インターフェイスに限られるなら、スクリーン・ペインタあるいはレポート・ジェネレータで十分である。しかし、データも格納しなければならないならば、データ・ディクショナリをもつDBMSが必要となる。また手法を支援する機能が常に必要とされる。第4世代環境にも分析者用ワークベンチにもスクリーン・ペインタ、レポート・ライター、ダイアログ・マネージャ、それに作図機能と開発データの一貫性を維持する機能が備わっている。一般に、第4世代環境に見られるスクリーン・ペインタ、レポート・ライター、ダイアログ・マネージャは、分析者用ワークベンチのそれよりも質的に優れている。

一方、分析者用ワークベンチは、構造化手法・技法を支援する点においてはより多くの機能をもっており、この状況は急激に変化している。つまり、第4世代言語の提供者は、プロジェクトの初期フェイズを支援するための機能を自社のツール群に加えつつあるからであり、また分析者用ワークベンチの提供者は（簡単な）DBMSと第4世代言語を含むプロトタイプ作成のためのツールを新たに加えつつあるからである。さらに分析者用ワークベンチとCOBOLジェネレータといったツール間のインターフェイスもできつつある。

### 6.3 システムの性格

プロトタイピングは適切なツールなしでは不可能である。このことは、要求定義にこのアプローチを適用できるシステムの数著しく限られることを意味する。というのも、現時点で購入できるCASEツールの大部分は、データ指向のユーザー・インターフェイスに力点を置く強力なオンラインの性格をもつアプリケーションのプロトタイピングに特に向いているからである。力点が、データそれ自体でなくデータの処理、つまりアルゴリズムに置かれる場合、プロトタイピング工程を支援する現在の言語およびツールは一般に要求を満たしていない。複雑な操作を非手続きの方法で正確に記述することは普通はできない。処理の結果は処理自体の言葉でしか表わせないことがしばしばある。言い換えれば、結果に到達するには、実行しなければならないステップを正確に示すことによるのみ記述できるのである。プロトタイピングを検討する場合、データの収集とその保守の機能（挿入・修正・削除）は、開発すべきシステムの主要な構成要素でなくてはならない。

現在の第4世代開発環境には、ユーザー・インターフェイスを迅速に定義し、実現するツールが見受けられる。開発環境の中には、データベースの構造が定義された後、データベースからのデータを抽出したり保守したりするのに必要な画面を生成するものがある。この簡単なユーザー・インターフェイスで初期のプロトタイプの一部を作ることができ、そして後の繰返しの中で画面定義機能の助けを借りて迅速に変更を加えることもできる。

プロトタイピングがなぜユーザーとシステム間のインターフェイスを重要視するシステムに特に向いているのか、その最大の理由は、多分実用モデルがプロトタイピングの作業期間コミュニケーションの方法として役に立つからであろう。システム開発者とユーザーはシステムの外部表示について議論し、それによって外部レベルだけでなく概念レベルに対しても理解を深めることになる。システムとユーザー間のインターフェイスが重要な構成要素ではない情報システムにとって、このインターフェイスを議論の特に重要な伝達手段として使用する要求定義は適切なアプローチではない。

プロトタイピングの適用は、開発するシステムのタイプに非常に関係すると文献の中で述べられることがある。一般に用いられているシステムの類型学は組織内で認識されている決定階層（図6.1

ー 2) を基にしている。以下にそのカテゴリを示す。

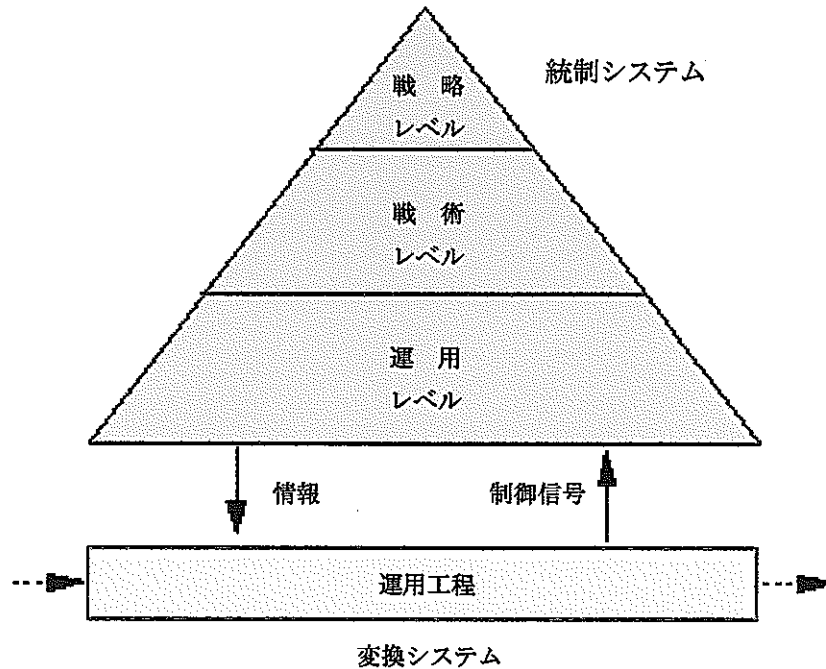


図 6. 1-2 組織内のシステム決定階層

- ・運用工程（変換工程とも呼ばれる）を支援する、あるいは自動化するシステム
- ・運用工程を制御するのに必要な運用管理を支援するシステム
- ・組織の戦術層で運用工程計画化を支援するシステム
- ・戦略的経営を支援するシステム

変換工程で必要な情報は安定しており、定型化されており、その範囲は決まっている。上位層になると、決定の範囲（組織の決定の結果）は広がり、決定者は組織外からの情報（経済発展、競争相手などについて）に左右されるようになる。戦略層になると、要求は組織の下部での事態を一目で把握できる高度に圧縮され選び抜かれたデータとなる。

戦略・戦術・運用管理間の違いは、このモデルのように明確でないことが多いが、これによって組織の情報インフラを理解しやすく管理しやすい部分に分割するのに役に立つ。しかし、プロトタイピングを持定の状況で使用するかどうかを判断するために、この分類が有用であるかは疑問である。どの組織のどの決定階層においても、情報システムが満たさねばならない要求は明確ではない。決定ピラミッドの底辺では、情報問題は構造化される傾向にある。

つまり、システム要求にまつわる不確実性は、ユーザー・インターフェイスの周囲に集まることになる。ピラミッドが高くなると、概念レベルで、特に決定者がシステム内にもちたいと思う機能について確実性が失われて行く。組織のすべての階層においては、たぶん戦術、戦略層のほうが多いと思われるが、要求と要望については不確実であり、そこでプロトタイピングは重要な役割を演じることができる。

技術情報システムでは、リアルタイムの側面がしばしば重要で、処理をモデル化することがデータ構造をモデル化するより、はるかに重要である。組込み式ソフトウェアの場合さらに問題があり、いったんこのソフトウェアが特定のマシン上で開発されると、別のハードウェアに移植せねばならぬ必要性が出てくる。“技術的”なカテゴリに含まれるソフトウェアのもう1つ別の例として、線形プログラミングのような数学的なソフトウェアがある。技術的な情報システムをプロトタイピングするためのツールは現在ほとんどない。したがって、プロトタイピングのアプローチは、このようなシステムには現在適していないといわざるをえない。技術情報システムのユーザー・インターフェイスはある程度無視されることがよくある。ユーザーとマシン間のインターフェイスが、技術情報システムの重要な構成要素であれば、この分野についてより深く検討するため、このインターフェイスをプロトタイプすることは役立つことになる。

#### 6. 4 システムの規模

プロトタイピングは、大規模で複雑なシステム開発には役に立たないという誤解が広く流布している。この誤解の主な原因は、多分プロトタイピングに皆がいただくイメージが型にはまったもので、このイメージとは、システム開発者とユーザーが画面の前に一緒に座り完全なシステムをすばやく作成するというものである。この解釈によると、どんな簡単なアプリケーションでもプロトタイピングに従うことはできなくなる。しかしながら、本項で唱えるように大規模なアプリケーションがプロトタイピング・アプローチに適さないという理由はない。その主な特徴はよく知られた構造化手法と技法を繰り返し行なうことと、ヒューリスティックなプロトタイピングとの工程と組み合わせるところにある。この方法が、問題の分析とその解決策の構造化が大いに注目を浴びているのである。

大規模アプリケーションのプロトタイピングが考慮に値すると思われるもう1つの要因は、第4世代開発環境であれば大規模なアプリケーションですら迅速に実現できるということである。実際、現在あるツールを使えば、25から35の画面をもち40から60のオンライン・プログラムから成る完全に文書化された初期のプロトタイプを4週から6週間で作成することは可能なのである。以上に述べたように、情報システムを開発する場合、同じ方法で全システムを開発する必要はない。大規模システムは一般に非常に性質の異なるユーザーとサブシステムから成っている。このような場合、プロトタイピングはこのアプローチに適した（この章で述べた基準に従い）システムのその部分に対して適用することになる。

#### 6. 5 システム開発者とシステムの利用者

プロトタイピングを用いてシステム要求定義を行なうことができるのは、以下のようなユーザーがそのモデルを評価できる場合だけである。

- ・全ユーザーを代表しているとみなされるユーザー
- ・システム要求定義に関して決定権をもっているユーザー
- ・要求定義について賢明な判断ができる能力をもっているユーザー

プロトタイピングは、要求定義工程でユーザーの中心的役割に非常に大きな力点を置く。多分、プロトタイピングが成功するための最も重要な前提条件は、ユーザーが自分に与えられた役目を喜んで

果たし、またその能力も備えていることである。

ユーザーは自分自身だけでなく、自分の仲間にも受け入れられることができる人間環境工学的に健全なユーザー・インターフェイスに到達できる能力がなくてはならない。そのシステムが機能的な観点から完全であり、またシステムの導入後もユーザー・インターフェイスに何ら修正を必要としないことを確認するのはユーザーの責任である。ユーザーは誰よりも問題の分野をよく知っているが、どのユーザーもシステム要求について賢明な判断が行なえる能力をもっているわけではない。

結局のところ、ユーザーは専門の分析者でも設計者でもないのである。プロトタイピングの進行を正しい線に沿って導いて行くには、システム開発者の役目について相当の専門的能力が求められる。開発中の情報システムを今後使うことになるユーザーが、そのシステムが備えるべき機能や形態について様々な相反する意見をもっているとすれば、プロトタイピング・アプローチに従うのは意味がない。プロトタイピングは構造的な意見の相違に対して万能策とはなりえない。しかし、もし観点の相違がその性格上構造的なものでなく、システムの機能と外見の不確実性によるものならば、プロトタイプの開発はやはり考慮すべきである。このプロトタイプはユーザー間での議論のベースとして役立つし、ユーザーはこの方法で共通の立場に立つよう努力すべきである。

将来のシステムを使用するユーザー代表者が見つからないことについては、いくつかの状況が考えられる。誰がそのシステムの今後のユーザーか正確にわかっていない、あるいは非常に大多数のユーザーが関係する場合はこの状況が起こりうる。たとえば、アプリケーション・パッケージの開発の場合である。プロトタイピングはこの工程にかかわるユーザーの時間と注意力を大いに必要とする。ユーザーはこのための準備をしなければならない。ユーザーはモデルを評価し、再調査するために進んで時間を割り当てねばならず、管理職側はそのことに時間を割けるようにしなければならない。誰もそのプロトタイプの評価に手が掛けられないとなると、プロトタイピングの工程は軋んで停止ということになる。アプリケーション・システム開発者は、情報システムの導入に反対するユーザーがプロトタイピング工程の使い方を誤り、重大な遅れを引き起こすことに常に注意しなければならない。この徴候としては要求や要望が常に変わり、重要なときにユーザーは時間が取れないといったことが挙げられる。このような場合、繰返しの工程が実りある結果を得るまでに長い時間がかかることになる。そこでシステム開発者は反対の原因を追跡し分析し、その改善策模索に努めるほうがよいということになる。プロトタイピングは、この工程で積極的な役割を積極的に引き受け、しかもその能力がある意識の高いユーザーを必要とする。

プロトタイピングは情報分析者に多くのことを求める。分析者は、もちろん立派な分析能力、構造化分析と技法についての知識、正しくコミュニケーションできる能力、そしてなるべくならアプリケーション分野の専門知識をもつべきである。情報分析者の経験的な技能に加え、プロトタイピングには現在使えるソフトウェア・ツールで何ができ、何ができないかについて堅固な知識が必要である。プロトタイプ作成者は現在使えるものの中から最適のソフトウェア・ツールを選び出し、それを使って完全に構造化されたプロトタイプをすばやく作り上げることができなければならない。

プロトタイプ作成者はまた、技術的な設計フェイズで持ち上がる問題、たとえば本番システムの一部となるべきプロトタイプの部分を伝統的な製造ラインに従い並行して開発されたサブシステムと、あるいはすでに運用されているものと統合しなければならないといったことを承知しておかねばならない。特に重要なことは、プロトタイプでもモデル化したものが本番システムで実際に実現できるこ

とを保証するため、目標とする環境のどのような制約についてもプロトタイプ開発者は承知しておかねばならない。

したがって、伝統的な工程では、いくつかの機能分野（情報分析者、技術的なシステム設計者、プログラマ）に分かれていた技能をプロトタイピングは混ぜ合わせて1人の人に求めることになる。しかし、もしそのような人がいなければ、情報分析者と専門の設計者からなるプロトタイピングのチームを導入することを考えることになるかもしれない。プロトタイピングでは、ユーザーに関して肯定的な態度がシステム開発者に求められる。過去、ユーザーは開発工程における邪魔な付属物とみなされたのは稀なことではなかった。設計と実現工程が複雑であるということは、情報問題を明らかにすることにではなく、開発の後の工程に力を注ぐことを意味していた。

ユーザーは自分が何を望んでいるのかを表現できないし、まして未知知識ことを前提にシステム開発者は問題の本質について、またその解決について自分自身の仮説を立てていたが、この状況は変わりつつある。システム部門は、他の部門へサービスを提供すべきだということが広く受け入れられるようになってきた。ユーザーの重要な役割は認識されているというのも、ユーザーだけが本当の情報はどこに存在するかを知っており、また最終的にそのシステムとともに働くことになるからである。この認識は、ユーザーの参加が基本であるアプリケーション・システム開発環境に対する戦略となる。つまり、これこそプロトタイピングの主要な目的である。これらの戦略に沿って仕事をうまく進めるためには、システム開発者は“ユーザー指向”でなくてはならない。開発者は、これらのアプローチを特徴付ける非常に密度の濃いユーザーとの会話に肯定的な態度を取るべきであり、またユーザーがかなりの範囲まで開発工程を管理できるように準備をしておかなくてはならない。プロトタイピングを用いる場合、プロトタイプ作成者は、このアプローチの特徴であるシステム要求が常に変わることを肯定的に受け止めねばならないし、さらにそれらの変更を勧めることも行わなければならない。プロトタイプを繰り返し変更するには、システム開発者の高い意識付けが必要である (Podolsky, 1977)。

現在、情報システム開発者には開発戦略、構造化手法・技法および自由に使えるソフトウェア・ツールが用意されているであろう。新たな情報問題が起こると、このコレクションの中から賢明な選択をすることができる。今日、情報システム開発に携っている専門家は様々な戦略と手法に存在する基本的な相違を詳しく知っていることが期待されている。専門のシステム開発者は、自分が自由に使えるツールを多く保持していなくてはならず、またどんな問題にも状況に応じた正しいツールを選び、使うことができなくてはならない。

プロジェクトの最初にプロジェクト・リーダーはこの章で述べた基準を基に、要求定義フェイズに従来の方法かプロトタイピングのアプローチかどちらを採用するか決めなければならない。プロトタイプを本番システムのベースとして残すことができれば、経済的な優位性はほとんど常にプロトタイピング・アプローチとなる。もしプロトタイプを捨てるのであれば、プロトタイピング・アプローチは、いまから開発する情報システムに対してユーザーの要求が不確実である場合にのみすべきである。この不確実性は以下の事柄によって決まる。

- ・支援を必要とする仕事から要求を演繹できるレベル
- ・ユーザーおよび分析者の知識と経験のレベル
- ・コミュニケーション問題の危険度

プロトタイピングは多くの無形の利益をもたらすが、この無形の利益は伝統的方法かプロトタイピング・アプローチかどちらかを選択する場合考慮すべきである。この中で最も重要なことは、プロトタイピングが開発工程と情報システム部門の両方の観点からユーザーの役割により積極的な姿勢をもたらし、そして情報システムに対するユーザーの満足度を高めることである。

CASEツールは開発工程の効率を高め、プロトタイプには要求や要望の変更に対し迅速で柔軟な対応を可能にしている。CASEツールの最大の見返りは、プロトタイピング・アプローチがこれらのツールを使うことによって得られるということである。したがって、プロトタイピングとCASEツールは連携しながら進むことになる。

現在使用できる第4世代開発支援には制約があることから、プロトタイピングは、データ指向のユーザー・インターフェイスを非常に重視するオンラインの性格を強くもったアプリケーションに特に適している。データとデータ構造の代わりに、データが従わねばならぬ処理に力点を置くのであれば、適当なプロトタイピングのツールはまだ広く使えるようにはなっていない。プロトタイピング・アプローチは現在経営情報システムには十分応えることができるが、ツールの欠如のため、技術的なアプリケーションにはそれほど、あるいはまったく応えることができない。情報システムは組織上の意志決定階層のためであるが、その階層を基にした選択基準は、要求についての不確実さが組織のあらゆる階層に散在するためほとんど価値はない。

プロトタイピングに参加するユーザーは、ユーザー社会の代表者であるべきで、システム要求定義についての決定ができ、また積極的にその決定をしなければならない。システム開発者の側では、専門の“プロトタイプ作成者”、つまり従来のシステム開発では何人かの専門家が分担したような技能を合わせもつ“ユーザー指向”の人が工程を管理しなければならない。

プロトタイピングを導入し利用しようとする、様々な問題が持ち上がる。プロトタイピングの目的を十分知らされていないユーザーは、自分達のニーズを十分満たしていない最初のプロトタイプに懐疑的な反応を示す。つまり、プロトタイプを使うのは非常に難しく、繰返し作業を迅速に行なうことなどできはしない。また長い骨の折れる開発工程を経た後では、本番のシステムが待てなくなり、プロトタイプを手放そうとはしなくなるといったことである。プロトタイピングを導入し利用する間に起こる問題はほとんど、ユーザーおよび開発担当者双方がその概念について十分な知識をもっておらず、一貫性のあるフェイズごとのアプローチについていけないことに起因する。

## 6.6 システム開発に潜在的な問題

### (1) 開発担当者間における開発要求の誤解

開発担当者がイメージしているプロトタイピングは、彼らが慣れ親しんでいるアプローチと非常に違っているということである。彼らが積み重ねてきた知識や経験が、要求定義への新しいアプローチの到来によって突然価値がなくなってしまうことを懸念している。プロトタイピングは彼らの身分と地位に対する脅威と見えるのである。経験ある開発担当者がプロトタイピングを拒絶することになることにもなる。彼らはこのアプローチに非常に懐疑的になり、それを妨害しようとすらす。そこで、変化に対して当然起こる抵抗は克服されねばならない。

プロトタイピングの導入によって起きる自動化社会への抵抗は、概念を正しく理解していないことによることが多い。教育と訓練により、この問題に対し機先を制することができる。プロトタイピン



グは革命的な新しい開発方法ではなく、むしろ現在の要求定義方法の改良であることを明らかにしなければならない。構造化手法と技法で培われた技能は、プロトタイピングの出現によって捨て去られることはない。逆に、このアプローチとうまく組み合わせられ、本当のプロトタイピングの基本的な部分となる。プロトタイピングは、情報分析者が自由に使えるひとそろいの戦略、手法、技法に付け加えられるべきもので、それらの代わりになるものではない。このことは、もちろんプロトタイピングが何も変えはしないといっているのではない。

プロトタイピングを適用しようとするれば、開発担当者は新しい技能とツール（たとえば第4世代ツール）を修得しなければならない。さらに開発工程と、そこでユーザーが果たす役割の両方についての考えを変えなければならない。開発担当者がプロトタイピングで成功を収めたいと熱望するならば、その変化に備えなければならない。しかし、（プロトタイピングに関する多くの出版物からすると）変化はまず最初に予想するほど大きくはない。したがって、教育と訓練が十分練られていれば、多くの抵抗の源を克服することができる。

### （2）管理者間における開発要求の誤解

プロジェクト予算に責任をもつ管理者は、プロトタイピングとは何度もシステムを構築しては改訂し、あるいは置き換え、最後には時間とお金の立派な無駄使いだということで捨ててしまう、といった方法（ただの単純化された実用モデルであるが）とみなすかもしれない。逆に費用の膨張効果だけを見る管理者とは対照的に、プロトタイピングにより開発工程の能率が相当向上し、したがって費用も低減できると固く信じている管理者もいる。プロトタイピングが純粋に経済的な観点から正当化できるかどうかは、主としてプロトタイプ（の一部）が本番システムのベースとして使える見込みがあるかないか、またシステムへの要求にどれだけ不確実性が存在するかその程度にもよる。さらに、プロトタイピングは目に見えないが様々な現実的な利益をもたらす、これらも考慮に入れるべきである。プロトタイピングが効率を上げることに結び付くか否かについては疑問がある。プロトタイピングの主要な目的は開発プロジェクトの有効性を高めることである。つまり、多くの人が認識できる効率の向上は、実際には第4世代言語の使用によるものである。担当の管理者は、確固たる根拠の基にこのアプローチの採用を決定するため、プロトタイピングそれ自体の確かな費用と利益について理解しておくべきである。

### （3）ユーザー間における開発要求の誤解

プロトタイピングへの誤った期待がユーザーの間にあるが、それは大部分プロトタイピングの目的と効果について十分な知識が欠如していることに起因している。この不十分な知識のためプロトタイピングを選んだり、拒んだり、またこのアプローチに失望したりすることになる。適切な手引きがない限り、経験のないユーザーは“最終的な”プロトタイプを運用可能なシステムとして期待しがちである。ユーザーはなぜ本番用のシステムが使えるまでに、かなりの時間待たねばならないか多分はつきりと理解できない。ユーザーは既に最終的なプロトタイプに満足しており、またあらゆる面からそれは完成しているとみなしてしまうことがよくある。このことは開発担当者にそのプロジェクトを終わらせ、他のシステムに目を向けさせる圧力になる。これは実際、しばしば起こる問題である。早まってプロトタイプ・システムをユーザーに渡すと、望ましくない結果になる。品質改良フェイズを省

略すると、本番システムを成功裏に導入し、使用するのに基本となる多くの要因が無視されることになる。以下にその要因を挙げる。

- ・性能
- ・例外および誤り状況とその処理方法
- ・バックアップと回復
- ・セキュリティ
- ・信頼性
- ・保守性
- ・テスト
- ・ユーザーおよび技術支援のための文書
- ・運用コスト

これらの面に適切な考慮を払わずに本番システムとしてプロトタイプを使うと、ユーザーは当初満足するがしばらくすると不満をいだくようになることが非常によく起こる。その失敗は、普通このプロジェクトで使われるアプローチに原因がある。プロジェクトの最初の時点で、プロトタイプの限界をユーザーに示すことはプロトタイプ作成者の責任である。プロトタイピングの作業が完了した後、特にプロトタイプを捨てる場合、本番のシステムが使えるまでにはやはり長い時間を要することを明確とさせておかねばならない。まずプロトタイプをシステムの要求定義とみなし、システム自体であるとみなすべきではない。この観点からの判断を基に同意した根拠にしたことを、第2編で述べるプロジェクト計画に記録しておかねばならない。

プロトタイピングに失望するもう1つの原因は、その手順と目的が十分理解されていないことにある。プロトタイピングの目的は、開発担当者が実用モデルを使ってユーザーの要求と要望を深く理解できるようにすることにあることをユーザーに明らかにしておかなければならない。もし明らかにしておかないと、ユーザーは初期のプロトタイプと期待しているシステムとがほとんど一致していないことから、好ましくない態度をとるかもしれない。この工程にユーザーが参加しようとするれば、ユーザーはプロトタイピングの間システムのある選別された側面、あるいはサブシステムだけに重点を置き、そして工程の一部としてプロトタイプを“あれこれと検討”するであろう。むしろ、必ずしなければならないことを認識する必要がある。プロトタイプは迅速に容易に変更できることから、経験のないユーザーが本番システムも同じ早さで変更できるという誤った結論を引き出すことがある。これはプロトタイプと運用システムとの間の相違を明確にしなければならないもう1つの理由である。

本番システムがプロトタイプから格上げされたものであっても、第4世代言語で書かれていたとしても、一般的に本番システムはプロトタイプより何倍も柔軟性に劣ることになる。ユーザーはプロトタイピング工程で重要な役割を果たし、作業を行ない、そして参加するには、労力あるいは時間が必要であることを十分認識しておかねばならない。プロトタイピングは、ユーザーに積極的に参加しようという意識があり、システムの要求を決定する機会も能力もあり、さらにプロトタイプを評価するのに十分な時間がある場合にのみ価値がある。さらにユーザーの管理者はプロトタイピングへの参加により、従業員の通常の作業がかなりの間中断されることを認めなければならない。

この落とし穴は、当事者を正しく教育し訓練することによって避けることができる。したがって、プロトタイピングを考える場合、要求定義に対するこのアプローチについてできるだけ早い時期に知ら

せることは賢明なことで、これによって関係者にはすべて長所短所が十分伝わることになる。

#### (4) 不十分な要求分析

プロトタイプを不注意に使うと、本来の情報問題を明らかにできない。まず情報問題を迅速に解決することをプロトタイプでは強調しすぎるために、プロトタイプ作成者の誤解を招き、問題を十分理解する前にプロトタイプ作成に熱中してしまうことになる。ユーザーが自分の要求しているものを既に熟知していれば、このようなプロトタイプはすぐに捨て去り、すばやく最終的なプロトタイプに集中できることから、必ずしもこれは困難ではない。

しかし、ユーザーが問題の性格について、またその解決策がどんな形になるかについて確信がなく、さらにシステム開発者の影響を受けるような場合は、また別の問題である。問題解決の作業中、人はえてして受けた提案の解決策に安住しようとする。この事象は定着化といわれることがある。当事者がすでに自分が望んでいるものに対し明確な考えをもっていないならば、根本的に異なる解決策が提案されることもない。

最初のプロトタイプが本当の問題を解決するには不十分であると、最終的に完成したシステムが結局間違った問題を解くことになるか、あるいは満足できる解決に達するため何度も必要以上に繰返し作業を行なうことになるか、そのいずれかとなる。このことを図6. 1-3に示した。

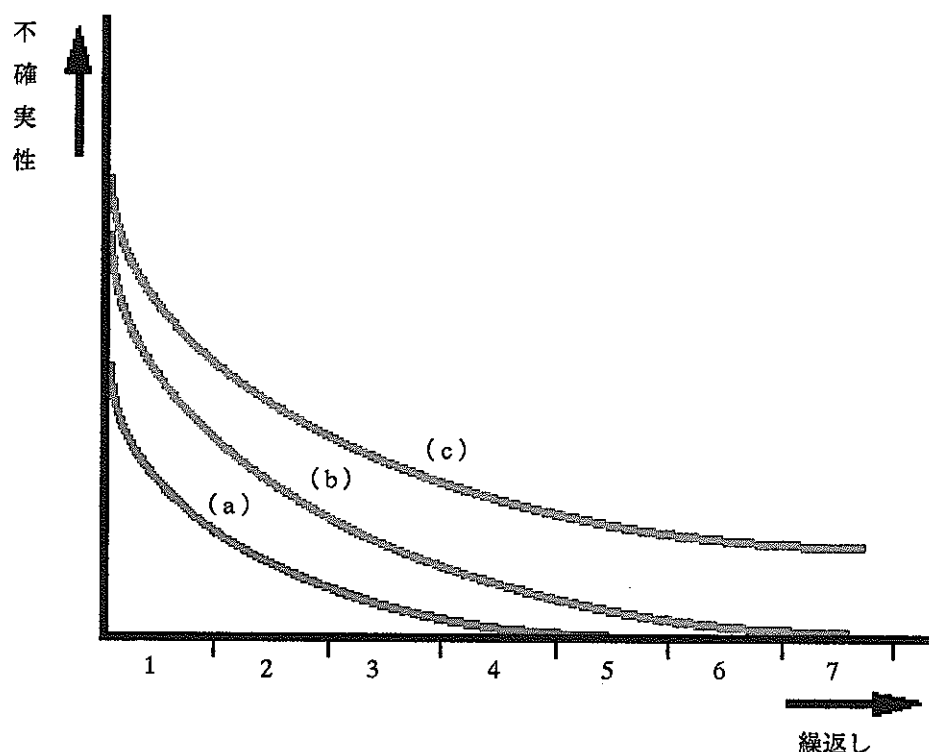


図6. 1-3 不十分な問題分析のリスク

(a) の場合、十分な問題分析が最初のプロトタイプを作る前になされており、最初の繰返しの作業中に問題の不確実性は急激に低減している。プロトタイプ作業がさらに進むと、繰返しの度に行なう変更の数は少なくなる。プロトタイプ作業は、プロトタイプにもはや著しい変化がなくなった時点で終りとなるが、そ

れにはユーザーの本当の要求がその時点で明確になったとの前提が必要になる。この前提は (b) の場合にも当てはまる。しかし、問題の分析が不十分なため、(b) に現われる不確実性はプロトタイプ開発時に相当あり、その結果繰返し作業の回数は必要以上に多くなっている。(c) の場合、本当の問題が現実にまだ明らかになっていない時点で、もうプロトタイプは根本的な変化を受け付けなくなっている。この場合、プロトタイプは正しい完全な要求定義に結び付くことにはならない。

細部に深入りするプロトタイプは、開発中のシステムの細部にまで注意を払うことができるため、システム全体の見通しを見失うことがある。ユーザーは、その時点で理解できるプロトタイプの部分についてあら捜しをしがちであり、その実大規模システムに関する限り、ユーザーはシステム内のそのような部分については全体として批評することはほとんどない。

このことは、プロトタイプを評価期間に完全に評価するにはあまりにもプロトタイプの範囲が広すぎたり、またプロトタイプ作成者が明確な構造（たとえば、階層的に構成されたメニューを使って）を利用しなかった場合に当てはまる。この問題は“おまけ”をシステムに欠くことのできないものと見てしまうユーザーの性癖によるものであり、その結果、繰返しの工程は長くなり、本番システムは不必要に高価になり、主要な問題から注意が逸れてしまう危険性が出てくる。

この原因は、プロトタイプ作成者がすばやく、また安易に新規の変更要求・希望をシステムに取り入れてしまうことにある。このような状況を排除するには、開発担当者が適宜な平衡感覚をもつ必要がある。新たな要求や希望を見境いなく受け入れてはいけない。あらゆる折にふれ、ユーザーと開発担当者はともに新たな機能が有用か、またそれらの機能を実現するため必要となる作業を正当化できるかどうかを更に検討を加えなければならない。細部にわたる配慮と全体に対する配慮との間に均衡のとれた解決策を見出すのはプロトタイプ作成者の責務である。インターフェイスへの配慮が不十分である開発中のシステムと他のシステムとのインターフェイスを最初おまかに分析する間に十分な注意を払わないと、統合フェイズで回避できない問題に当面する。その後再び、様々なサブシステム間を流れる情報を分析する必要がある。プロトタイプはシステムに関するユーザーの要求と要望について洞察を深めることにある。もし入力が現在の情報システムに向けられたものであるならば、あるいは出力がそこから来るものであるならば、どのような情報が作られるべきか、または受け入れられるべきかはすでにわかっていることである。

プロトタイプは役割はそこではない。つまり、インターフェイスの分析によってのみ、統合の問題は回避できる。プロトタイプは情報システムとユーザーとのインターフェイスに対してのみ適用でき、他のシステムとのインターフェイスには適用できない。問題分析と問題解決が混同されている従来の情報システム開発方法は、問題指向活動と解決指向活動を厳密に区別することに断固たる力点を置いている。問題指向の活動は、最終的にモデルの中で見つけ出した問題を解決するため情報システムは何をし何を備えていなければならないかを示すことになる。つまり、ユーザーの問題とニーズに焦点が集められる。解決指向の活動は、ユーザーの様々な要求や要望を満たすシステム（自動化されていようがいまいが）をどのように実現するかその方法を確立することにある。解決策を実行するには、様々なソフトウェア・ツールの可能性と制約は重要になってくる。要求定義は決して現在の技術的環境（ハードウェア、オペレーティング・ソフトウェア、開発ツール、その他同種のもの）の制約によって影響をなる。

この前提は (b) の場合にも当てはまる。しかし、問題の分析が不十分なため、(b) に現われる不

確実性はプロトタイプ開発時に相当あり、その結果繰返し作業の回数は必要以上に多くなっている。

(c) の場合、本当の問題が現実はまだ明らかになっていない時点で、もうプロトタイプは根本的な変化を受け付けなくなっている。この場合、プロトタイピングは正しい完全な要求定義に結び付くことにはならない。細部に深入りしすぎるプロトタイピングは、開発中のシステムの細部にまで注意を払うことができるため、システム全体の見通しを見失うことがある。ユーザーはその時点で理解できるプロトタイプの部分について粗探しを行いがちであり、その実大規模システムに関する限り、ユーザーはシステム内のそのような部分については全体として批評することはほとんどない。

大規模なあるいは複雑なシステムの場合、従来のトップダウンの作業、すなわち工程分析手法（たとえば、構造化分析と構造化設計）を使って、かなり自主独立した管理可能なサブシステムにすることができる。その後再び、様々なサブシステム間を流れる情報を分析する必要がある。プロトタイピングの目標は、システム（の側面）に関するユーザーの要求と要望について洞察を深めることにある。もし入力が現在の情報システムに向けられたものであるなら、あるいは出力がそこから来るものであるなら、どんな情報が作られるべきか、または受け入れられるべきかは既にわかっていることである。プロトタイピングの役割はそこではない。つまり、インターフェースの分析によってのみ、統合の問題は回避できる。プロトタイピングは、情報システムとユーザーとのインターフェースに対してのみ適用でき、他のシステムとのインターフェースには適用できない。問題分析と問題解決が混同されている従来の情報システム開発方法は、問題指向活動と解決指向活動とを厳密に区別することに断固たる力点を置いている。

問題指向の活動は、最終的にモデルの中で見つけ出した問題を解決するため情報システムは何をし何を備えていなければならないかを示すことになる。つまり、ユーザーの問題とニーズに焦点が集められる。解決指向の活動は、ユーザーの様々な要求や要望を満たすシステム（自動化されていようがいまいが）をどのように実現するかその方法を確立することにある。解決策を実行するには、様々なソフトウェア・ツールの可能性と制約は重要になってくる。問題分析と問題解決とを厳密に区別する理由をいくつか次に紹介する。

- a. 要求定義は、決して現在の技術的環境（ハードウェア、オペレーティング・ソフトウェア、開発ツール、その他同種のもの）の制約によって影響を受けるべきでない。要求定義は、情報システムが問題解決のため満たさねばならない要求を厳密に表現すべきである。機能的要求のほんのいくつかだけがシステム設計に移されることになれば、問題は完全に解決されないことは論理的な帰結である。この線に沿って考えれば、ハードウェアの構成は定義された要求から直接派生すべきもので、要求定義フェイズでは重要ではない。
- b. 要求定義は、システムが実現される技術的環境に無関係であり、たとえその環境が変わったとしても相変わらず有効でなくてはならない。要求定義は、論理的にはどのハードウェアでもシステムの設計と実現化の確たる出発点となる。つまり、要求定義はいわば移植可能なのである。
- c. 問題の分析とその解決策を厳密に分ければ両方の面が十分考慮される保証となり、また問題を定義する前に解決方法を考えるといったことはなくなる。情報分析者は、要求上の問題だけに力点を置くことになる。そのことにより、結果の要求定義は設計者にとっての確固たる出発点となる。要求定義による枠組みの中で、設計者は様々な解決方法を自由に考え、最高のもの

を選ぶことができる。

- d. システムの利用者は、技術的な事柄や技術的な細部について一般に興味を示さない。実際、システムがどのように作られようが彼らにはまったく興味がないのである。彼らは設計と実現化の工程をブラックボックスとして見たいのであり、入力及要求定義であり、出力はこれらの要求を満たす本番システムなのである。

プロトタイピングは、一見したところ問題分析と問題解決とを厳密に分けるのをより難しくしている。結局、プロトタイピングは要求定義のフェイズに実用モデル、つまり問題に対する解決策を持ち出すことになる。このことは、プロトタイピングが定義と設計の厳密な分別による利点をすべて無にしてしまうのではないかと多くの人を恐れさせることになる。

この方法をより深く調べると、それは必ずしも事実とはいえないことが明らかになる。プロトタイピングの間、問題指向と解決指向の活動はかなり厳密に分けられる。もちろん、プロトタイプを実現するのに用いるツールの制限や特性が要求定義に影響を与えることは事実である。

しかし、ユーザーがプロトタイプ（とそれに関する要求定義）を受け入れることについて最終決定権をもっていることから、それは問題ではない。結局、もしユーザーが完全に、そして十分な理解の基にシステム要求定義を受け入れるならば、それがどのように達成されようが大した問題ではない。定義と設計を厳密に分けることは理想的ではあるが、この考えをあまりに強調するのは実際非現実的であることに気をつけるべきである。組織は、一般に持定のハードウェアとシステム・ソフトウェア戦略に委ねられている。この戦略に対するどんな変更も歓迎されないし、また高価につく。あらゆる機能的な要求や好みには定価札が張り付けられているのである。

要求定義段階に対する現実的なアプローチは、初期の段階で技術的な可能性とコストの両方を考慮に入れることである。コストと利益の比較により、要求が経済的に見合うものか、そしてその要求をシステム要求定義に合めるべきかどうかを明らかにしてくれる。このように、ある程度分析と設計の工程を混ぜ合わせることは好都合なことすらありうる。プロトタイピングにより、開発者は機能を実現するために必要となる作業を現実的に眺めることができ、また初期の段階で機能を実現できることを確かめることができる。

## 6.7 プロトタイピング手法の導入

従来の要求定義方法に代わるものとしてプロトタイピングを受け入れるかどうかは、プロトタイピングを導入する方法と非常に関係がある。プロトタイピングが確実に失敗する1つの方法は、十分な準備もしないで偶然うまく行きそうな何処かのプロジェクトにそれを使い始めることである。このような環境で不運な経験をした人は、このアプローチに敵意をもつことは確かである。組織にプロトタイピングを導入するには準備や計画に十分気をつけ、そしてうまく管理を行なわねばならない。

大々的に実施する前に、懐疑論者にはその有用性とプロトタイピングの価値を認めてもらわねばならない。理論上の学術論文だけでは不十分で、プロトタイピングの価値は実際のテストで成功した後、初めて受け入れられるのである。このためパイロット・プロジェクトは必須であり、開始する前に正しい初期条件を設定しておかねばならない。

## (1) プロジェクトの準備

順調に、そして成功するプロトタイピングを導入できるか否かは次の要因に左右される。

- ・ 要求定義用のプロトタイピングに対する管理者の態度
- ・ 導入に伴う教育と活性化活動
- ・ 適切なツールが使えること

パイロット・プロジェクトでこのアプローチを実際に始める前に、これらの重要な成功要因それぞれに注意を払わねばならない。どんな改革でも組織で成功を収めるには前提条件が必要である。それは非常に高い意識をもって現状を変革し推進せねばならないという管理者の積極的な態度である。

もし管理者層からの支持、公約、献身がないとなれば、プロトタイピング導入のどんな試みも暗い運命に遭遇することになる。優秀な情報システム部門が問題に直面すると、その管理者は喜んで問題を解決しようとするのが一般的である。保守作業が大変で、またアプリケーションのバックログがたまることは、現在の作業方法が適切でないことを十分に示している。何かを変えねばならないことは管理者にとって明らかである。しかしながら、この問題に取り組む最良の方法、最も有効な手段は明らかでないことが多い。

この点については、4つの面を、つまりエンドユーザー・コンピューティング、アプリケーション・パッケージ、CASEツール、プロトタイピングをどう組み合わせるかで適材すればよいかを述べておいた。責任ある管理者の支持と約束が得られるのは、彼らがプロトタイピングについて十分精通したときだけ可能となる。このアプローチについての誤解はありうる。十分丁寧にその概念と背景を説明できれば、最初の疑惑を感激へと変えることができる。さらにこのように十分説明できれば、プロトタイピングを実際に評価するための、また必然的に発生する支出についての正式な承認を得ることができる。最低、少なくとも以下の点に対処しなければならない。

- ・ アプリケーション・バックログの主な原因、構造化手法と技法の長所と短所、さらにプロトタイピングでこれらを補う方法
- ・ プロトタイピング・アプローチの広範囲な説明書
- ・ プロトタイピングの費用・便益性
- ・ プロトタイピングの前提条件
- ・ 提示された評価方法とその費用

少なくとも管理者の支持を得ると同様に大事なことは、パイロット・プロジェクトに参加しているユーザーと開発担当者を活性化し教育することである。プロトタイピングの概念について知識が不十分であると細部に深入りすることになる。細部に深入りすると開発担当者の役割への反抗とユーザーの役割への過大な期待になる。したがって、プロトタイピングの目標とこのアプローチの実施によって開発工程の各人の役割が受ける影響を認識すべきである。開発担当者やユーザーの立場にとって組織の土壌はプロトタイピングに好意的でなくてはならない。ユーザーはプロトタイピングによって求められる努力を惜しまないように意識付けされねばならない。開発担当者はこのアプローチに積極的な態度を取り、それが結局彼らの作業方法に大きな成果をもたらすのである。教育と訓練はプロトタ

イピングに対してやる気を起こさせる土壌を育てるのに重要な役割を果たす。プロトタイプ作成者になろうとする情報分析者は、CASEツールの使用にかかわる実際的な技術を習得しなければならない。DBMSとデータ・ディクショナリ・システム (DDS)、その環境で使われる非手続き言語、それに画面定義とレポート生成機能を使って迅速に実用モデルを構築できなければならない。

さらに（もし使用できれば）分析者用ワークベンチも容易に使えなくてはならない。情報システム開発に対するアプローチがプロトタイピングと従来の方法とでは違いがあることから、設定された標準、基準、手順がプロトタイピング環境では制約になってしまう。たとえば、1つの潜在的な対立の原因は開発工程におけるデータベース管理者の役割である。従来、データベース管理者は関係するデータベースを構築するための実際の行動を起こす前に、データベースの設計内容を承認していた。プロトタイピングでは、まずすばやくデータベースを構築し、その後引き続き迅速に最初のデータベースを変更することが必要であるが、この場合時間を要する形式的な手続きが入り込む隙はない。プロトタイピングでのデータベース管理者は、本番システムで実現するデータベースの設計内容を承認することだけが求められる。つまり、データベース管理者は違った方法で開発工程に携わることから、現在の手続きの改訂が必要となる。標準と手続きについて責任のある人は重大な混乱を予想しておかねばならず、またプロトタイピング・アプローチの影響を認識しておかねばならない。プロトタイピングの導入により標準の変更が必然的に起こることはかなり明らかではあるが、このことに実際十分気づいている人はいないようである。

## (2) プロトタイピング・ツール

プロトタイピングに必要なCASEツール（なるべくならRDBMS、非手続き言語、優れたスクリーン・ペインタ/レポート・ジェネレータを含む）を使えない場合は失敗することになる。このようなツールを使って仕事をしていない組織にプロトタイピングを導入することは無意味である。DBMSと統合されたDDSがあれば、優れた分析者用ワークベンチのようにプロトタイプの作成は簡単になる。このようなツールを組織に導入することと、プロトタイピングを導入することは、一般的には無関係である。

これらのツールは開発工程の効率向上を目指したものであり、従来の直線的な開発アプローチを含めいくつかの開発方法に使用されている。“ソフトウェア危機”から逃れる1つの可能性の高い方法は前述したように、より効率的な開発をすることである。しかし、CASEツールの使用によってもたらされる最大の利益は、プロトタイピングと組み合わせたときに得られる。プロトタイプ作成者がハードウェアの制約に束縛されないことは重要である。各プロトタイプ作成者には（インテリジェントな）ワークステーションが与えられ、十分な内部・外部記憶装置が用意されるべきである。応答時間は十分速くなくてはならない。オフィス環境もまた重要で、進行を邪魔されずにエンドユーザーとコミュニケーションできるように、開発担当者は十分なミーティング用の場所をもつべきであり、可能ならば大きなスクリーン・プロジェクタのような発表用設備を備えた所がよい。

## (3) パイロット・プロジェクトの実施

実質的な評価のためのスタートをうまく切れると、次の段階は適当な1つのプロジェクト（あるいは多くのプロジェクト）を選択することである。この選択を行なう場合、細心の注意を払わねばなら



ない。つまり、選んだシステムがプロトタイピングに役立たなければ、その結果は単に失敗や失望だけでなく、多分プロトタイピングは不当に拒まれることになる。プロトタイピングの選択基準として以下の項目を挙げる。

#### システムのタイプ

プロトタイピングが、ユーザー・インターフェイスを重視しなければならない情報システムの開発に特に適しているのは、現在のCASEツールの性格による。データの処理に重点を置いた場合、プロトタイプを迅速に作り上げるのに十分適した手段は現在のところない。

#### システムの規模

一般に、大規模なアプリケーションも小規模なアプリケーションもプロトタイプは可能である。しかし、開発するシステムの規模はパイロット・プロジェクトを選択するときの要因となる。アプリケーションは、このアプローチを大規模システム開発にも使えることを示すために適当な大きさでなくてはならないが、経験を積みつつあるプロトタイプ作成者が、そのプロジェクトの大きさのために問題に衝突するほどの大きさであってもいい。プロジェクトの様子が組織内の皆によくわかるようにするには（興味をもった外観者が多数いるだろう）、あまり大きなプロジェクトを選ばないようにすることである。そうすれば結果をすばやく示せるというのがもう1つの理由である。

パイロット・プロジェクトを始める前に、このプロジェクトを評価するための基準を作ることが重要である。パイロット・プロジェクトが成功に終われば、プロトタイプによって完全な要求定義も、あるいはもしプロトタイプの性能を改良できれば、本番システムも作れることを示すことができる。パイロット・プロジェクトを通して、懐疑者はこのアプローチが可能であり、実際的であり、また成果をもたらすことを納得することができる。しかしながら、客観的な基準がないため、この成果の質を測定することができない。保守費用を基にすれば、そのシステムの有効性についてある見識を得ることができるはずである。しかし、パイロットという状態であることから、長期間保守費用を測定するためにシステムを使用することは実際的にほとんど不可能である。したがって、従来のアプローチと比較して経済的な利益が生まれるかどうかについて客観的に示すことは難しい。さらに、従来のアプローチによるそのプロジェクトへの影響は（第4世代言語と組み合わせられているかどうかにかかわらず）わかっていない。したがって、パイロット・プロジェクトを評価するために使える基準は大部分主観的である。たとえば、その工程と成果物の両方に対するユーザーの満足度、要求定義にこのアプローチを用いたことに対する開発担当者の評価、プロジェクトがその過程で行なったプロジェクト管理とその結果についての印象などがある。参加する人は、すでに従来のアプローチに経験があるほうが望ましい。なぜならば、適切な比較ができるからである。十分踏み固められた道からそれたプロジェクトは、型にはまったプロジェクトより失敗の危険度が高いということを認識すべきである。もしパイロット・プロジェクトがいくつかの点で不満足な結果になろうとも、これは必ずしも使用した開発アプローチの問題ではない。

その原因は新しい作業方法に十分な経験がない、あるいはツールの使用に熟練していないことにある。パイロット・プロジェクトを開始する前に、開発者が小さなテスト問題をユーザーとのやりとりなしに全工程を通して試みる如果能够できれば失敗の危険度は低くなる。パイロット・プロジェクトでは、開発工程の他の改革に気をとられずプロトタイピングの評価にのみ集中することが重要である。

新たな構造化手法、新たなツール、それに新たな開発アプローチを同時に導入し、そしてさらに未知のアプリケーション分野に乗り出そうとすれば、失敗を避けることはほとんどできない。

#### (4) プロトタイピング・アプローチ

プロトタイピングは、従来の要求をまとめる方法に代わる完全に独立した方法として受け入れるべきであるとの勧告でパイロット・プロジェクトが終了し、管理者もその勧告に同意するならば、組織内ではプロトタイピングの基盤の拡大方法を検討しなければならない。プロトタイピングの提案の前に、アプリケーションが成功するための十分な条件を設定しておかねばならず、また組織の用途に合わせた実用的なマニュアルを作成しなければならない。標準のプロジェクト・モデルは、要求定義フェイズではプロトタイピングを、また本番システムのベースとしてはプロトタイプを使えるように変更が可能でなくてはならない。プロジェクトの管理者と開発担当者のための訓練コースの開設が必須である。最初にプロトタイピングで作業をする組織・部門は、このアプローチの経験をコンサルタントに相談できなくてはならない。小さな組織では、パイロット・プロジェクトとその成果を誰でも認識できる。

もしパイロット・プロジェクトにかかわる人がプロトタイピングに熱中すると、他の人にプロトタイピングを使うように勧める必要はほとんどない。大きな組織のパイロット・プロジェクトは、開発手法・技法およびツールの選択と標準化に責任をもった企業内組織の部門によって進められることが多い。組織の大きさや手法を定める部門の権限にもよるが、プロトタイピングが広く用いられるようになる前に、相当渉外活動を行なわねばならない。すなわち、プロトタイピングの正しい提案をしなければならない。そうすることで、本当の組織に受け入れられるための不可欠な広い支持が得られることになる。プロトタイピングが、情報システム開発の分野における重要な進展であることは疑いない。他のエンジニアリングの分野では、すでに一般に通用している開発システムの実用モデルを導入することを通して、プロトタイピングは開発工程をさらに専門化することになる。プロトタイピングを用いると、自動化はユーザーの目に“より親しみのある顔”となり、そこでの要求定義戦略はユーザーと情報システム専門家との大きな隔りを埋めるのに役立つ。

##### ・開発費用に対するプロトタイピングの効果。

プロトタイピングにより開発費用はかきみ、プロジェクトの進行が遅れることが予想される。この仮定は、主としてプロトタイピングを従来の手法・技法の代替としてではなく、それを補うものとして見なければならない事実を基にしている。プロトタイプが本番システムのベースとして維持されるとしても、工程は繰り返し行なわれることから、要求定義にプロトタイピングを使うプロジェクトは多分従来の開発より費用を要す。もしこの仮定が確たる数字で裏付けされれば、それは役に立つことになる。この分野で測定を行なうのは難しい。最大の悩みの種は、ソフトウェア開発費用が非常に広い様々な要因に依存していることである。開発工程に及ぼすプロトタイピングの影響について断言するためには、非常にしっかりと管理された“実験室”の条件のもとで調査を行なうか、あるいは膨大な観察を行なうかのいずれかとなる。

##### ・保守費用に対するプロトタイピングの効果。

プロトタイピングが、従来のアプローチより質の優れた要求定義に結び付くと期待できるかどうかはまだにはっきりとしない。これも確たる数字に欠けるのである。残念ながら、要求定義の質を計るための優れた基準はない。しかしながら、保守費用の大きさとシステム要求定義の品質との間には逆の関係があることから、ある洞

察を行なうことは可能である。プロトタイピングは主に保守フェイズで顕著な効果を上げることができる。しかしながら、保守費用に対するプロトタイピングの効果を測定する場合、非常に多くの費用決定要因があるという問題に再び直面する。プロトタイピングの効果を取出すには、数多くの観察が必要となる。この種の調査は費用が掛かり非常に長期間を要する。事実、測定はシステムが“期限切れで捨てられる”まで続けなければならない。したがって、プロトタイピングが保守費用にどんな影響を及ぼすかについて正確に測定しようとするれば、異常なほどの忍耐が要求される。実験室での調査は、かなり長期間にわたって実際にシステムを使うことが興味の対象となることから、観察に代わるものとはほとんど考えられない。十分な期間にわたって興味ある要因をすべて固定しておくことはほとんど不可能である。

#### ・開発工程と完成したシステムの両方に対するユーザーの態度にプロトタイピングが与える影響

このことについてはAlaviの調査を引用したこれは数少ない調査の1つで、ここではプロトタイピングの期待効果についてある経験的基礎資料を得るための努力がなされている。この調査によれば、プロトタイピングでより一層ユーザーと親和性のあるシステムには必ずしもならないようである。このような驚くべき結論には将来の研究が求められる。

上に述べた点について、より優れた経験的基礎数字を得ることは学術的な世界にとっても、現場の社会にとっても1つの挑戦である。実際、プロトタイピングの費用対便益効果を経験的に調査するには、費用を要し時間的な浪費となってしまう。様々な研究機関と産業界とで情報を交換し協力し合うことは、この問題についての内容を理解するのに不可欠である。プロトタイピングと接点をもつ多くのことについていままで簡単に述べてきた。これらのうちいくつかについて、現在の文献は十分な注意を払っていない。短期的に行なわれるべき調査活動は以下の通りである。

#### ・プロジェクト管理と開発作業の両方のため、開発戦略、手法、技法、自動化ツールの選択を行なうための暫定的アプローチの作成と適用。

プロトタイピングと従来のアプローチとを選択する基準に注目してきたが、これはより広い観点から見なければならない。つまり、プロジェクト・リーダーは他の選択もできるのである。実際、以下の選択ができるよう広範囲な枠組みが緊急に求められている。

- a. エンドユーザー・コンピューティングによる開発か、情報システム専門家による開発か
- b. 標準パッケージを購入するか、システムを社内で開発するか
- c. 従うべき開発アプローチ（参加型、進化型、プロトタイピング型、直線型、プロセス指向型、データ指向型など）
- d. それぞれのアプローチで使うべき構造化手法・技法、CASEツール

#### ・戦略、手法、技法、ソフトウェア・ツールの生産性に対する効果の評価。

現在新しい開発ツールを組織に導入する場合の企業の判断には、ほとんどビジネス上の適切な理解に欠けている。プロトタイピングは単なる要求定義戦略ではない。また十分な定量的データがそろっているわけでもない。現場では、費用対便益の観点から妥当であるか否か前もって分析せずに、その時点での流行に従うこともありえる。文献はほとんど手掛りを与えてくれない。生産性についての記述は一般に定性的であり、定量的なことが述べられる場合は、ほとんど実践的な証拠がそろっていることはない。様々な開発戦略、手法・技法、（第4世代あるいはその他の）ツールの生産性に対する効果についての調査が緊急に求められている。

#### ・新たな戦略、手法・技法、およびツール導入にまつわる問題点の評価。

実際（たとえば）構造化手法あるいは第4世代ツールの導入は少しも簡単ではないことがよくあるようである。

マネジメント科学の分野において、最近の数十年間多くの注意が組織変更のプロセス管理に払われてきた。この知識をシステム開発戦略その他に導入可能性の調査は、まだ組織的にはなされていないようである。新しいCASEツールの出版物や提供者は、この問題にもっと注目すべきである。

## 6. 8 試作システムの仕様

地層処分バーチャル・エンジニアリング・システム開発用のハードウェアは、大別して下記の5種類に機能分割した。本項では、試作したシステムの仕様を一覧する。

機能分野1：シミュレーション用システム (Visual Environment Simulation System)

機能分野2：操作システム (Operator Console System)

機能分野3：データベース用システム (Data Base System)

機能分野4：パラメータ・ディスプレイ用システム (Status Display System)

機能分野5：オーディオ・ビデオ用システム (Audio Video System)

### 6. 8. 1 全体システム概念構成

図6. 8-1に試作した地層処分バーチャル・エンジニアリング・プロトタイプ・システムの概念構成を示す。

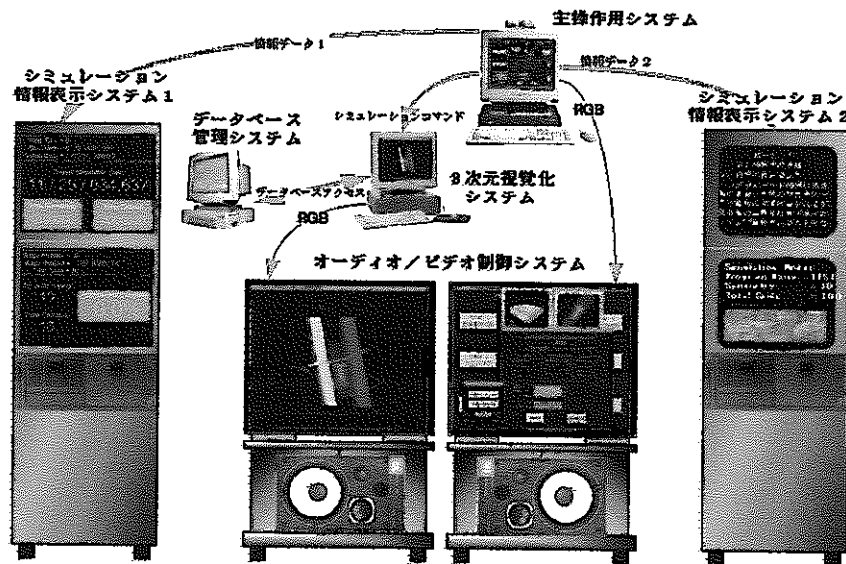


図6. 8-1 システム概念構成図

### 6. 8. 2 多重並列分散動作機能概念

図6. 8-2に試作したネットワーク並列処理機能の概念構成を示す。

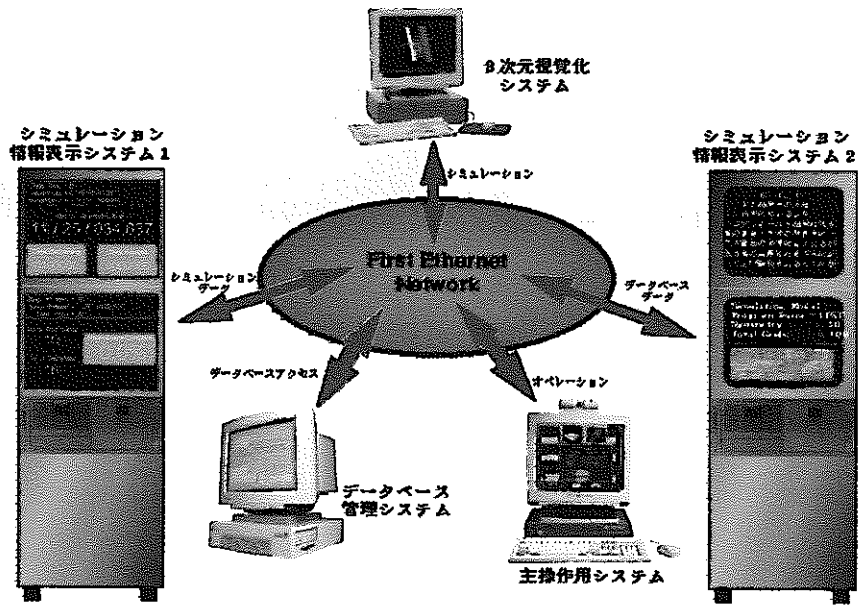


図6. 8-2 ネットワーク並列処理機能概念

### 6. 8. 3 ビデオ信号制御仕様

図6. 8-3に試作したビデオ信号処理系を示す。

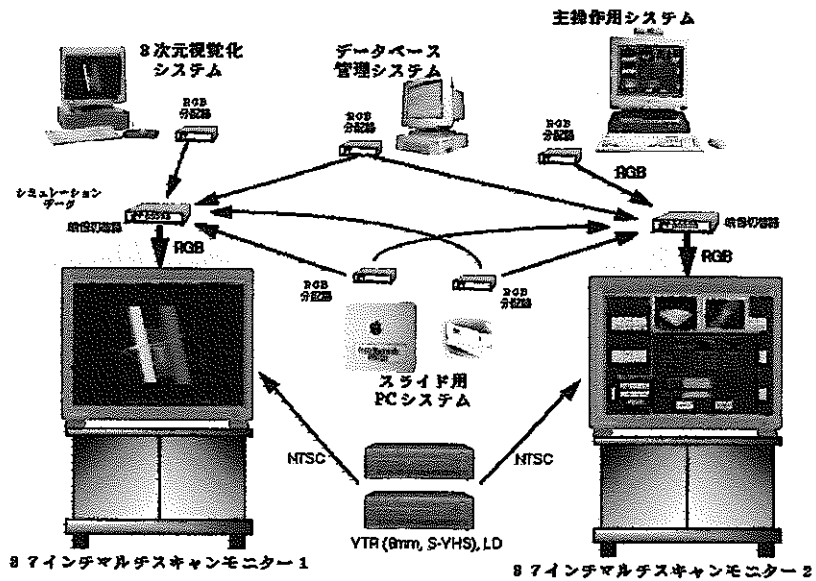


図6. 8-3 ビデオ信号処理系

#### 6. 8. 4 オーディオ信号制御仕様

図6. 8-4に試作したオーディオ信号処理系を示す。

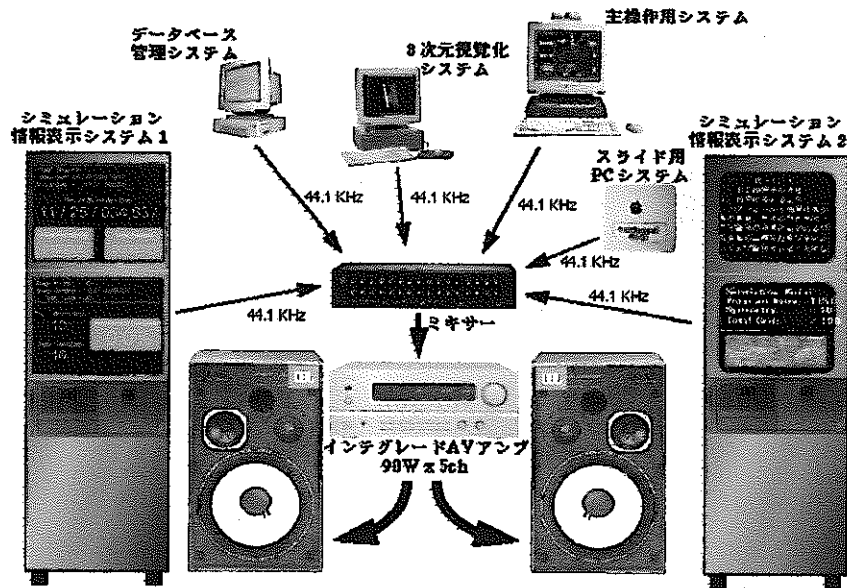


図6. 8-4 オーディオ信号処理系

#### 6. 9 シミュレーション・システム

シミュレーション用システムは、地層処分性能評価にかかわる数値計算を行うとともに、3次元グラフィックス機能を利用し、ビジュアルなシミュレーション結果の視覚再現を行うシステムとして構成した。システム・ハードウェアはシリコン・グラフィックス社製 Indigo2を利用した。



図6. 9-1 シミュレーション・システムの外観

(1) 処理装置仕様 (Processor Specification)

用途 : 解析コードの実行等を処理する為に用いる  
選定論拠: 3Dグラフィクス機能を内蔵していること  
型式 : SGI Indigo2 Extreme  
仕様 : MIPS R4400/150MHz  
数量 : 1

(2) 主記憶装置仕様 (Main Memory Specification)

用途 : シミュレーションの実行等をメモリー上に常駐させる為に用いる  
選定論拠: 96MB以上のメモリーを内蔵していること  
型式 : メモリ (Memory)  
仕様 : 96MB  
数量 : 1 空スロット無し

(3) 磁気ディスク装置仕様 (Hardware Disk Specification)

用途 : システムOSを格納する為に用いる  
選定論拠: 2GB以上の容量があること  
型式 : System Disk (From Above)  
仕様 : 2BG Disk (1.03GB + 1.03GB Internal Disk)  
数量 : 2

(4) 追加ディスク装置仕様 (Additional Hardware Disk Specification)

用途 : シミュレーション・ソフトウェアを格納する為に用いる  
選定論拠: 8GB以上の容量がある事  
型式 : Work Disk  
仕様 : 8.6 GB External Disk (4.3GB + 4.3GB)  
数量 : 2

(5) モニター仕様 (Monitor Specification)

用途 : シミュレーション結果を確認する為に用いる  
選定論拠: 1280x1024の解像度をサポートする事  
型式 : Stand Alone 20" Monitor (From Above)  
仕様 : 20"  
数量 : 1

(6) 基本ソフトウェア仕様 (System Software Specification)

型式 : IRIX Operating System (Include Current System)  
仕様 : IRIX6.2

数量 : 1  
用途 : OSとして使用する  
選定論拠 : AT & T UNIX System V + Berkeley BSD準拠で、64Bit, 32bitをサポートする事

型式 : Fortran Compiler (Include Current System)  
仕様 : Fortran 77 Compiler for IRIX6.2  
数量 : 1  
用途 : Fortranプログラムの開発、保守に用いる。  
選定論拠 : 解析コードとの整合性をとる事

型式 : Open Inventor (Include Current System)  
仕様 : Open Inventor 3D Toolkit  
数量 : 1  
用途 : 3Dグラフィクスオブジェクトの開発に用いる  
選定論拠 : SGIマシン上で3Dグラフィクスオブジェクトを扱える事

型式 : ImageVision Library (Include Current System)  
仕様 : ImageVision Library kit for IRIX6.2  
数量 : 1  
用途 : 画像処理アプリケーションに用いる  
選定論拠 : SGIマシン上でグラフィクスデータを扱える事

型式 : ONC/3NFS (Include Current System)  
仕様 : Network File System for IRIX 6.2  
数量 : 1  
用途 : ネットワーク間リソース共有に用いる  
選定論拠 : SGIマシンをネットワークを扱える事

## 6.10 操作システム

操作システムは、シミュレーションにシナリオの組立、シミュレーションの実行、パラメータ・ディスプレイの設定等、本システムの全ての操作を行う対話システムとする。システム・ハードウェアはシリコン・グラフィックス社製 Indyを利用した。





図 6. 1 0 - 1 操作システムの外観

(1) 処理装置仕様 (Processor Specification)

用途 : シミュレーション等の操作を行う為に用いる  
選定論拠: シミュレーション用システムとの整合性を計る事  
型式 : SGI indy  
仕様 : MIPS R4000/100MHz  
数量 : 1

(2) 主記憶装置仕様 (Main Memory Specification)

用途 : 操作画面の実行等をメモリー上に常駐させる為に用いる  
選定論拠: 64MB以上のメモリーを内蔵していること  
型式 : メモリ (Memory)  
仕様 : 64MB  
数量 : 1

(3) 磁気ディスク装置仕様 (Hardware Disk Specification)

用途 : システムOSを格納する為に用いる  
選定論拠 : 2GB以上の容量がある事  
型式 : System Disk (From Above)  
仕様 : 2BG Disk (1.03GB + 1.03GB Internal Disk)  
数量 : 2

(4) モニター仕様 (Monitor Specification)

用途 : 操作、または結果結果を確認する為に用いる  
選定論拠 : 1280 x 1024の解像度をサポートする事  
型式 : Stand Alone 17" Monitor (From Above)  
仕様 : 17"  
数量 : 1

(5) 基本ソフトウェア仕様 (System Software Specification)

型式 : IRIX Operating System  
仕様 : IRIX6.2  
数量 : 1  
用途 : OSとして使用する  
選定論拠 : AT & T UNIX System V + Berkeley BSD準拠で、64Bit, 32bitをサポートする事

型式 : NFS (Include Current System)  
仕様 : Network File System for IRIX 6.2  
数量 : 1  
用途 : ネットワーク間リソース共有に用いる  
選定論拠 : SGIマシンをネットワークを扱える事

型式 : Builder Xcessory  
仕様 : Builder Xcessory for IRIX  
数量 : 1  
用途 : 操作GUIの開発、保守に用いる  
選定論拠 : AT & T X11 R4/R5をサポートする事

## 6. 1 1 データベース・システム

データベース用システムは、地層処分性能評価シミュレーションに必要な、評価シナリオ、廃棄物の工学情報、解析モデル・データ等、地層処分ヴァーチャル・エンジニアリング・システムの稼働に必要な情報を管理するシステムとした。システム・ハードウェアは、サン・マイクロ社製SUN SPARC 10により構成した。



図6. 1 1 - 1 データベース・システムの外観

### (1) 処理装置仕様 (Processor Specification)

用途 : データベースツール実行等を処理する為に用いる  
選定論拠 : AT & T UNIX System V + Berkeley BSDをサポートしている事  
型式 : SUN SPARC 10  
仕様 : Model 10  
数量 : 1

### (2) 主記憶装置仕様 (Main Memory Specification)

用途 : データベースツールの実行等をメモリー上に常駐させる為に用いる  
選定論拠 : 3 2 MB以上のメモリーを内蔵していること  
型式 : メモリ (Memory)  
仕様 : 32MB  
数量 : 1

(3) 磁気ディスク装置仕様 (Hardware Disk Specification)

用途 : 基本OSとデータベースツールで使用するデータを格納する為に用いる  
選定論拠: 10GB以上の容量がある事  
型式 : System Disk + External Disk  
仕様 : 2.1GB(From Above) + 9.0GB  
数量 : 1

(4) モニター仕様 (Monitor Specification)

用途 : データベースツールを構築操作する為に用いる  
選定論拠: 1280 x 1024の解像度をサポートする事  
型式 : Stand Alone 20" Monitor (From Above)  
仕様 : 20"inch Color Monitor  
数量 : 1

(5) 基本ソフトウェア仕様 (System Software Specification)

型式 : SUN-OS Version Solaris 2.1  
仕様 : Solaris 2.1  
数量 : 1  
用途 : OSとして使用する  
選定論拠: AT & T UNIX System V + Berkeley BSDをサポートする事

型式 : C Compiler  
仕様 : GNU (Free Soft-ware)  
数量 : 1  
用途 : C言語プログラムの開発、保守に用いる  
選定論拠: データベースツールを構築出来る事

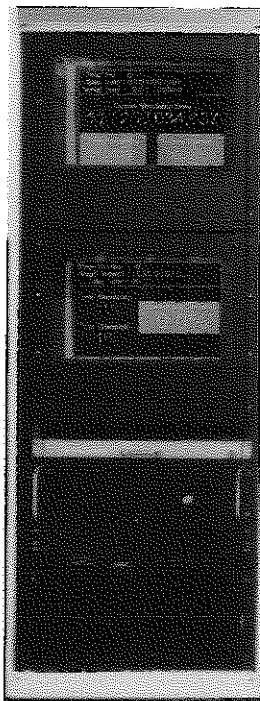
型式 : C++ Compiler  
仕様 : GNU (Free Soft-ware)  
数量 : 1  
用途 : C++言語プログラムの開発、保守に用いる  
選定論拠: データベースツールを構築出来る事

型式 : Data-base Application  
仕様 : Postgres 95 ( Free Soft-ware)  
数量 : 1  
用途 : データベースツールとして用いる  
選定論拠: 配列型のリレーショナルデータベースとする事

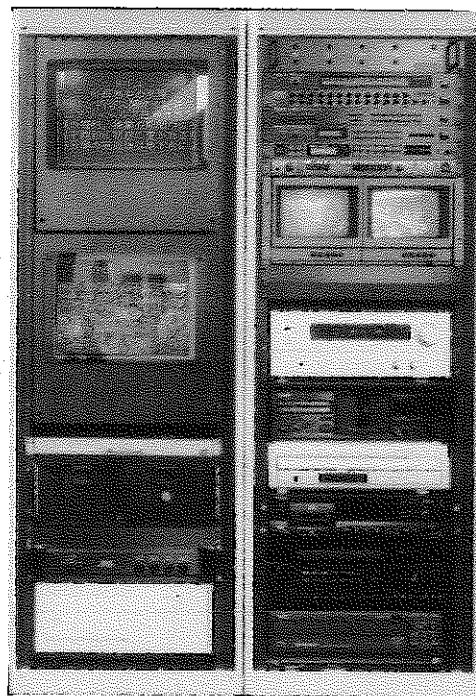
## 6. 1 2 パラメータ表示システム

パラメータ表示システムは、地層処分ヴァーチャル・エンジニアリング・システムのシミュレーションにかかわるパラメータを表示するシステムとした。

パラメータとは例えば、現在シミュレーション中の時刻、選択されたシナリオ、選択された廃棄体工学情報等とした。パラメータ・ディスプレイ用システムは、シミュレーション・システム等ネットワークで接続された、他のシステムと並列にメッセージ通信を行いながら多重処理動作を行う機能が要求されたため、市販の製品としては適当なものがなかったことから専用のハードウェアとして基盤等を調達して製作した。



パラメータ表示システム (左)



パラメータ表示システム (右)

図 6. 1 2 - 1 パラメータ表示システムの外観

### (1) 処理装置仕様 (Processor Specification)

用途 : シミュレーション中の状態情報を実行する為に用いる

選定論拠: シミュレーション用、操作用システムと整合性がとれる事

型式 (Type) : GTA PCZ3

仕様 (Spec.) : Pentium 166MHz/32MB Memory

数量 (Quantity) : 2

### (2) 主記憶装置仕様 (Main Memory Specification)

用途 : 状態情報の実行等をメモリー上に常駐する為に用いる

選定論拠 : 32MB以上のメモリーを内蔵していること

型式 (Type) : Memory

仕様 (Spec, ) : 32MB

数量 (Quantity) : 2

### (3) 磁気ディスク装置仕様 (Hardware Disk Specification)

用途 : 基本システムと状態情報アプリケーションを格納すす為に用いる

選定論拠 : 2GB以上の容量がある事

型式 (Type) : 2GB Disk

仕様 (Spec, ) : 2GB Disk

数量 (Quantity) : 2

### (4) モニター仕様 (Monitor Specification)

用途 : 状態情報を表示する為に用いる

選定論拠 : 19インチラックに取付けられ、且つ1280x1024の解像度をサポートする事

型式 (Type) : Rack Mount 17inch Color Display

仕様 (Spec, ) : 17inch

数量 (Quantity) : 4

### (5) CD-ROM仕様 (CD-ROM Specification)

用途 : システム保守を行う為に用いる

選定論拠 : ISO9660準拠とし8倍速以上の読み込み速度をサポートする事

型式 (Type) : 8 Times CD-ROM Drive

仕様 (Spec, ) : 4x CD-ROM Drive

数量 (Quantity) : 2

### (6) イーサネット仕様 (Ether-Net Specification)

用途 : シミュレーション用、操作用システムとのメッセージ交換を行う為に用いる

選定論拠 : シミュレーション用、操作用システムとの整合性がとれる事

型式 (Type) : Ether-Net Card

仕様 (Spec, ) : First EtherLink PCI 10MB

数量 (Quantity) : 2

### (7) グラフィック仕様 (Graphics Specification)

用途 : 状態情報を1枚のグラフィックカードにより2画面に表示する為に用いる

選定論拠 : 2画面以上の表示を可能とし且つ2560x2048以上の解像度をサポートする事

型式 : Graphics Card

仕様 : Graphics Card  
数量 : 2

(8) サウンド仕様 (Sound Specification)

用途 : 状態情報の変動、変化を音によって表現する為に用いる  
選定論拠: 44.1KHz以下のサンプリング・レートをサポートする事  
型式 : Sound Card  
仕様 : Sound Card  
数量 : 2

(9) 基本ソフトウェア仕様 (System Software Specification)

型式 : Windows 95 Japanese Version  
仕様 : Windows 95 Japanese Version  
数量 : 2  
用途 : OSとして使用する  
選定論拠: UNIX Systemとの整合性がある事  
型式 : Microsoft Visual C++  
仕様 : Microsoft Visual C++ Japanese Version  
数量 : 1  
用途 : C++言語プログラムの開発、保守に用いる  
選定論拠: 状態情報アプリケーションを開発できる事

### 6. 1 3 オーディオ・ビデオ・システム

オーディオ・ビデオ用システムは、ネットワークにて接続統合された、各個別システムの出力する、オーディオ・ビデオ信号を一元化する装置として構成した。これにより、個別のコンピュータシステムより出されるモニター信号、ビープ信号等を一元化して表示管理することができた。オーディオ・ビデオ用システムは、シミュレーション用システム、操作用システム、データベース用システム等のモニター信号を表示すると共にナレーション、効果音等を再生することが可能とした。

#### (1) 大型カラー・ディスプレイ・モニター



図 6. 1 3 - 1 モニターの外観 1

用途 : シミュレーション操作、結果を表示する為に用いた  
選定論拠 : 37インチ以上のブラウン管で1280 x 1024の解像度をサポートする事  
型式 : 37inch Multiscan Monitor  
仕様 : MITSUBISHI 37inch Multiscan Monitor (XC-3730C)  
数量 : 2

#### (2) その他 (The other requirement)

型式 : LD-player  
仕様 : LD-player  
数量 : 2  
用途 : パラメータシステム用の音源等として用いる  
選定論拠 : 映像、音響データを再生できる事

型式 : 9inch Monitor  
仕様 : SONY 9inch Color Monitor  
数量 : 2  
用途 : ビデオ映像制御装置からの信号を確認する為に用いる



選定論拠：NTSC信号をサポートする事

型式 : Swiching Hab

仕様 : Swiching Hab

数量 : 1

用途 : 各システム間とのデータ交換をスムーズに行う為に用いる

選定論拠：スイッチドネットワークをサポートしている事

型式 : 19'inch Rack form GTA PCZ3

仕様 : 19'inch Rack

数量 : 2

用途 : パラメータシステムを19インチラック内へ収納する為に用いる

選定論拠：EIA規格19インチラックとする事

型式 : CD-ROM Writer

仕様 : PIONEER (DW-S114X-J + CDM-V74)

数量 : 1

用途 : 成果物をCD-ROMにて配付する為に用いる

選定論拠：ISO9660をサポートし書込、読み込みができる事

## 付録一A

### 試作動作シミュレーション画面集

\*利用したデータは、動画化プログラムの動作を確認することを目的として作成したものであり、個別の解析結果ではない。平成8年度の研究では、視覚化を重視し解析コードとのインターフェイスは次年度の課題とした。

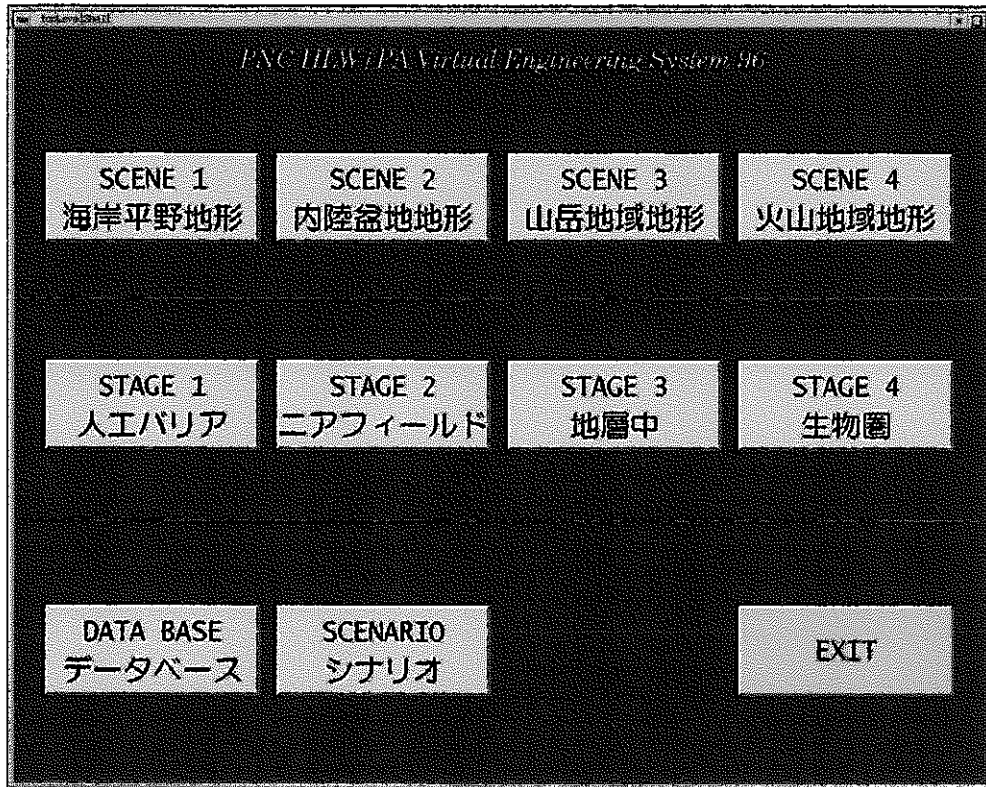


図1 主操作画面1

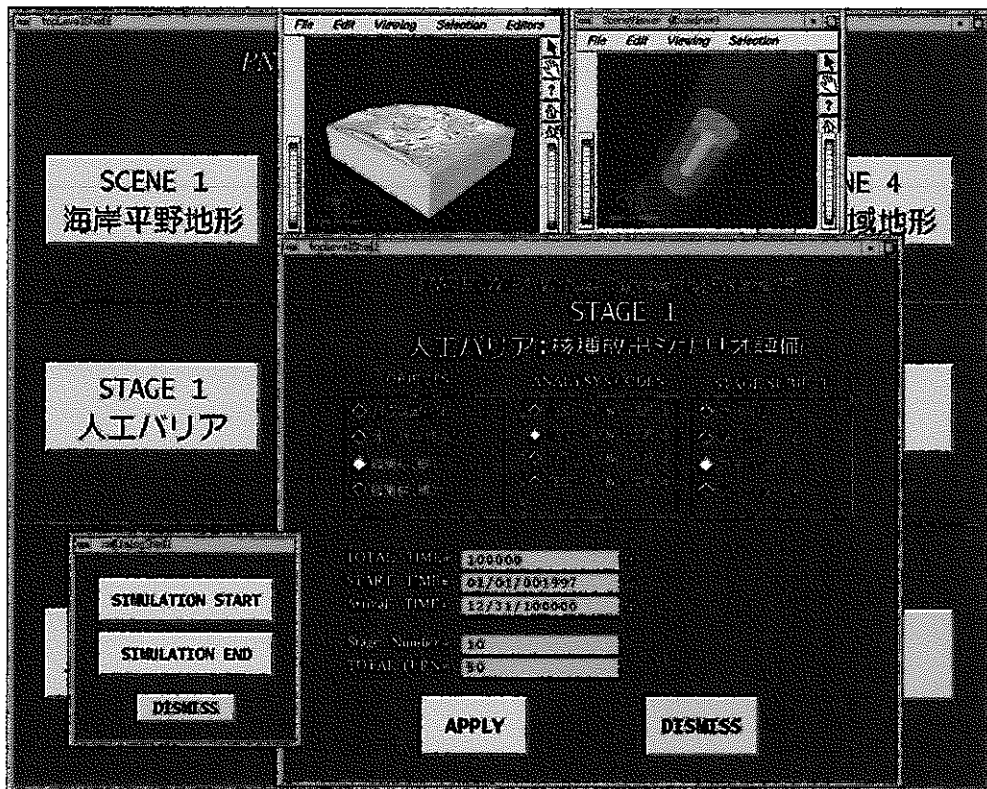


図2 主操作画面2

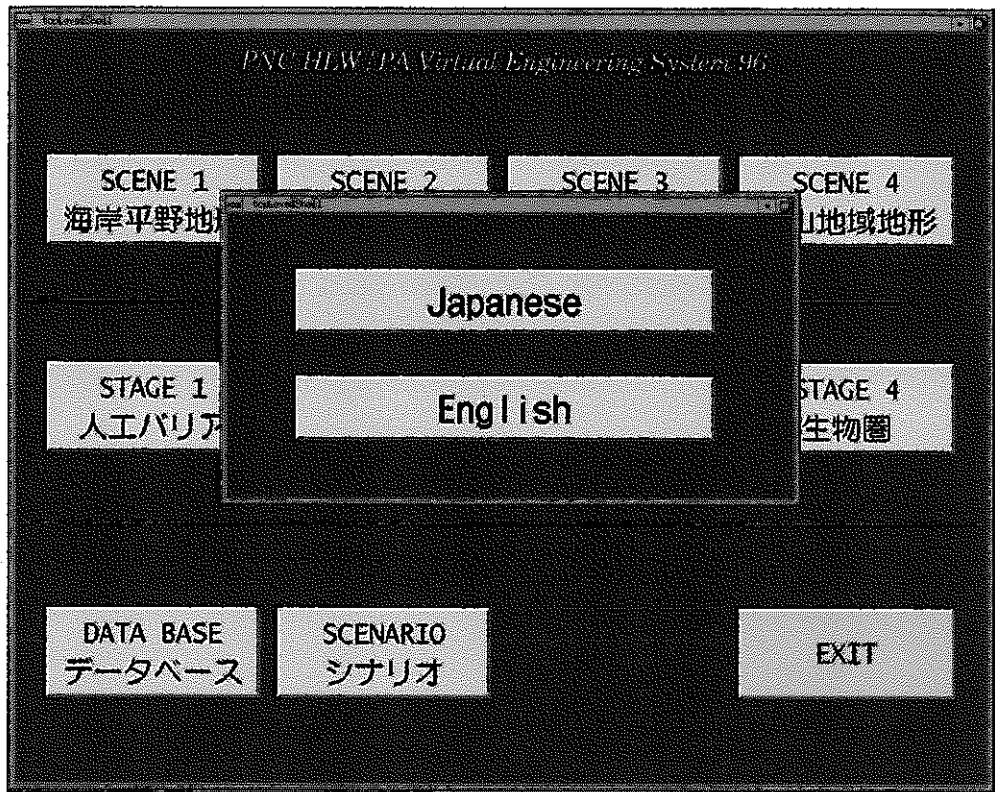


図 3 FEP操作画面 1

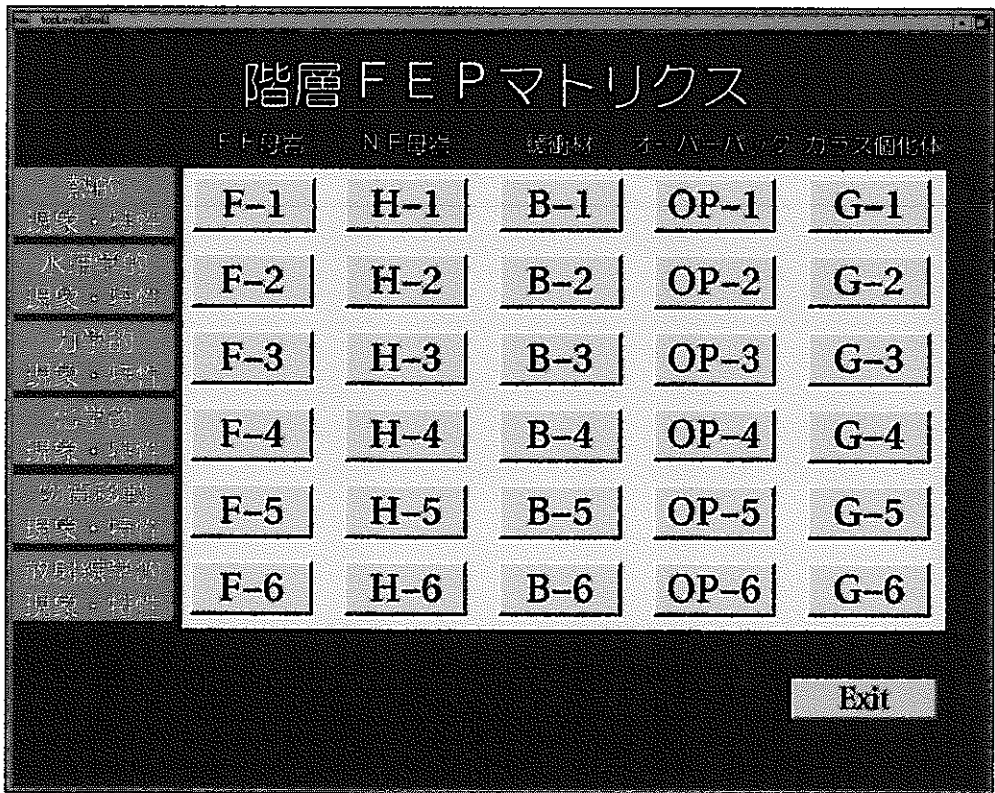


図 4 FEP操作画面 2

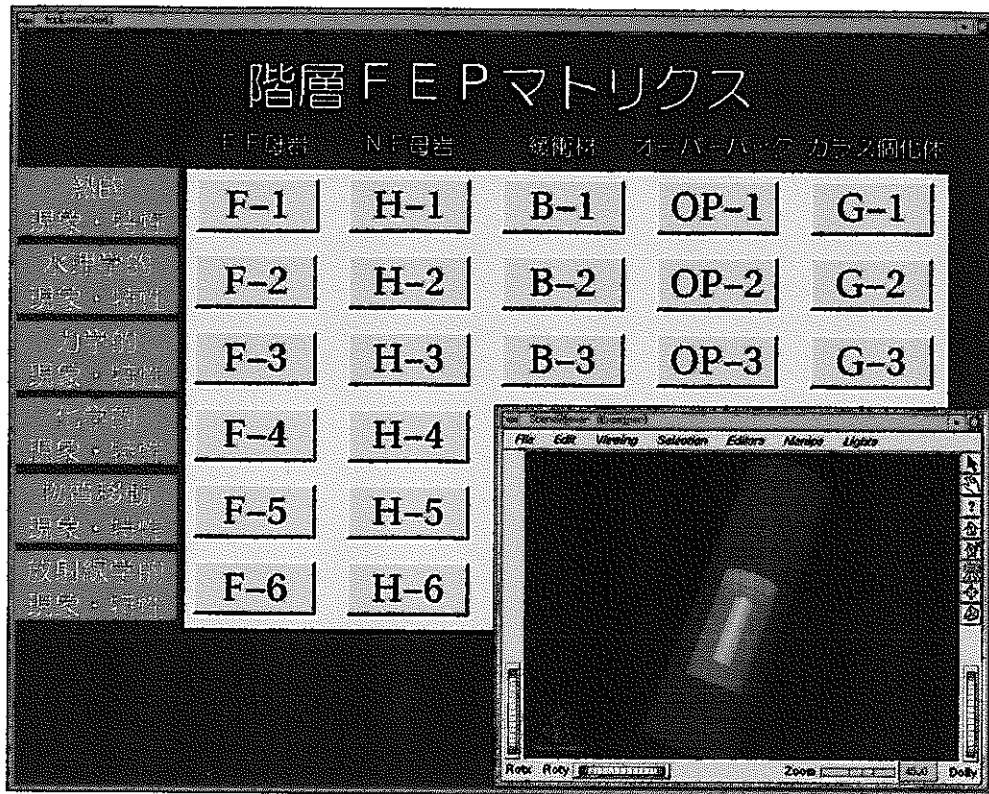


図5 FEP操作画面3



図6 FEP操作画面4

## 階層 FEP マトリクス

F 母岩    N F 母岩    補填材    オーバーハング    ガラス固化体

物理的現象・特性	F-1	H-1	B-1	OP-1	G-1
水理学的現象・特性	F-2	H-2	B-2	OP-2	G-2
力学現象・特性	F-3	H-3	B-3	OP-3	G-3
化学的現象・特性	F-4	H-4	B-4	OP-4	G-4
物質移動現象・特性	F-5	H-5			
放射線学的現象・特性	F-6	H-6			

CONCENTRATION IN BACKFILLING MATERIAL (G/M<sup>3</sup>)

DISTANCE	TIME AFTER LEACHING (Y)							
	M	100	200	300	400	500	600	700
5.200E-01	3.770E-03	3.770E-03	3.770E-03	3.770E-03	3.770E-03	3.770E-03	3.770E-03	3.770E-03
1.440E-01	4.030E-04	1.010E-04	3.370E-04	1.020E-03	1.100E-03	1.210E-03	1.270E-03	1.320E-03
5.050E-01	1.240E-05	4.340E-05	8.310E-05	1.230E-04	1.600E-04	1.940E-04	2.210E-04	2.410E-04
6.420E-01	2.530E-07	1.070E-06	2.270E-06	1.020E-05	1.620E-05	2.220E-05	2.670E-05	2.970E-05
6.910E-01	4.820E-09	6.110E-08	2.540E-07	6.610E-07	1.250E-06	2.150E-06	3.220E-06	4.270E-06
7.400E-01	1.220E-11	1.600E-09	1.030E-08	5.240E-08	1.530E-07	4.050E-07	8.620E-07	1.460E-06
7.890E-01	3.790E-13	3.620E-11	3.400E-10	1.370E-09	4.750E-09	1.120E-08	2.240E-08	3.620E-08
8.380E-01	3.660E-15	7.030E-13	1.020E-11	6.040E-11	2.310E-10	6.470E-10	1.430E-09	2.430E-09
8.870E-01	4.930E-17	1.200E-14	2.310E-13	2.080E-12	9.820E-12	3.300E-11	8.850E-11	1.510E-10
9.360E-01	4.040E-19	1.860E-16	5.970E-15	6.370E-14	3.750E-13	1.510E-12	4.710E-12	8.110E-12
9.850E-01	3.030E-21	2.500E-18	1.230E-16	1.760E-15	1.290E-14	6.260E-14	2.270E-13	4.270E-13
1.030E+00	2.160E-23	3.330E-20	2.300E-18	4.470E-17	4.090E-16	2.370E-15	1.030E-14	3.620E-14
1.080E+00	1.630E-25	3.920E-22	4.130E-20	1.040E-18	1.190E-17	1.050E-16	4.620E-16	8.620E-16
1.130E+00	9.350E-28	4.470E-24	6.790E-22	2.250E-20	3.210E-19	2.650E-18	1.300E-17	4.340E-17
1.180E+00	5.780E-30	4.710E-26	1.030E-23	4.370E-22	8.060E-21	8.020E-20	3.340E-19	1.030E-18
1.230E+00	3.450E-32	4.750E-28	1.490E-25	4.480E-24	1.000E-22	2.260E-21	1.750E-20	5.110E-20
1.280E+00	2.000E-34	4.480E-30	2.030E-27	1.940E-25	4.210E-24	5.990E-23	3.410E-22	1.030E-21
1.330E+00	1.120E-36	4.050E-32	2.620E-29	2.490E-27	3.030E-25	1.490E-24	1.570E-23	4.340E-23
1.380E+00	6.190E-39	3.520E-34	3.250E-31	4.170E-29	1.700E-27	3.350E-25	4.340E-24	1.030E-23
1.430E+00	3.330E-41	2.950E-36	3.700E-33	6.300E-31	3.310E-29	6.010E-27	1.180E-25	3.620E-24
1.480E+00	1.740E-43	2.380E-38	4.220E-35	9.100E-33	5.030E-31	1.630E-29	2.770E-27	6.420E-26
1.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00

図 7 FEP操作画面 5

## H-4 N F 母岩の化学的現象・特性

H-4.1 N F 母岩中の地下水化学
H-4.2 N F 母岩の鉱物組成・特性
H-4.3 N F 母岩中のガス（空気）
H-4.4 N F 母岩中の微生物の活動
H-4.5 N F 母岩中の有機物
H-4.6 天然コロイドの形成

閉じる

図 8 FEP操作画面 6

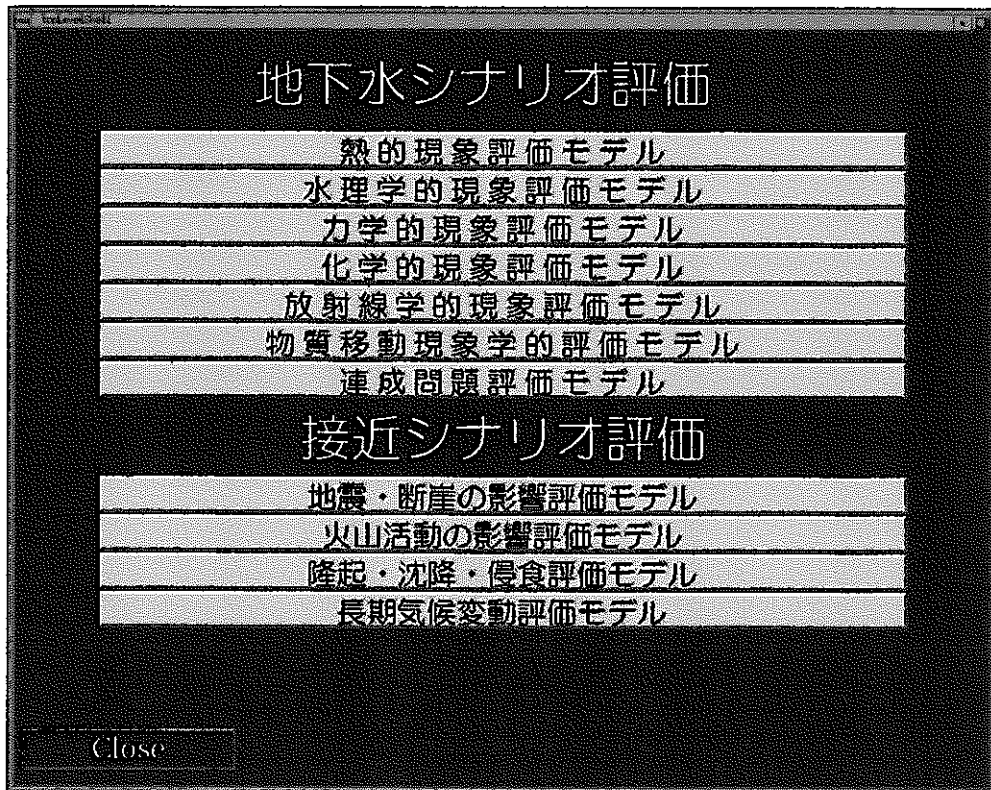


図9 データベース操作画面1



図10 データベース操作画面2

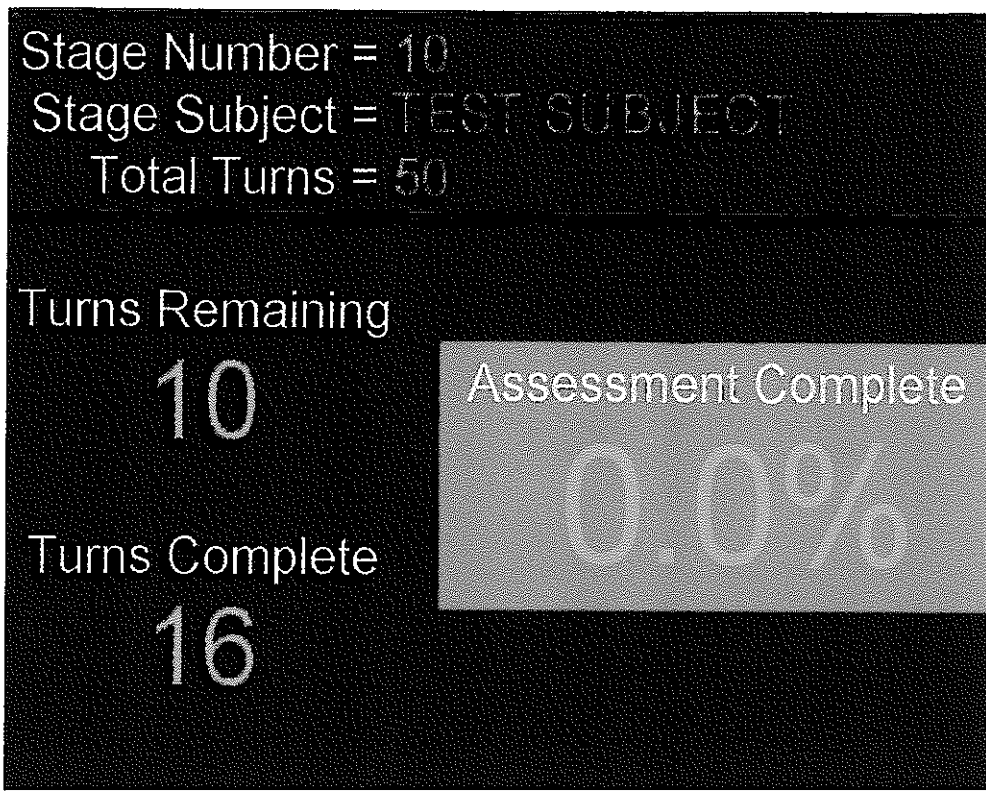


図 1 1 シミュレーション・パラメータ表示画面 1

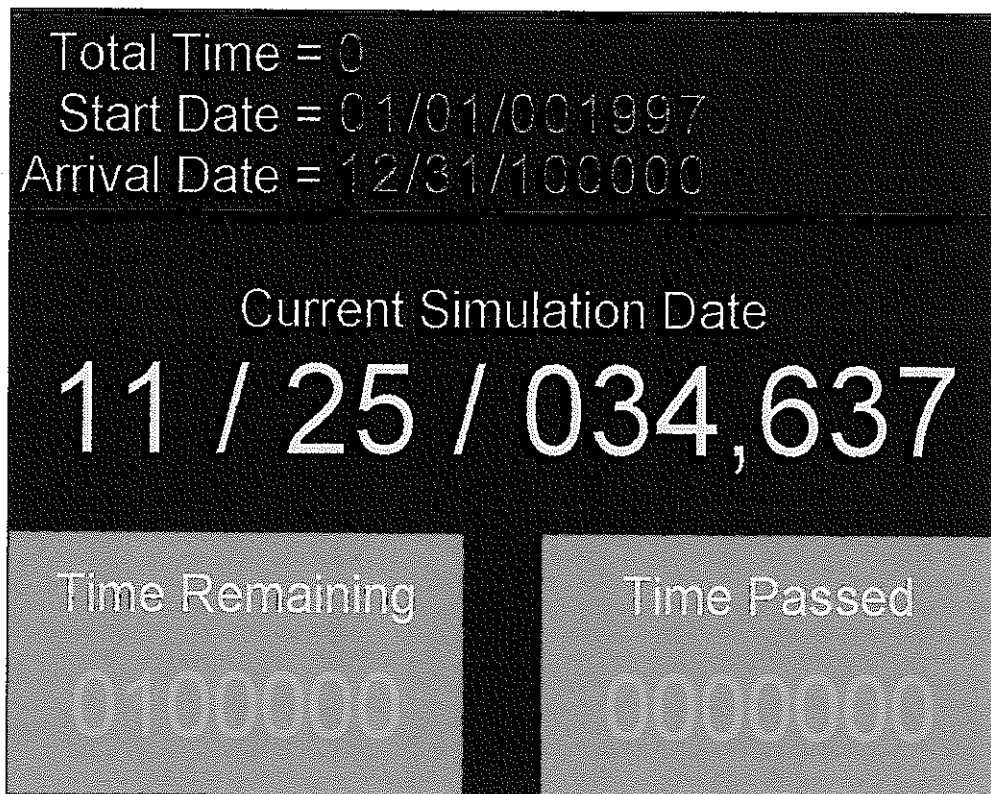


図 1 2 シミュレーション・パラメータ表示画面 2



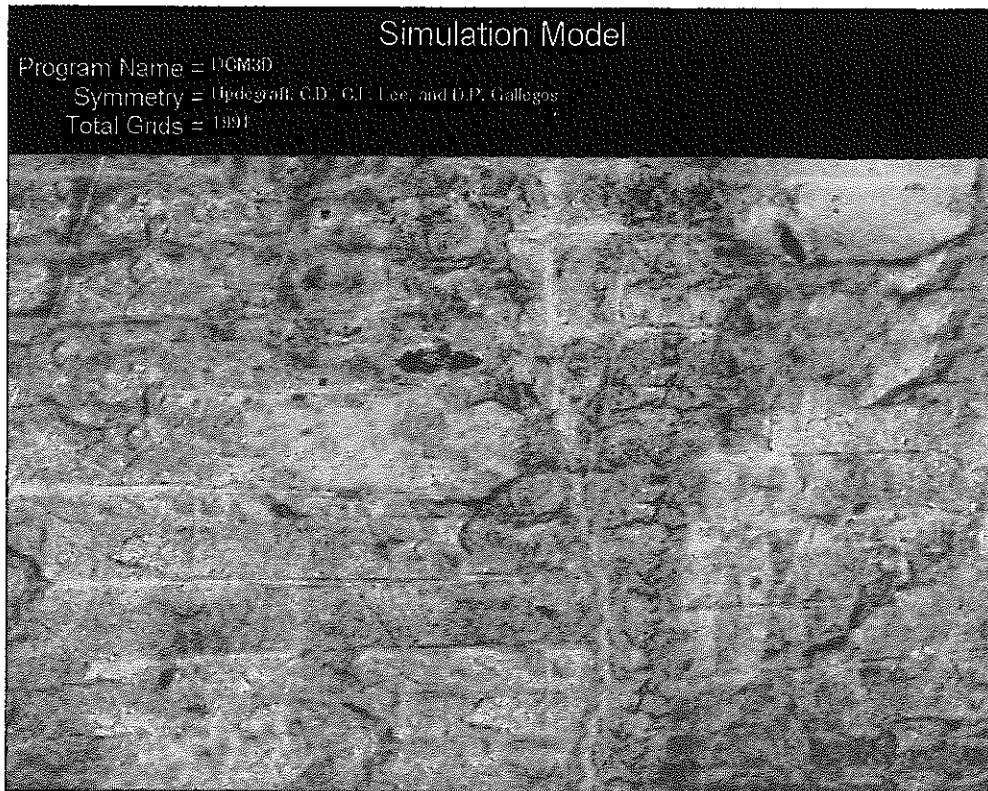


図 1 3 シミュレーション・パラメータ表示画面 3

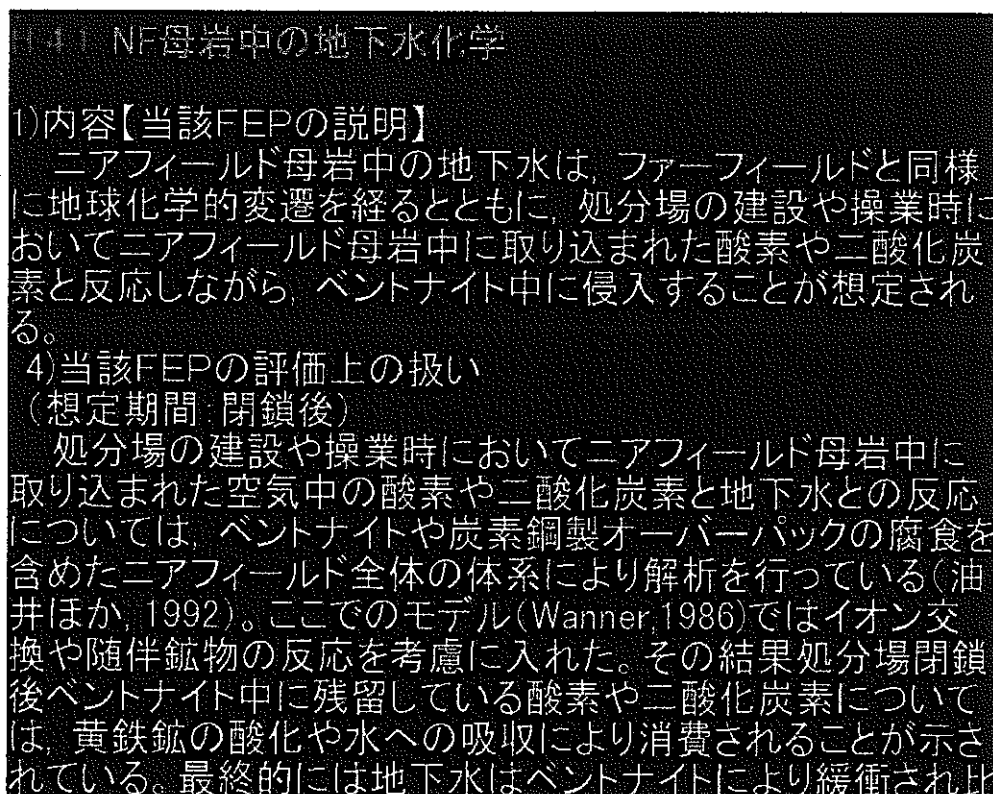


図 1 4 シミュレーション・パラメータ表示画面 4



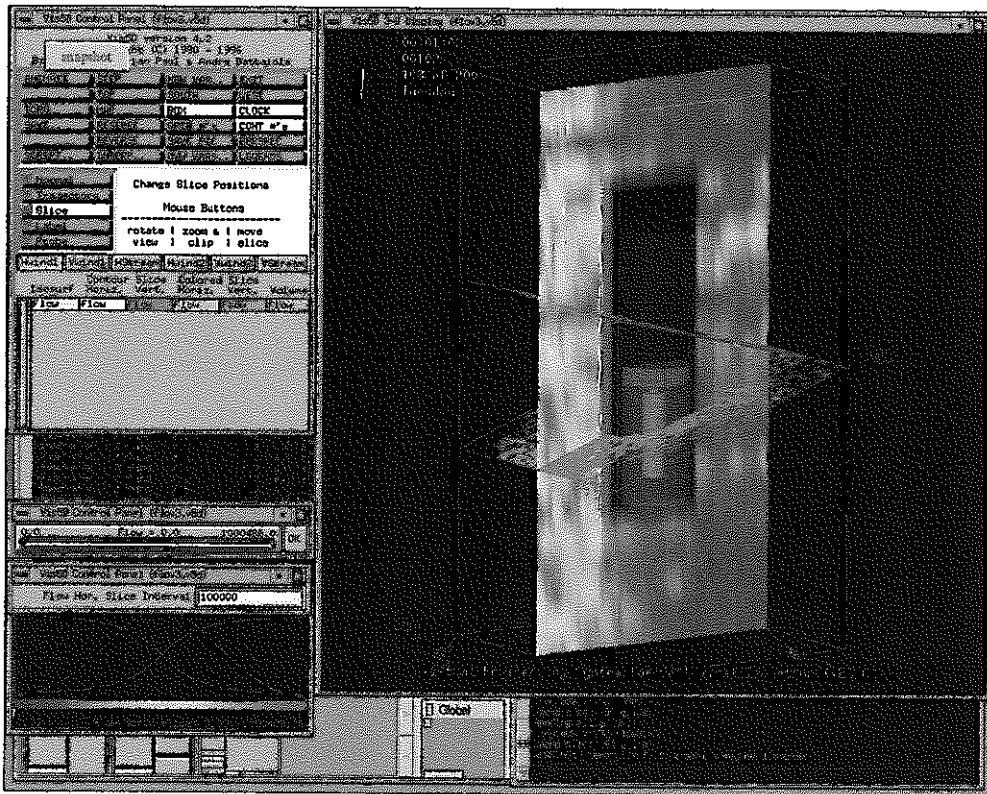


図 1 7 動画化シミュレーション表示画面 3

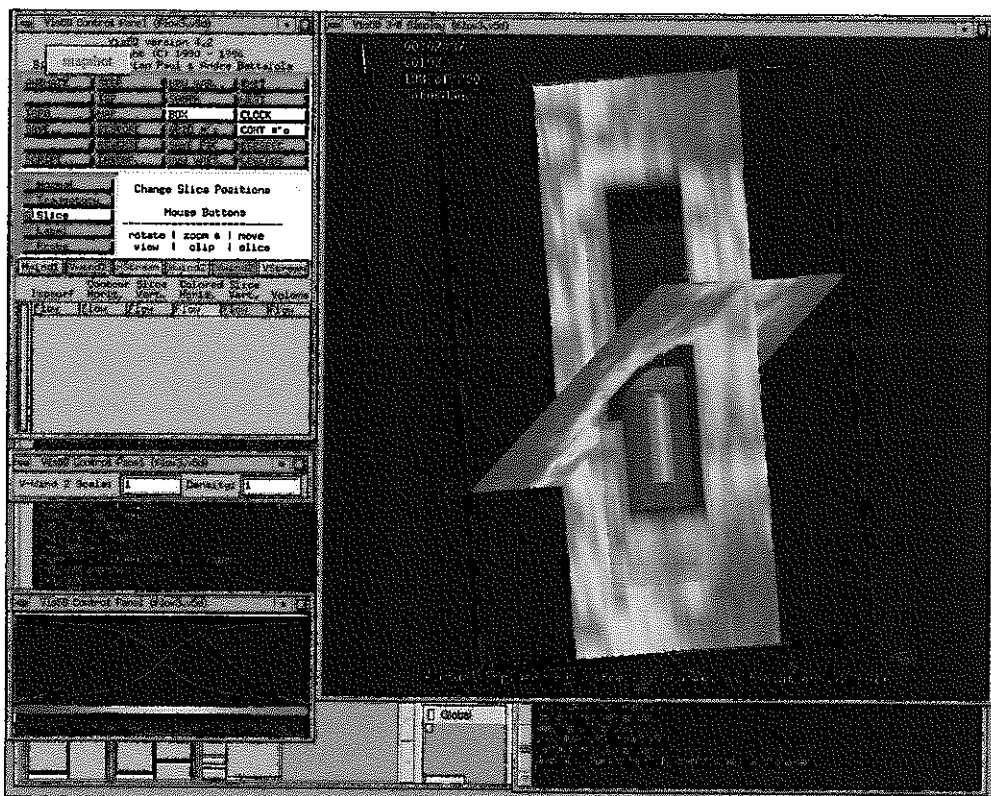


図 1 8 動画化シミュレーション表示画面 4

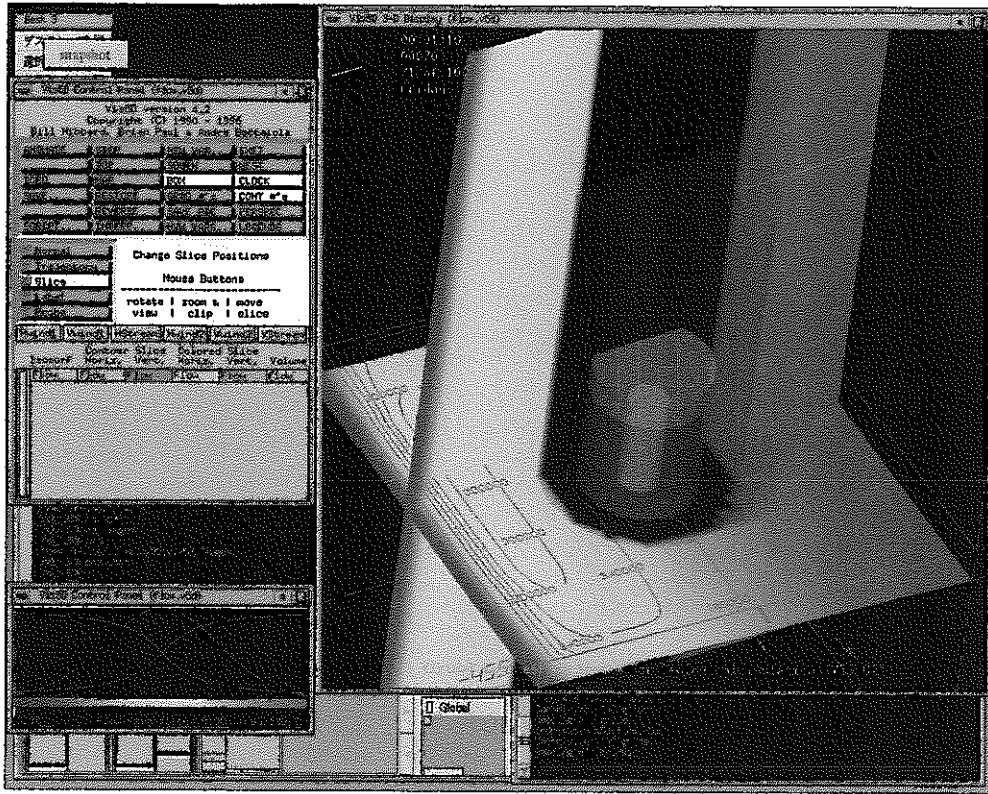


図 19 動画化シミュレーション表示画面 5

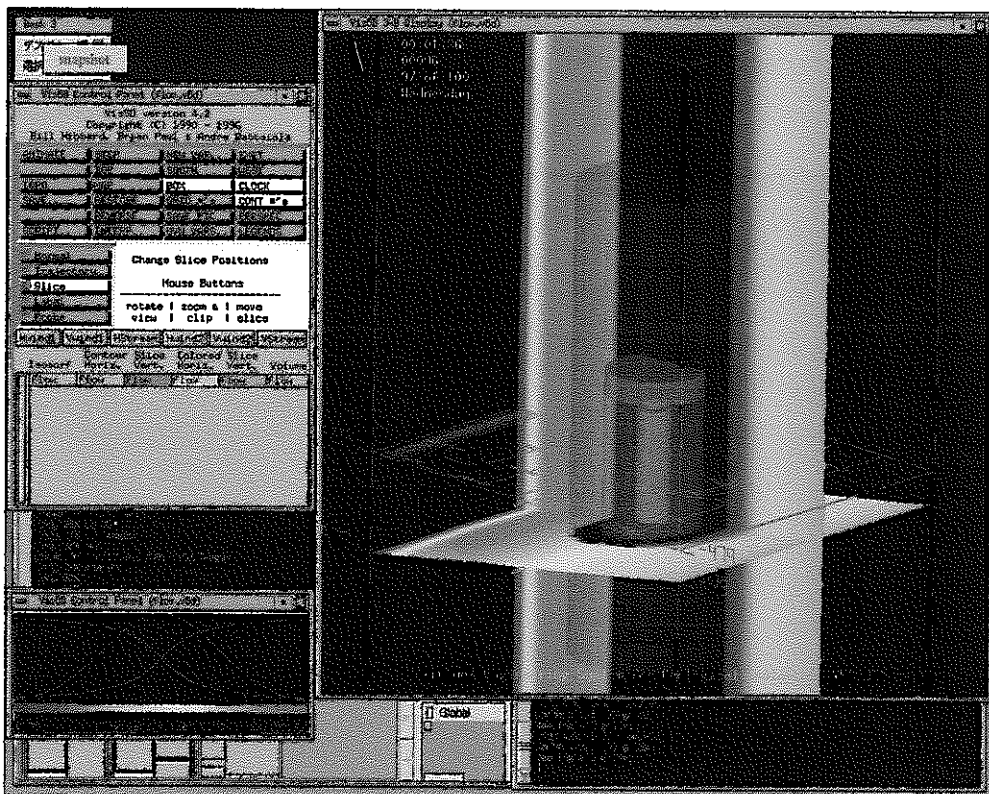


図 20 動画化シミュレーション表示画面 6



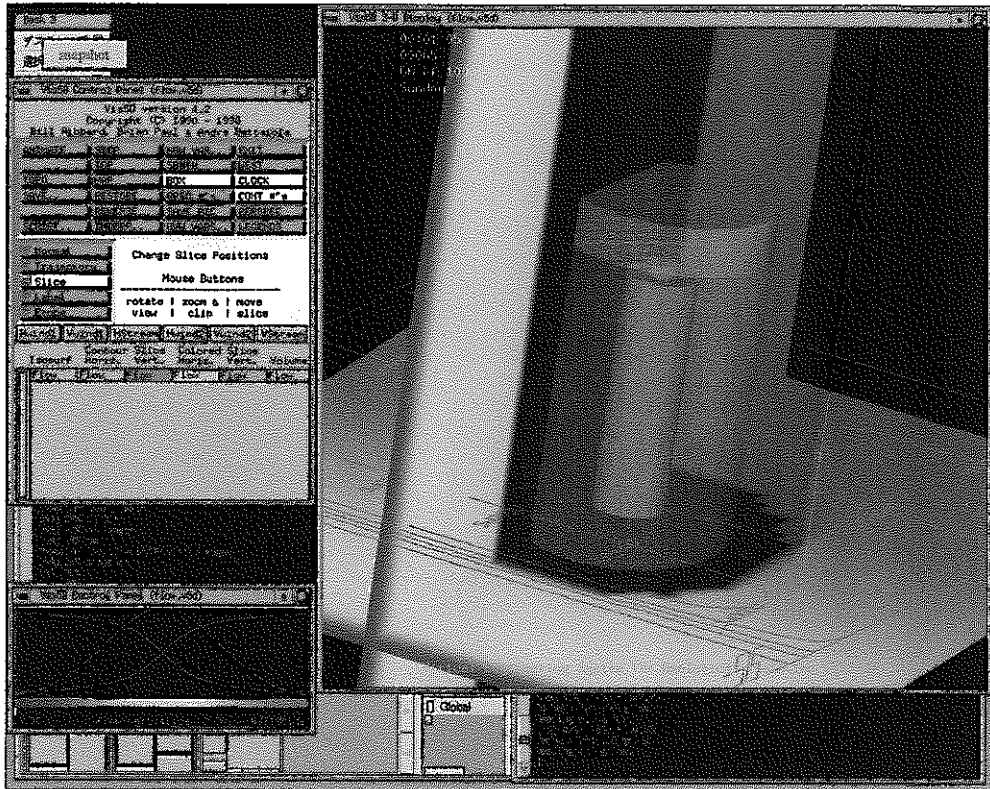


図 2 3 動画化シミュレーション表示画面 9

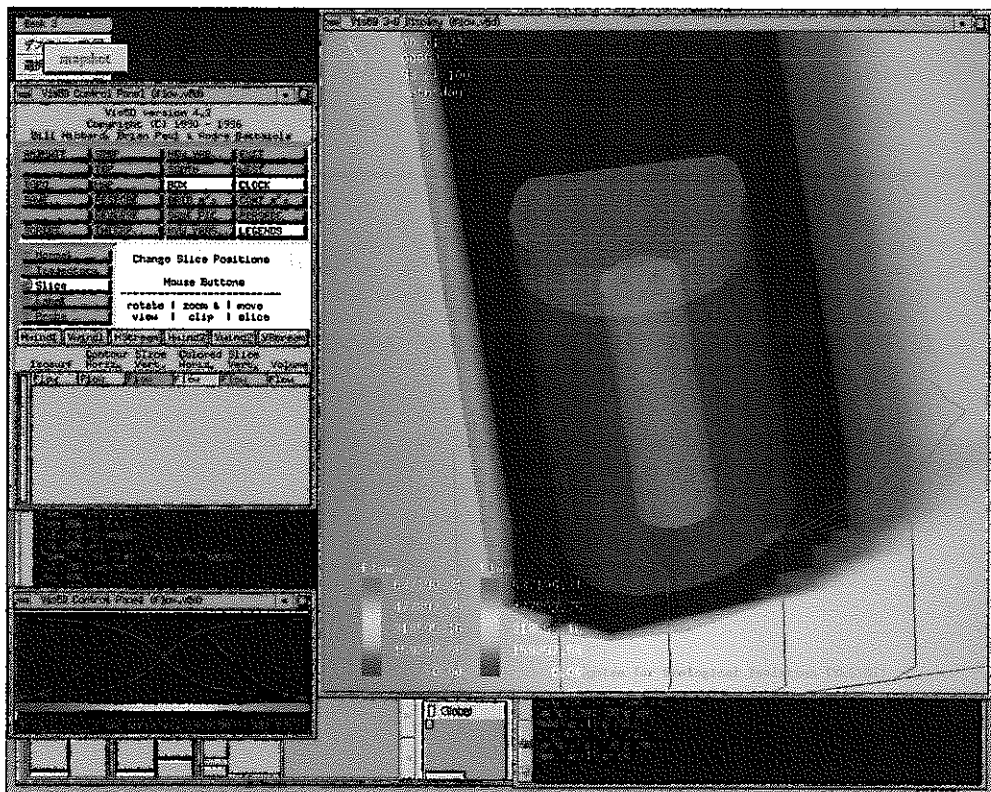


図 2 4 動画化シミュレーション表示画面 1 0

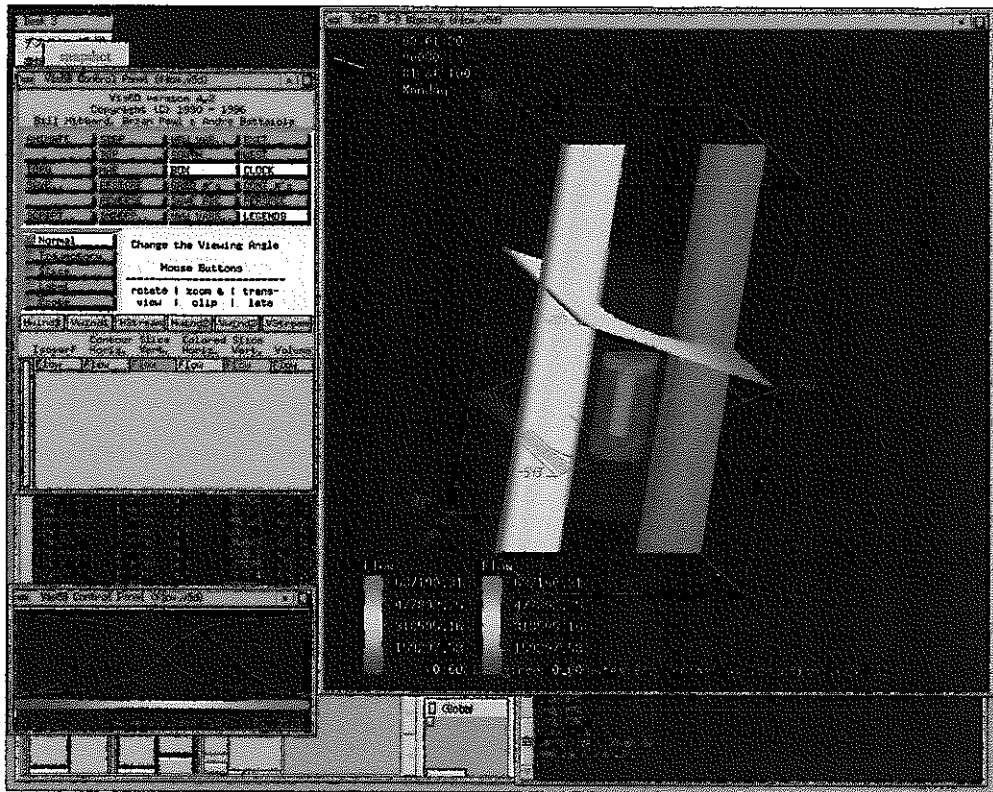


図 2 5 動画化シミュレーション表示画面 1 1

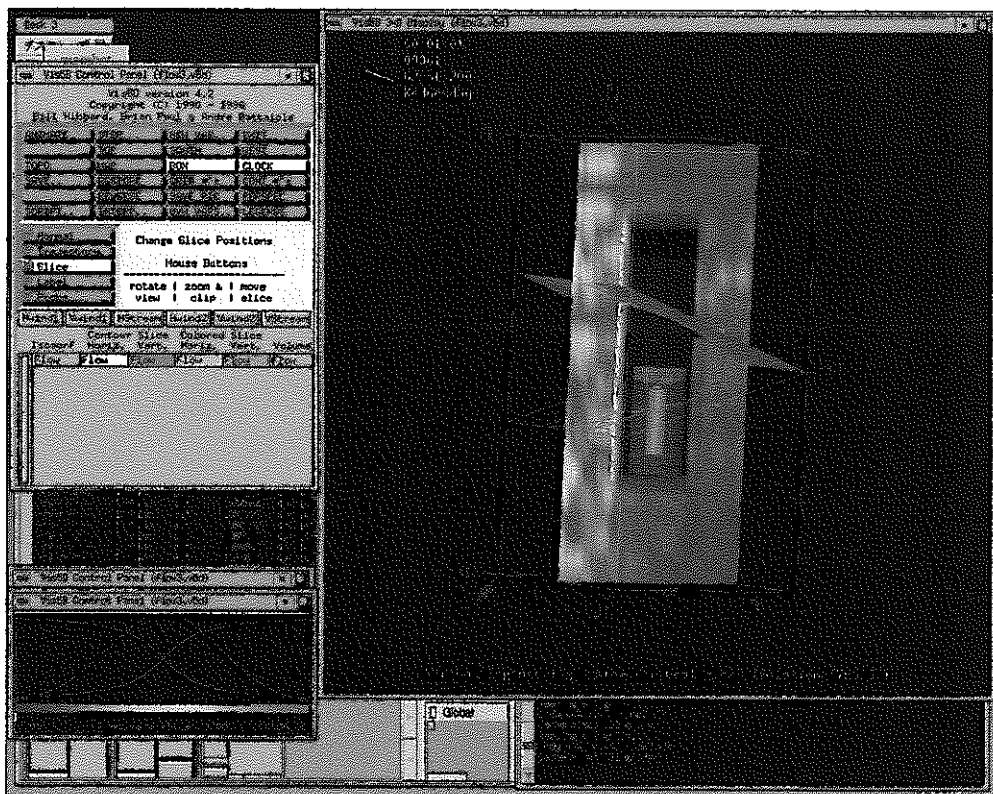


図 2 6 動画化シミュレーション表示画面 1 2

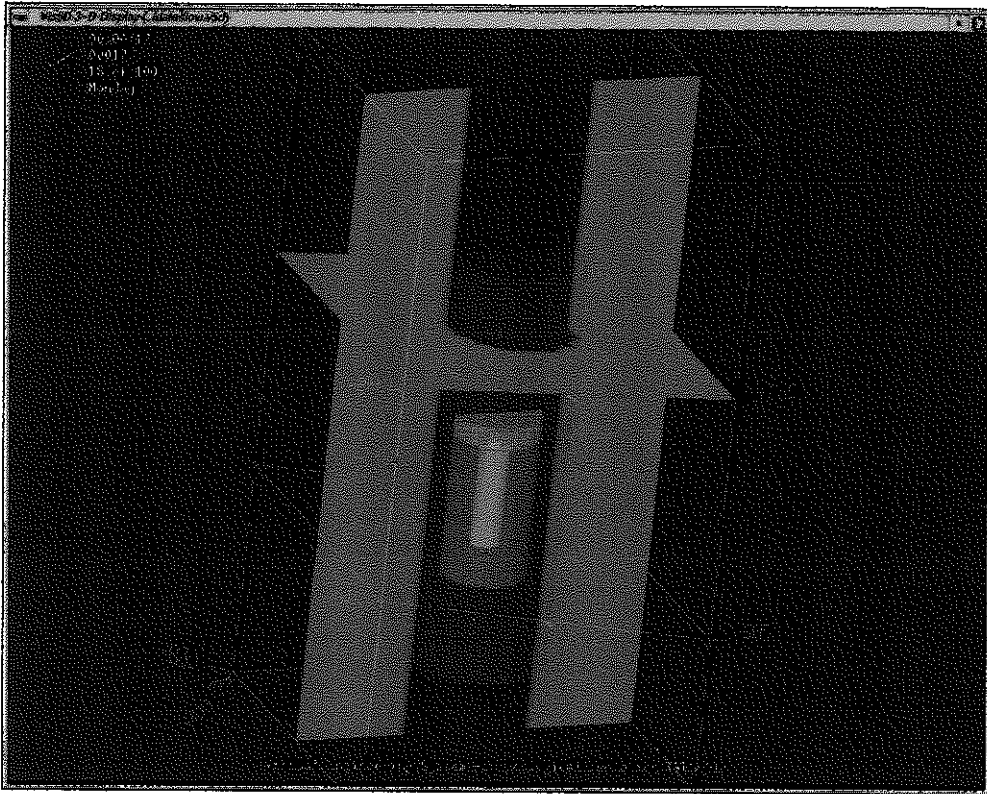


図 2 7 動画化シミュレーション表示画面 1 3

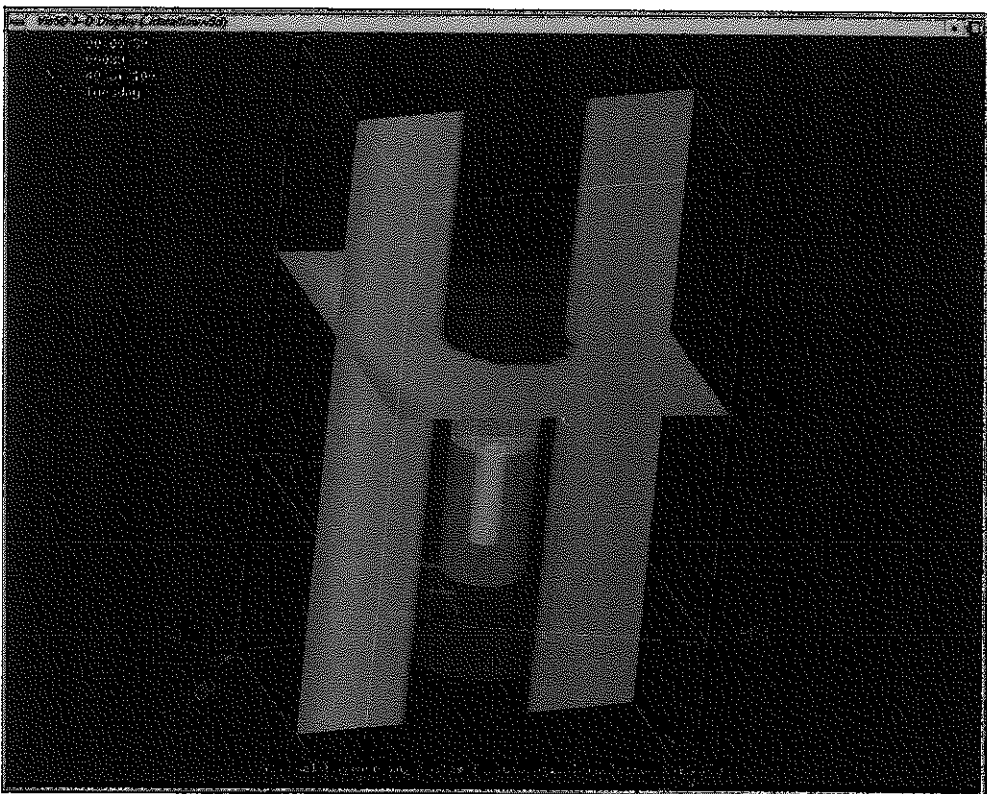


図 2 8 動画化シミュレーション表示画面 1 3



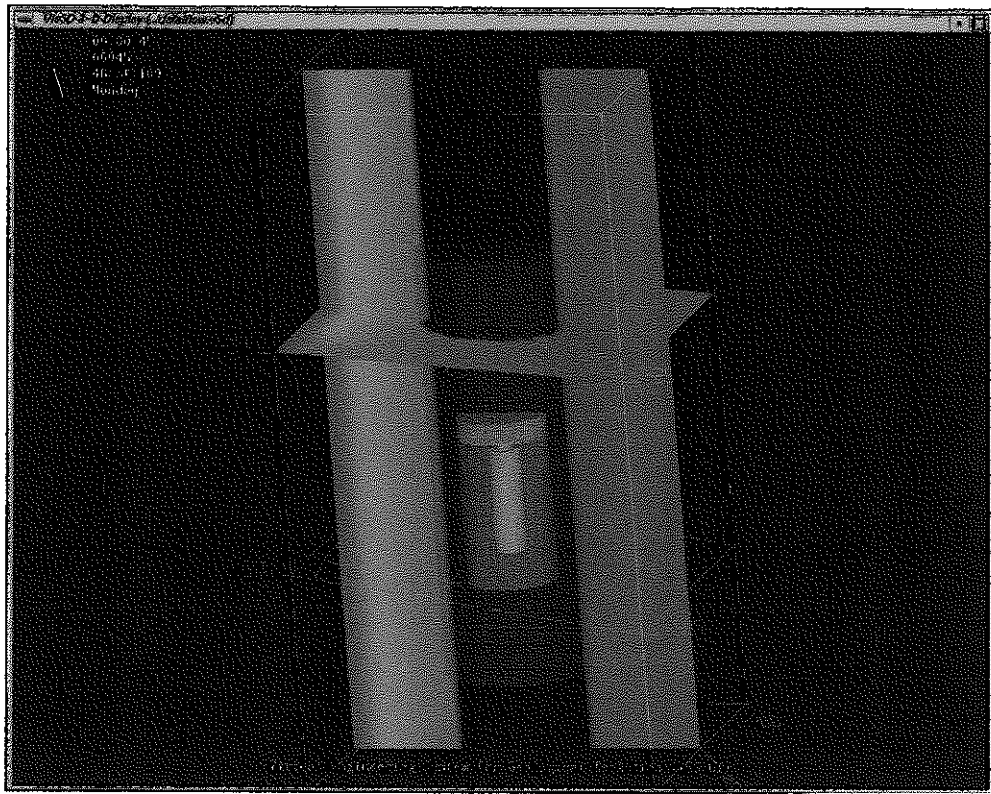


図 2 9 動画化シミュレーション表示画面 1 4

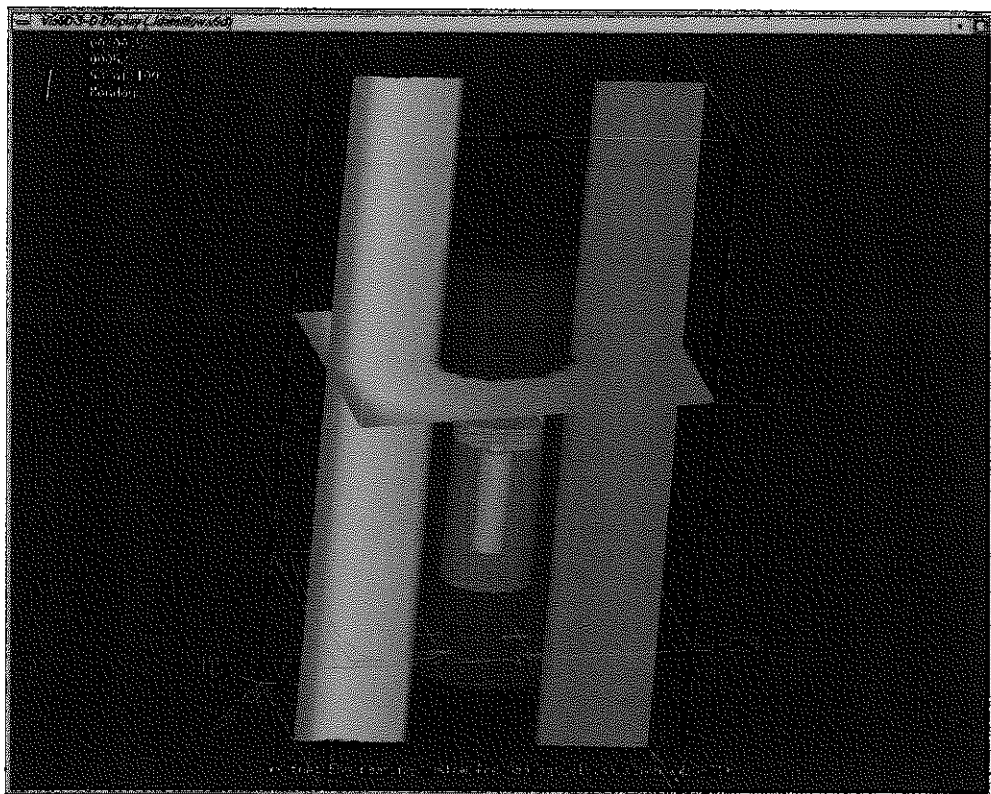


図 3 0 動画化シミュレーション表示画面 1 5

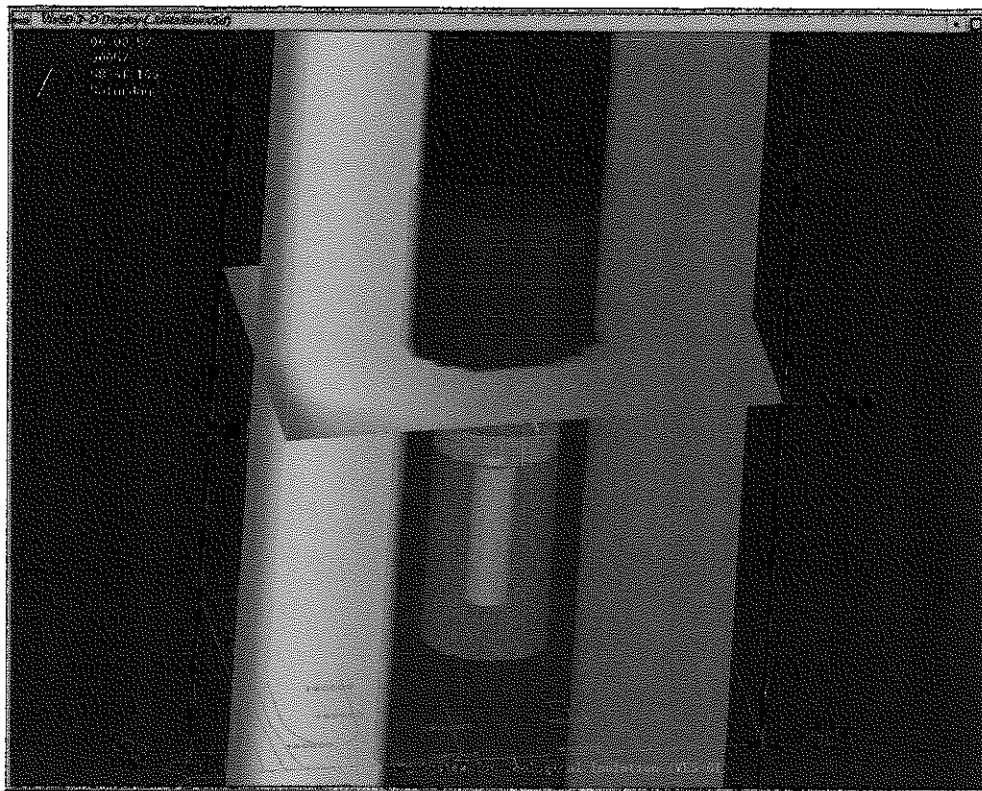


図 3 1 動画化シミュレーション表示画面 1 6

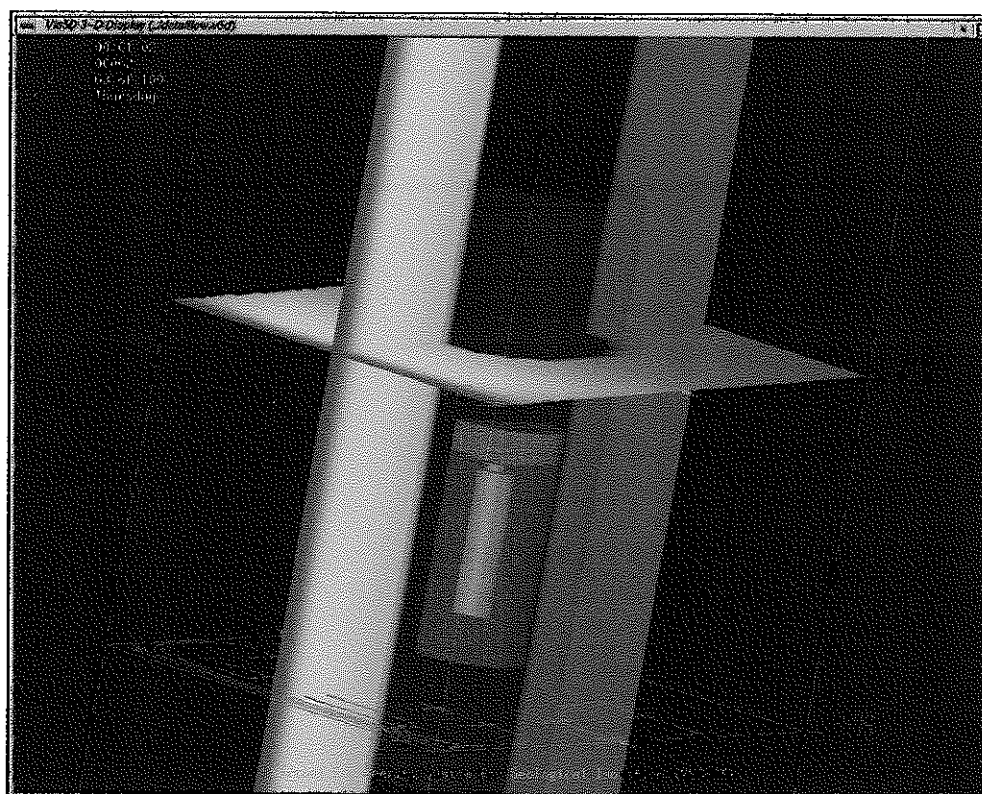


図 3 2 動画化シミュレーション表示画面 1 7

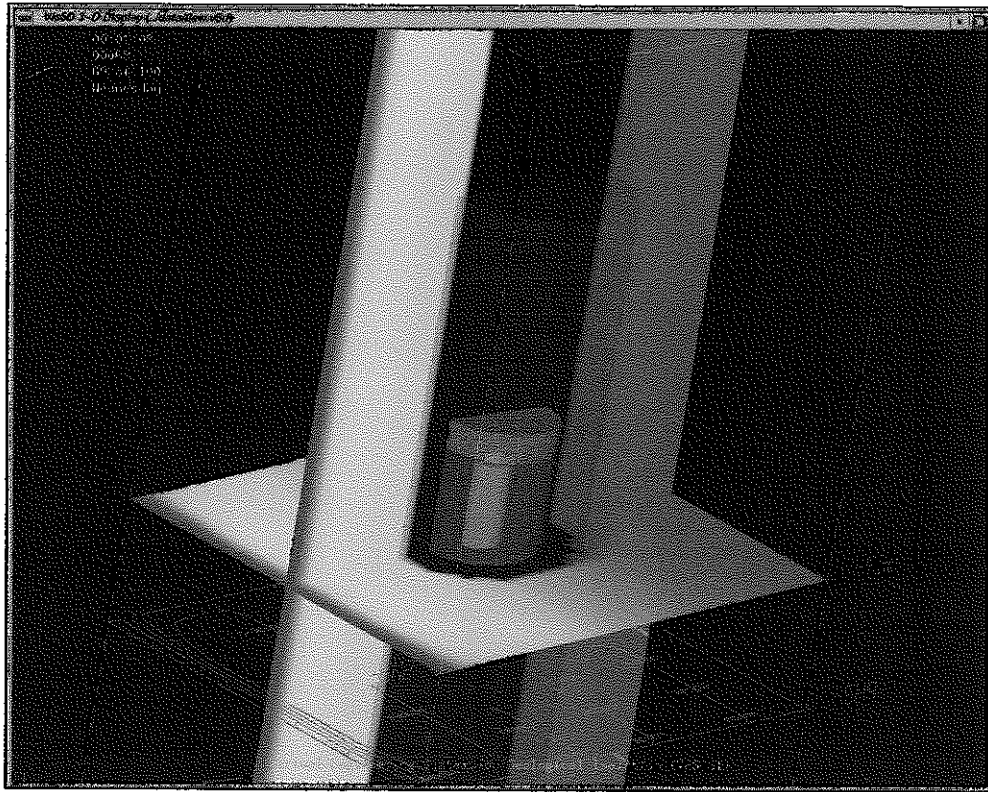


図 3 3 動画化シミュレーション表示画面 1 8

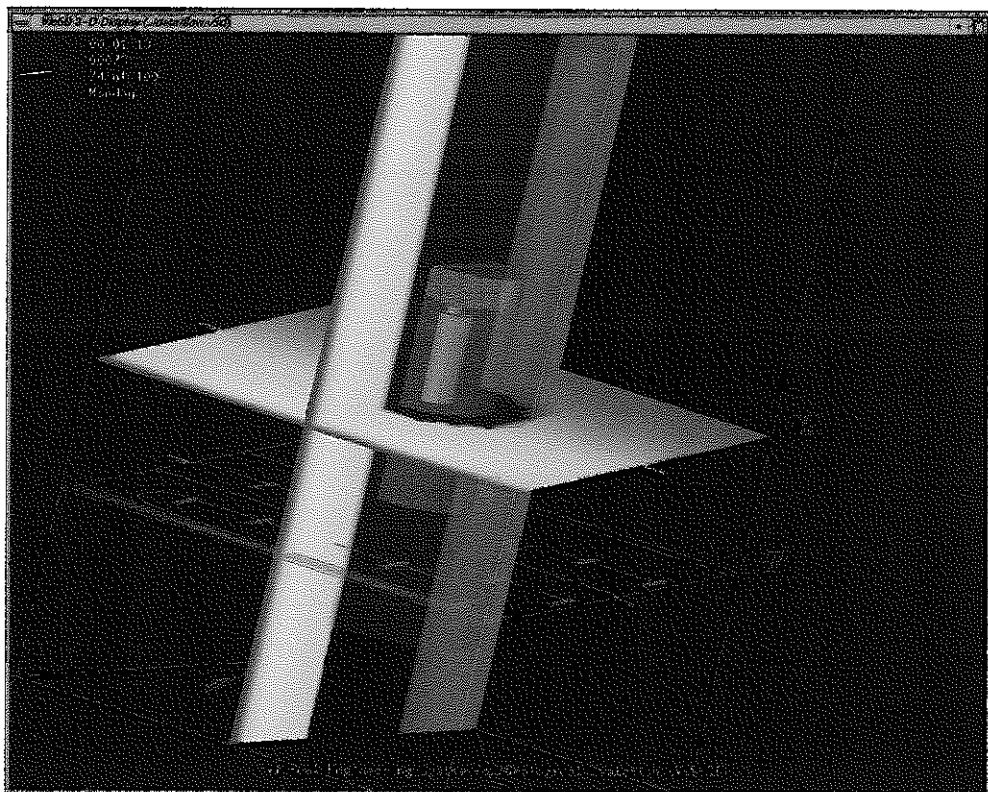


図 3 4 動画化シミュレーション表示画面 1 8

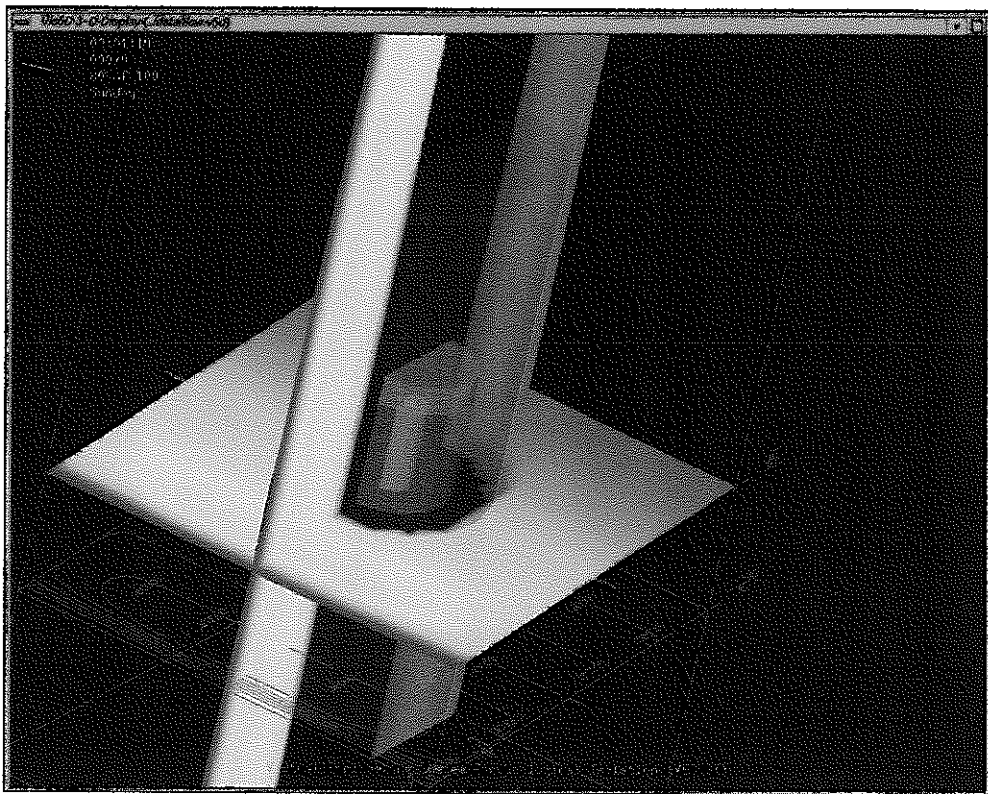


図 3 5 動画化シミュレーション表示画面 1 9

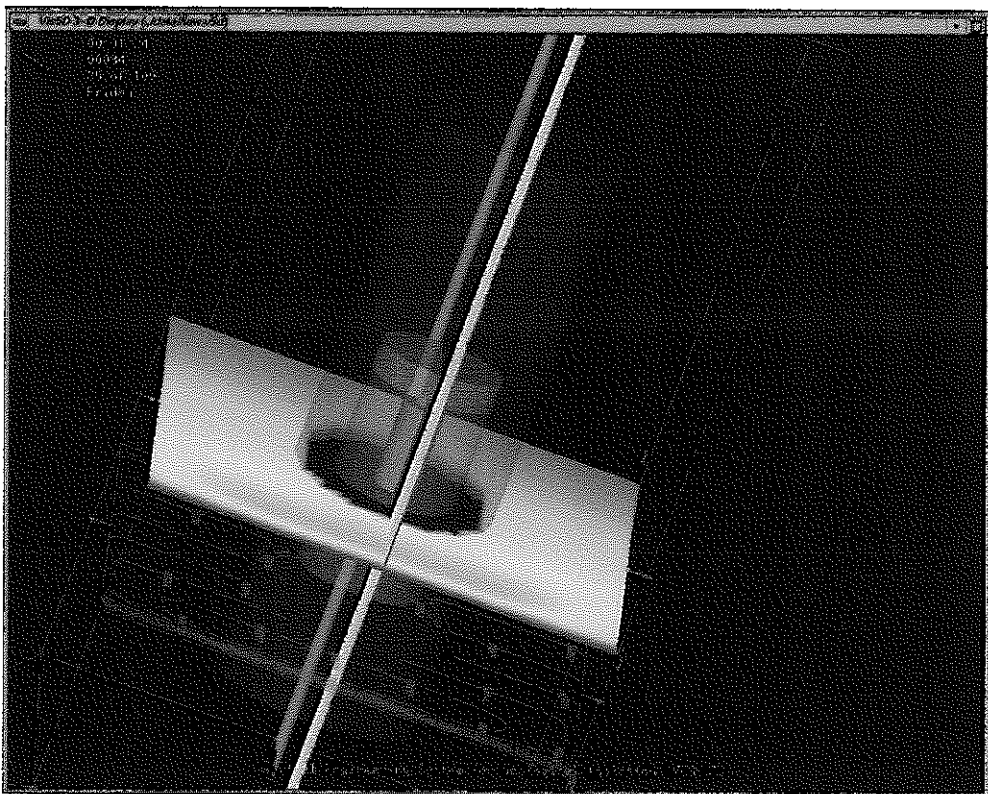


図 3 6 動画化シミュレーション表示画面 1 9

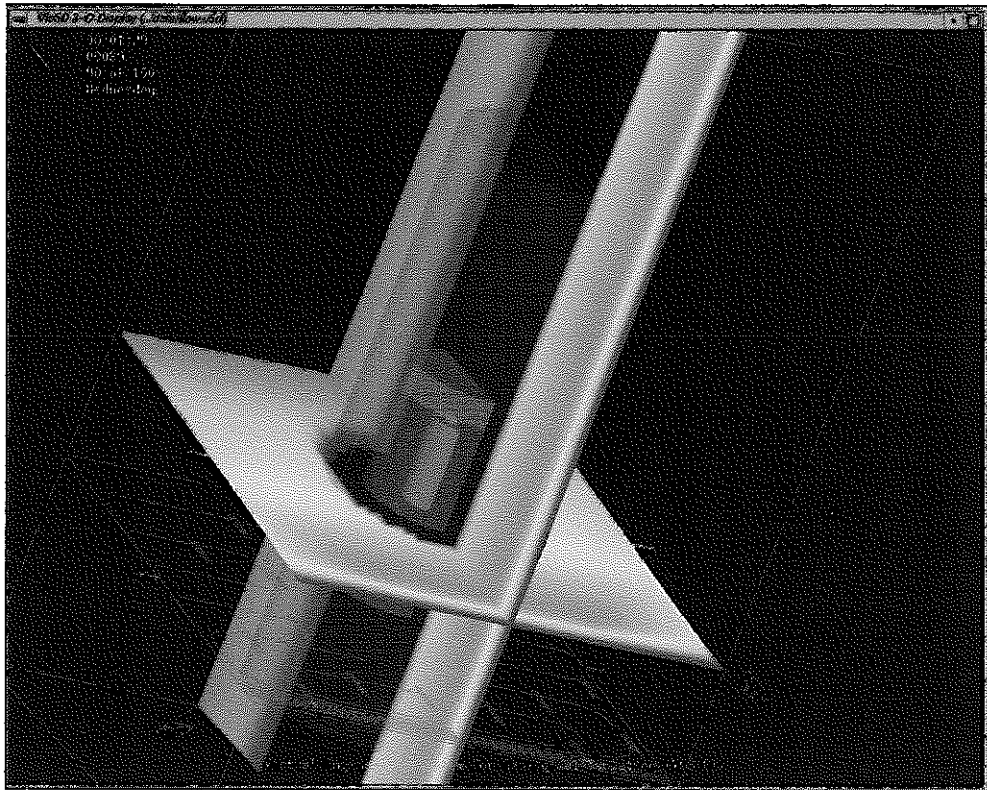


図 3 7 動画化シミュレーション表示画面 2 0

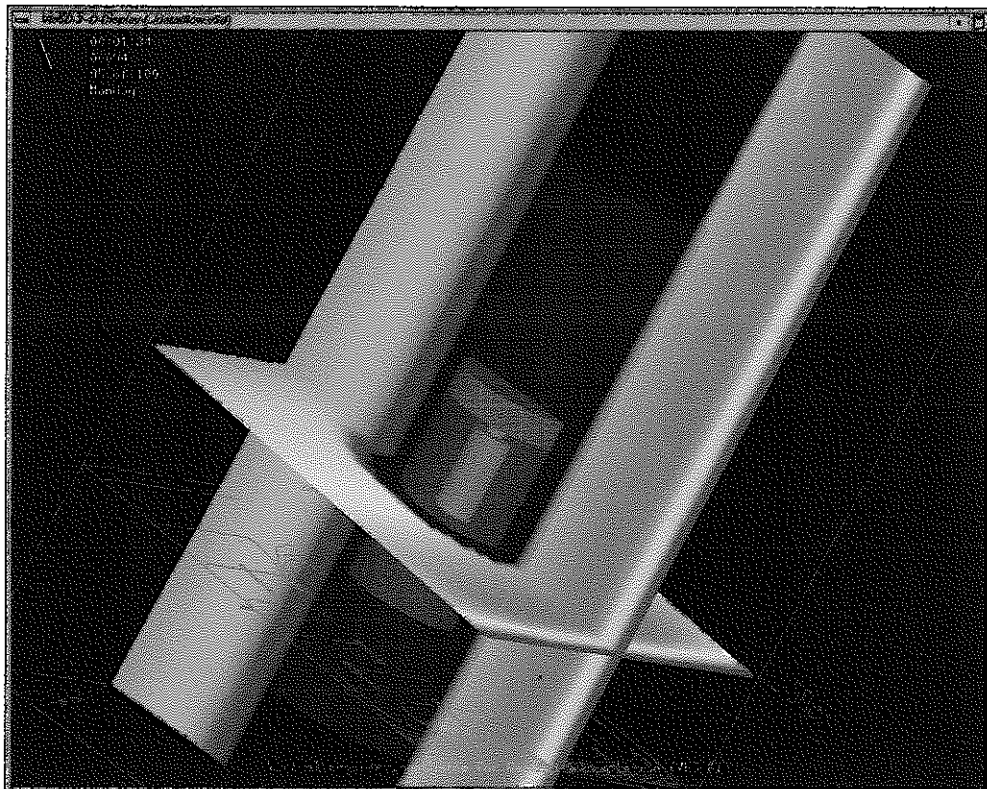


図 3 8 動画化シミュレーション表示画面 2 1

付録一B

参考文献

## 対話技術関連の参考文献

- OpenGL Programming Guide, The Official Guide to Learning OpenGL, Release 1, OpenGL Architecture Review Board, Jakie Neider, Tom Davis, Mason Woo, Addison - Wesley Publishing Company.
- OSF/Motif ウィジェッド上達法「第2版」リリース1. 2、D.L.マックマインズ、小畑喜一、林秀幸訳、アジソン・ウエスレイ・トッパン 情報科学シリーズ 37
- Object - Oriented Programming with the X Window System Tool Kit Jerry D. Smith BNN Bug News Network Wiley.
- Alan Borning. The programming language aspects of ThingLab, a constraint-oriented simulation laboratory. ACM Transactions on Programming Languages and Systems, Vol. 3, No. 4, pp.353-387, October 1981.
- Alan Borning and Robert Duisberg. Constraint-based tools for building user interfaces. ACM Transactions on Graphics, Vol. 5, No. 4, pp.345-374, October 1986.
- Joelle Coutaz. Architecture models for interactive software. In ECOOP 89 Proceedings, pp. 383-399, July 1989.
- Adele Goldberg and David Robson. Smalltalk-80: The Language and its Implementation. Addison-Wesley, 1983.
- Mark Green. Report on dialogue specification tools. In Gilnther E. Pfaff, editor, User Interface Management Systems, pp. 9-20. Springer-Verlag, 1983.
- H. Rex Hartson and Deborah Hix. Human-computer interface development: Concepts and systems for its management. ACM Computing Surveys, Vol. 21, No. 1, pp.5-92, March 1989.
- Glenn Krasner and Stephen Pope. A cookbook for using the Model-View-Controller user interface paradigm in Smalltalk-80. Journal of Object-Oriented Programming, Vol. 1, No. 3, pp.26-49, August/September 1988.
- Wm Leler. Constraint Programming Languages, Their Specification and Generation, pp. 10. Addison-Wesley, 1988.
- Henry Lieberman. Using prototypical objects to implement shared behavior in object oriented systems. In OOPSLA '86 Proceedings, pp. 214-223, September 1986.
- Mark A. Linton, Jon M. Vlissides, and Paul R. Calder. Composing user interfaces with interviews. IEEE Computer, Vol. 22, No. 2, pp.8-22, February 1989.
- Pattie Maes. Concepts and experiments in computational reflection. In OOPSLA '87 Proceedings, pp. 147-155, October 1986.
- Johan H. Maloney, Alan Borning, and Bjorn N. Freeman-Benson. Constraint technology for user-interface construction in ThingLab II. In OOPSLA '89 Proceedings, pp. 381-388, October 1989.
- David A. Moon. Object-oriented programming with Flavors. In OOP-SLA '86 Proceedings, pp. 1-8, September 1986.
- Makoto Murata and Koji Kusumoto. Daemon: Another way of invoking methods. Journal of Object-Oriented Programming, Vol. 2, No. 2, pp.8-12, July/August 1989.
- Makoto Murata and Hiroshi Masuichi. V-A: An architecture model for multiple views. Submitted to ACM User Interface Software Technology, 1992.
- Geoffrey A. Pascoe. Encapsulators: A new software paradigm in Smalltalk-80. In OOPSLA '86 Proceedings, pp. 341-346, September 1986.
- Mark J. Stefik, Daniel G. Bobrow, and Kenneth M. Kahn. Integrating access-oriented programming into a multiparadigm environment. IEEE Software, Vol. 3, No. 1, pp.10-18, January 1986.
- Alan Borning. The programming language aspects of ThingLab, a constraint-oriented simulation laboratory. ACM Transactions on Programming Languages and Systems, Vol. 3, No. 4, pp.353-387, October 1981.
- Alan Borning and Robert Duisberg. Constraint-based tools for building user interfaces. ACM Transactions on Graphics, Vol. 5, No. 4, pp.345-374, October 1986. [Coutaz 89] Joelle Coutaz. ECOOP 89 Proceedings, pp Architecture models for interactive software. In . 383-399, July 1989.
- Adele Goldberg and David Robson. Smalltalk-80: The Language and its Implementation. Addison Wesley, 1983.
- H. Rex Hartson and Deborah Hix. Human-computer interface development: Concepts and systems for its management. ACM Computing Surveys, Vol. 21, No. 1, pp.5-92, March 1989.
- Glenn Krasner and Stephen Pope. A cookbook for using the Model-View-Controller user interface paradigm in Smalltalk-80. Journal of Object-oriented Programming, Vol. 1, No. 3, pp.26-49, August/September 1988.
- Henry Lieberman. Using prototypical objects to implement shared behavior in object oriented systems. In OOPSLA '86 Proceedings, pp. 214-223, September 1986.
- Mark A. Linton, Jon M. Vlissides, and Paul R. Caldering user interfaces with interviews. IEEE Computer, Vol. 22, No. 2, February 1989.

- Pattie Maes. Concepts and experiments in computational reflection. OOPSLA '87 Proceedings, pp. 147-155, October 1986.
- Johan H. Maloney, Alan Borning, and Bjorn N. Freeman-Benson. Constraint technology for user-interface construction in ThingLab II. In OOPSLA '89 Proceedings, pp. 381-388, October 1989.
- David A. Moon. Object-oriented programming with Flavors. In OOPSLA '86 Proceedings, pp. 1-8, September 1986.
- Makoto Murata and Koji Kusumoto. Daemon: Another way of invoking methods. Journal of Object-Oriented Programming, Vol. 2, No. 2, pp.8-12, July/August 1989.
- Mark J. Stefk, Daniel G. Bobrow, and Kenneth M. Kahn. Intergrating access-oriented programming into a multiparadigm environment. IEEE Software, Vol. 3, No. 1, pp.10-18, January 1986.
- Makoto Murata and Hiroshi Masuichi. V-A: An architecture model for multiple views. Submitted to ACM User Interface Software Technology, 1992.
- Geoffrey A. Pascoe. Encapsulators: A new software paradigm in Smalltalk-80. In OOPSLA '86 Proceedings, pp. 341-346, September 1986.
- Brooks, R.A.: A Robust Layered Control System for a Mobile Robot. IEEE J. Robotics and Automation, RA-2 '86.
- Newell, A. Unified , Theory of Cognition, Harvard University Press
- Nelson, Minar, Roger Burkhart, Chris Langton, and Manor Askenazi, The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulation.
- Roger Burkhart, Manor Askenazi, and Nelson Minar. Swarm release documentation. Available as <http://www.santafe.edu/projects/swarm/release/swarmdocs/> -
- NeXT corporation. Object Oriented Programming and the Objective C Language. Addison-Wesley, 1993. Also available as <http://www.next.com/Pubs/Documents/OPENSTEP/ObjectiveC/objectoc.htm>.
- D.L. DeAngelis and Louis J. Gross. Individual-based models and approaches in ecology. Chapman and Hall, 1992.
- John Carnahan, Song-gang Li, Carlo Costantini, Yeya T. Tour6, and Charles E. Taylor. Computer simulation of dispersal by Anopheles Gambiae s.l. in West Africa. In Artificial Life V, 1996.
- John K Ousterhout. Tcl and the Tk toolkit. Addison-Wesley, 1994.
- David Robson. Smalltalk-80: the language and its implementation. Addison-Wesley, 1985.
- John H. Miller, Active Nonlinear Test (ANTs) of Complex Simulation Models.
- Bankes, Steve. 1993. "Exploratory Modeling for Policy Analysis." Operations Research 41: 435-49.
- Bankes, Steven and James Gillogly. 1994. Proceedings of the Third Annual Conference on Evolutionary Programming, Anthony V. Sebald and Lawrence J. Fogel (eds.). River Edge, New Jersey: World Scientific Publishing Co., pp. 353-60.
- Cox, David C. and Paul Baybutt. 1981. "Methods for Uncertainty Analysis: A Comparative Survey." Risk Analysis 1:251-8.
- Goldberg, David E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Reading, Massachusetts: Addison Wesley.
- Hayes, Brian. 1993. "Balanced on a Pencil Point." American Scientist 81:510-16.
- Holland, John H. 1975. Adaptation in Natural and Artificial Systems. Ann Arbor, Michigan: University of Michigan Press.
- Holland, John H. and John H. Miller. 1991. "Artificial Adaptive Agents and Economic Theory." American Economic Review, Papers and Proceedings 81:365-70.
- Iman, Ronald L. and Jon C. Helton. 1988. "An Investigation of Uncertainty and Sensitivity Analysis Techniques for Computer Models." Risk Analysis 8:71-90.
- Meadows, Dennis L., William W. Behrens 111, Donella H. Meadows, Roger F. Naill, Jorgen Randers and Erich K.O. Zahn. 1974. Dynamics of Growth in a Finite World. Cambridge: Wright-Allen Press.
- Meadows, Donella H., Dennis L. Meadows, and Jorgen Randers. 1992. Beyond the Limits. Post Mills, Vermont: Chelsea Green Publishing.
- Morgan, M. Granger and Max Henrion. 1990. Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis. Cambridge: Cambridge University Press.
- Nordhaus, William D. 1992. "Lethal Model 2: The Limits to Growth Revisited." Brookings Panel on Economic Activity, draft.
- George J. Gumerman and Timothy A. Kohler, Crating Alternative Cultural Histories in the Prehistoric Southwest: Agent based Modeling in Archaeology.
- Binford. L. R.. and J. A. Sabloff 1982 Paradigms. Systematics. and Archaeology. Journal of Anthropological Research 38:137-153.



- Brumfiel, E. 1992 Distinguished Lecture in Archaeology: Breaking and Entering the Ecosystem-Gender, Class, and Faction Steal the Show. *American Anthropologist* 94:551-567
- Boyd, R., and P. J. Richerson 1985 *Culture and the Evolutionary Process*. University of Chicago Press, Chicago.
- Cavalli-Sforza, L. L., and M. W. Feldman 1981 *Cultural Transmission and Evolution: A Quantitative Approach*. Princeton University Press, Princeton.
- Cordell, L. 1972 *Settlement Pattern Changes at Wetherill Mesa, Colorado: A Test Case for Computer Simulations in Archaeology*. Ph.D. Dissertation, University of California Santa Barbara. University Microfilms International, Ann Arbor.
- Dean, J. S., A. J. Lindsay and W.J. Robinson 1978 Prehistoric Settlement in Long House Valley, Northeastern Arizona. In *Investigations of the Southwestern Anthropological Research Group : An Experiment in Archaeological Cooperation*. The Proceedings of the 1976 Conference, edited by R.C. Euler and G.J. Gumerman, pp. 25-4. Museum of Northern Arizona, Bulletin 50, Flagstaff.
- DeVore, I. 1988 Prospects for a Synthesis in the Human Behavioral Sciences. In *Emerging Syntheses in Science*, edited by D. Pines, pp. 5-65. Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley, Reading MA.
- Doran, J., M. Palmer, N. Gilbert, and P. Mellars 1994 The EOS Project: Modelling Upper Paleolithic Social Change. In *Simulating Societies: The Computer Simulation of Social Phenomena*, edited by Nigel Gilbert and Jim Doran, pp. 195-221, UCL Press, London.
- Dove, D. E. 1984 Prehistoric Subsistence and Population Change along the Lower Agua Fria River, Arizona: A Model Simulation. *Anthropological Research Papers* 32, Arizona State University, Tempe.
- Elster, J. 1989 *Nuts and Bolts for the Social Sciences*. Cambridge University Press, Cambridge.
- Epstein, J. M. and R. L. Axtell 1996 *Growing Artificial Societies: Social Science from the Bottom Up*. MIT Press and Brookings Institution Cambridge. (in Press).
- Gell-Mann, M. 1994 Complex Adaptive Systems. In *Complexity: Metaphors, Models, and Reality*, edited by G. A. Cowan, D. Pines, and D. Meltzer, pp.17-28. Proceedings Volume XIX, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, MA.
- Gumerman, G. J. 1971 The Distribution of Prehistoric Aggregates. *Proceedings of the Southwestern Anthropological Research Group*, Prescott College Studies in Anthropology No. 1, Prescott College Press, Prescott.
- Gumennan, G. J. and J. S. Dean 1989 Prehistoric Cooperation and Competition in the Western Anasazi Area. In *Dynamics of Southwest Prehistory*, edited by L. S. Cordell and G. J. Gumerman, pp. 99-148. Smithsonian Institution Press, Washington, D.C.
- Gumerman, G. J., and M. Gell-Mann, editors 1994 *Understanding Complexity in the Prehistoric Southwest*. Santa Fe Institute Studies in the Sciences of Complexity Proceedings Volume XVI. Addison-Wesley, Reading, MA.
- Hodder, I. 1986 *Reading the Past: Current Approaches to Interpretation in Archaeology*. Cambridge University Press, Cambridge.
- Kauffman, S. A. 1994 Whispers from Camot: The Origin of Order and Principles of Adaptation in Complex Nonequilibrium Systems. In *Complexity: Metaphors, Models, and Reality*, edited by G. A. Cowan, D. Pines, and D. Meltzer, pp. 83-136. Santa Fe Institute Studies in the Sciences of Complexity. Proceedings Volume XIX, Addison-Wesley, Reading, MA.
- Kohler, T. A., J. D. Orcutt, K. L. Petersen, and E. Bliedman 1986 Anasazi Spreadsheets: The Cost of Doing Agricultural Business in Prehistoric Dolores. In *Dolores Archaeological Program: Final Synthetic Report*, compiled by D. A. Breternitz, C. K. Robinson, and G. T. Gross, pp. 525-538. Bureau of Reclamation, Engineering and Research Center, Denver.
- Kohler, T. A., and C. R. Van West 1996 The Calculus of Self Interest in the Development of Cooperation: Sociopolitical Development and Risk Among the Northern Anasazi. In *Evolving Complexity and Environment: Risk in the Prehistoric Southwest*, edited by J. A. and B. Bagley Tainter, pp. 169-196. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Vol. XXIV. Addison-Wesley, Reading, MA.
- Kohler, T. A., C. R. Van West, E. P. Carr, and C. G. Langton 1996 Agent-based Modeling of Prehistoric Settlement Systems in the Northern American Southwest. Paper presented at the Third International Conference/Workshop on Integrating GIS and Environmental Modeling, January, Santa Fe.
- Palmer, M., and J. Doran 1993 Contrasting Models of Upper Paleolithic Social Dynamics: A Distributed Artificial Intelligence Approach. In *Computing the Past: Computer Applications and Quantitative Methods in Archaeology (C192)*, edited by J. Andresen, T. Madsen, and I. Scollar, pp. 251-262. Aarhus University Press, Aarhus, Denmark.
- Renfrew, C. 1987 Problems in Modelling Socio-cultural Systems. *European Journal of Operational Research*

30:179-192.

- Renfrew, C., M. J. Rowlands, and B. A. Seagraves, editors 1982 *Theory and Explanation in Archaeology: The Southampton Conference*. Academic Press, New York.
- Reynolds, R. G. 1986 *An Adaptive Computer Model for the Evolution of Plant Collecting and Early Agriculture in the Eastern Valley of Oaxaca*. In Guila Naquitz.
- *Archaic Foraging and Early Agriculture in the Oaxaca, Mexico*, edited by K. V. Flannery, pp. 439-500. Academic Press, New York
- Sabloff J. A., editor 1981 *Simulations in Archaeology. A School of American Research Book*. University of New Mexico Press, Albuquerque
- Shennan, S. J. 1991 *Tradition, Rationality, and Cultural Transmission*. In *Processual and Postprocessual Archaeologies: Multiple Ways of Knowing the Past*, edited by R.
- W. Preucel, pp. 197-208. Center for Archaeological Investigations, Carbondale. 1993 *After Social Evolution: A New Archaeological Agenda?* In *Archaeological Theory: Who Sets the Agenda?*, edited by N. Yoffee and A. Sherratt, pp. 53-59. Cambridge University Press, Cambridge.
- Thomas D. H. 1972' *A Computer Simulation Model of Great Basin Shoshonean Settlement Patterns*. In *Models in Archaeology*, edited by D. Clarke, pp. 79-102. Methuen, London
- Trigger, B. D. 1984 *Archaeology at the Cross Roads: What's New?* *Annual Review of Anthropology*, 13:4, edited by D. Siegel, A. Beals, and S. Tyler, pp. 275-300. Annual Reviews, Inc., Palo Alto.
- Van West C. 1994 *Modeling Prehistoric Agricultural Productivity in Southwestern Colorado: A GIS Approach*. Reports of Investigations 67. Department of Anthropology, Washington State University, Pullman.

## 協調計算関連の参考文献

- PNC TN1410 91-009、地層処分研究開発の現状（平成元年度）、1990年3月、動力炉・核燃料開発事業団
- PNC TN 1410 92-081、高レベル放射性廃棄物地層処分研究開発の技術報告書、平成3年度、平成4年9月、動力炉・核燃料開発事業団
- PNC TN 1340 93-001、動燃技報、高レベル放射性廃棄物の地層処分研究開発特集、1993年3月、動力炉・核燃料開発事業団
- PNC TN 1410 94-094、地層処分研究開発の現状（平成5年度）、1994年11月、動力炉・核燃料開発事業団
- PNC TN 1410 96-071、地層処分研究開発の現状（平成8年度）、1996年12月、動力炉・核燃料開発事業団
- JAERI - research 96-011、高レベル放射性廃棄物地層処分の安全性研究、1996年3月、日本原子力研究所
- デービス、P. 他、『数学的経験』、柴垣和三雄・清水邦夫他訳、1987年、森北出版
- アイゲン、M. 他、『自然と遊戯（ゲーム）』、寺本英・伊勢典夫他訳、1981年、東京化学同人
- グリック、『カオス』、止田院亮監修、大貫昌子訳、1991年、年新潮文庫
- ピーターソン、I.、『現代数学ワンダーランド』、奥田晃訳、1990年、新曜社
- ・パイトゲン、H.、リヒター、P.、『フラクタルの美』、宇敷重広訳、1991年、シュプリンガー・フェアラーク東京
- ポーストル、D.、『宇宙の綱目』、はやしはじめ訳、1993年、白揚社
- ラッカー、R.、『無限と心』、好田順治訳、1986年、現代数学社
- スチュアート、I.、『カオスの世界像』、須田不二夫・三村和男訳、1992年、白揚社
- パーシグ、R.、『禅とオートバイ修理技術』、五十嵐美克訳、1990年、めるくまーる
- ポーデン、G.、ハリス、K.、『ことばの科学入門』、廣瀬肇訳、1984年、メディカルリサーチセンター
- オッペンハイム、A.、シェイファー、R.、『デジタル信号処理（上・下）』、伊達玄訳、1978年、コロナ社
- ガモフ、G.、『1、2、3—無限大』、(G-ガモフコレクション、『宇宙=1、2、3—無限大所収』)、崎川範行訳、1992年、白揚社
- ニューマン、W. 他、『対話型コンピュータグラフィックス (I, U)』、大須賀節雄監訳、1984年、マグロウヒル出版
- ウィルソン、I.、『トリノの聖骸布』、木原武一訳、1985年、文芸春秋
- ブリッグス、J.、ピート、D.、『鏡の伝説』、高安秀樹他訳、1991年、ダイヤモンド社
- フェダー、J.、『フラクタル』、松下貢・早川美徳他訳、1991年、啓学出版
- マンデルブロ、B.、『フラクタル幾何学』、広中平祐監訳、1985年、日経サイエンス社
- パイトゲン、H.、ザウペ、D.、『フラクタル・イメージ』、山口昌哉監訳、1990年、シュプリンガー・フェアラーク東京
- エイブラハム、H.、ショー、D.、『ダイナミクス力学系（工・下）』、東保光彦訳、1990年、現代数学社
- ベルジェ、P. 他、『カオスの中の秩序』、相澤洋二訳、1992年、産業図書
- ホフスタッター、D.、『メタマジック・ゲーム』、竹内郁雄他訳、1990年、白揚社
- シュベンク、T.、『カオスの自然学』、赤井敏夫訳、1986年、工作舎
- ガードナー、M.、『数学ゲーム (I, I)』、高木茂男訳、1974年、講談社ブルーバックス
- ホフスタッター、D.、『ゲーデル、エッシャー、バッハ』、野崎昭弘他訳、1985年、白揚社、

河日津一郎、『Morphogenesis』、1985年、JCC出版局

- V. Brauer, et. al. "Analysis of criteria for high level waste disposal"
- M. Jensen et. al. "Protection of man and nature - Some criteria acceptance of nuclear waste repositories"
- S. Norrby "Radioactive waste management · An international view"
- NEA, IAEA, CEC "Disposal of radioactive waste: Can long - term safety be evaluated? An international collective opinion" (1991)
- NEA, Safety series No. 111-G-4.1 Siting of Geological disposal facilities (1994)
- Andersson, J. and Eng, T. (1989) The joint SKI/SKB Scenario development project. In Proc. NEA/IAEA/CEC Symposium, Paris 1989.
- Andersson, J. et al. (1989) The joint SKI/SKB Scenario development project. SKI reports TR 89:14. Billington, D.E. et al (1989) Radiological assessment of deep geological disposal: work for UK Nirex Ltd. In Proc. NEA/IAEA/CEC Symposium, Paris 1989.
- Bingham, F.W. and Barr, G.E. (1979) Scenarios for long-term release of radionuclides from a nuclear waste repository in Los Medanos region of New Mexico. Sandia National Laboratories report SAND78-1730.
- Bonano, E.J., et al. (1989) Demonstration of a Performance Assessment methodology in High-level waste disposal in basalt formations. NUREG/CR-47 SAND86-2325.
- Bonano, E.J., et al. (1990) The use of expert judgements in performance assessment of High-level waste repositories. In Safety Assessment of Radioactive Waste Repositories. Proceedings of the Paris Symposium, OECD, Paris, 1990.
- Bonano, E.J., Cranwell, R. and Davis, P. (1989) Performance assessment methodologies for the analysis of high-level nuclear waste repositories. Rad Waste Man and Nucl Fuel Cell Cycle, Harwood Acad. Publ. 13 (1-4) 229-239
- Boulton, G.S. Time-dependent modelling of environmental change: The prediction of Quaternary glaciations. In Safety assessment of radioactive waste repositories. Proceedings, OECD, Paris, 1990.
- CEC: Performance assessment of geological isolation systems for radioactive waste - Summary. CEC Nuclear and Science report series PAGIS, 11775EN (1998).
- Cranwell, R.M.; Campbell, J.E. Helton, J.C.; Inman, R.L.; Longsine, D.E.; Ortiz, N.R. Runkle, G.E. and Shortencarrier, M.J. (1987) "Risk methodology for Geological disposal of Radioactive Waste." Final Report, SAND81-2537, NUREG/CR-2452, Sandia National Laboratories, USA
- Cranwell, R. M. et al. (1987) Risk methodology for geologic disposal of radioactive waste: final report. NUREG/CR-2452, SAND81-2573.
- Cranwell, R. M. et al. (1982, revised 1990) Risk methodology for geologic disposal of radioactive waste: scenario selection procedure. NUREG/CR-1667, SAND80-1429.
- D'Alessandro, M. and Bonne, A. Radioactive waste disposal into a plastic clay formation. CEC report EUR7111 (Also by Harwood Academic Publishers, 1981)
- (1986) First report on the acquisition of data for use in Probabilistic risk assessment of underground disposal of radioactive waste. UK DoE report DoE/RW/86.071.
- Dames and Moore Intl. (1977) Evaluation of environmental change and its effects on the radiological performance of a shallow engineered disposal facility at Elstow, Bedfordshire. UK DoE report: TR-D&M-9; DoE/RW/87.124. UK Department of the Environment, London, Wiley, London.
- Dames and Moore (1988) Earthquake effects on groundwater systems: an Introductory Review. UK DoE report DoE/RW/89.052.
- Dames and Moore (1988) Background studies: climatic and geomorphological aspects of the evolution of shallow-land buried sites for radioactive waste disposal. UK DoE report DoE/RW/88.055.
- Environmental Protection Agency (1985) Environmental standards for the Management and disposal of spent nuclear fuel, high level and transuranic radioactive wastes, Final Rule. 40CFR Part 191, Federal Register 50 (182) Washington D.C.
- Foley, M.G. et al. (1982) Geologic simulation for a hypothetical site in the Columbia Plateau: Results. Pacific Northwest Laboratory report PNL-3542-2. Check versus other ref which may be earlier version.
- Eisenberg, N. (1992) Performance assessment for High-level waste repositories in risk assessment: A survey of characteristics, applications and methods used by Federal Agencies for Engineered systems. Nuclear Regulatory Commission, Washington D.C. 20555-0001.
- Grimwood, P. and Thegerstrom, C. (1990) Assessment of the risk associated human intrusion at radioactive waste disposal sites. Some observations from NEA workshop. In Safety assessment of radioactive waste repositories. Proceedings of the Paris Symposium. OECD, Paris 1990.

- Guzowski, R. V., et al. (1989) Potential scenarios for use in performance assessment high-level radioactive waste repositories in unsaturated tuff. NUREG/CR-477 SAND86-7170.
- Hunter, R.L. (1983) Preliminary scenarios for the release of radioactive waste from a hypothetical repository in basalt of the Columbia Plateau. NUREG/CR-3353
- Hunter, R.L. (1983) Preliminary scenarios for the release of radioactive waste from a hypothetical repository in basalt of the Columbia Plateau. Sandia report SAND83-1352
- Hunter, R.L., Barr, G.E. and Bingham, F.W. (1983) Scenarios for consequence assessments of radioactive waste repositories at Yucca Mountain, Nevada test site. Sandia National Laboratories report SAND82-1277.
- Imbrie, J. et al. (1984) The orbital theory of Pleistocene climate. In Milankovitch and climate, ed Berger, A.L., Reidel, Dordrecht, 269-305.
- INTERA Environmental Consultants Inc., (1983) GSM: Geologic simulation model for a hypothetical site in the Columbia Plateau: Large computer version of nuclear waste isolation report ONWI-447
- INTERA Environmental Consultants Inc., (1983) FFSM Far-field state model, Office of Nuclear waste Isolation report ONWI-436.
- International Atomic Energy Agency (1981) Safety assessment for the underground disposal of radioactive wastes. Safety series No, 56, IAEA, Vienna.
- Kane, P. (1992) VANDAL version 1.3 Technical overview. UK DoE report. DoE/RR/92.095. UK Department of the Environment, London.
- Laurens, J.-M., Thompson, B.G.J. and Sumerling, T.J. (1989) The development and application of an integrated radiological risk assessment procedure using Time-dependent probabilistic risk analysis. In Proc NEA/IAEA/CEC Symposium, Paris.
- Logan, S.E. and Berbeno, M.C. (1977) "Geologic modelling in risk assessment radioactive waste management" in Risk analysis and Geologic modelling: relation to the disposal of radioactive wastes into geological formations. Proceedings of an OECD/NEA and CEC workshop in Ispra 23-27 May, 1977. 77-1158.
- NAGRA: Projekt Gewähr 1985 - Feasibility study; NAGRA Projekt Gewähr report series NGB 85-01 - NGB 85-09, NAGRA, Baden, Switzerland, 19485.
- National Research Council: A study of the isolation system for geologic disposal of radioactive wastes; National Academy Press, Washington D.C. (1983).
- NEA (1989) Risks associated with human intrusion at radioactive waste disposal sites. Proceedings of an NEA workshop. OECD/NEA, Paris.
- NEA (1994) Characterisation of Long-term Geological Changes for Disposal sites. Proceedings of NEA workshop, Paris, France 19-21 September 1994
- NEA SCENARIO CATALOGUE (1988) A compilation of questionnaire responses on scenarios for performance assessments of radioactive waste disposal. OECD/NEA, Paris.
- NEA [PAAG/DOC (88)2] Systematic approaches to Scenario development. A preliminary report of the NEA working group on the Identification and selection of scenarios for Performance assessment of nuclear waste disposal.
- (1990) Safety assessments of radioactive waste management. Proceedings of the Paris Symposium, 1989. OECD/NEA, Paris 1989.
- (1991) Disposal of radioactive waste: Review of safety assessment methodologies. A report of the Performance Assessment Advisory Group of the Radioactive Waste Management Committee. OECD/NEA, Paris, 1991
- Peaudecerf, P. and Blanc, P.L. (1990) French language publication
- Read, D., Krol, A. & Thompson, B.G.J. & Thompson (1987) A structured programme of research into the Chemical aspects of radioactive waste disposal. In Scientific basis for waste management, Vol 10. Mat Res Soc, Boston USA.
- Ringrose, P.S. et al. Probabilistic simulation of the long-term evolution of radioactive waste disposal sites. In Proc. NEA/IAEA/CEC Symposium, Paris 1989.
- Roberds, W.J., Plum, R.J. and Visca, P.J. (1984) Proposed methodology for completion of scenario analysis for the basalt waste isolation project, Rockwell Hanford Operations Report RHO-BW-CR-147P.
- Sagar, B. and Janetzke, R.W. (1991) Total performance assessment computer code: description of executive module. Center for Nuclear Waste Regulatory Analyses, San Antonio, Texas. CNWRA 91-009.
- SKBF/KBS: Final storage of spent nuclear fuel - KBS-3, Stockholm (1989).
- Stephens, M.E. and Goodwin, B.W. (1989) Scenario analysis for post-closure assessment of the Canadian concept for nuclear waste disposal. In Proc. NEA/IAEA/CEC Symposium, Paris 1989.
- Sumerling, T.J. (1991) Dry run 3: A trial of methodology for the probabilistic risk assessment of underground disposal of radioactive waste that accounts for long-term environmental change: Overview. UK DoE Report TR-DR3-10.

- Sumerling, T.J. and Read, D. (1993) Approaches to long term performance assessment of deep underground disposal of radioactive wastes: A European perspective. *Mat. Res. Soc. Symp. Proc.* [294] pp 951-960
- Thompson, B.G.J. (1989) The time dimension in risk analysis: examples from recent work in the United Kingdom. In *Risk Analysis in Nuclear Waste Management*, ed. A. Saltelli et al. Kluwer, Dordrecht.
- Thompson, B.G.J. (1987) The development for procedures for the assessment of the underground disposal of radioactive wastes: research funded by the Department of the Environment, 1982-1987. *Rad Waste Man and the Nucl Fuel Cell Cycle*, Vol 9 pp215-256.
- Thompson, B.G.J. [PAAG/DOC (88)11] A method of overcoming the limitations of conventional scenario-based risk assessments by using Monte Carlo simulation of possible future environmental changes. Technical report TR-DoE-12.
- Thompson, B.G.J. & Sagar, B. (1993) "The development and application of integrated procedures for post-closure performance assessment, based on Monte Carlo simulation: the probabilistic systems assessment (PSA) approach". *Reliability Engineering and System Safety* [42] pp125 - 160
- Thome, M.C. (1993) The use of expert opinion in formulating conceptual models of underground disposal systems and the treatment of associated bias. *Reliability Engineering and System Safety*.
- Tierney, M.S. (1991) Combining scenarios in a calculation of the overall probability distribution of cumulative releases of radioactivity from the Waste Isolation Pilot Plant, South Eastern New Mexico. SAND 90-0838. Sandia National Laboratories, Albuquerque, NM.
- Tyrer, M. Bennett, D.G. Read, D & Yunus, I (1995) "Near Field and Chemical Transport Modelling. Rep. UK DoE/HMIP TR-Z2-9
- USEPA (1985) "Environmental standards for the Management and disposal of spent nuclear fuel, high level waste and transuranic radioactive waste"; Final Rule. *Federal Register* [50] Washington D.C.
- Wilmot, R.D. (1993) The treatment of climate-driven environmental change and associated uncertainty in post-disposal, AECL-8373-4.
- Ash, J., Sexton, H. (1985) A surface with one local minimum. *Math. Magazine*. 58: 147-149.
- *Computer Graphics World* (1989) Simple Beauty. Nov. 12(11): 78-81. (Describes author's work.)
- Dawkins, R. (1986) *The Blind Watchmaker*. W. W. Norton: New York. (Dawkins also uses the term "biomorph" to describe his interesting computer-generated biological shapes.)
- Dewdney, A. K. (1989) Computer recreations. Catch of the day: biomorphs on Truchet tiles, served with popcorn and snails. *Scien. Amer.* July 261(1): 110-113.
- Fadman, C. (1958) *Fantasia Mathematica*. Simon and Schuster: New York (A book of stories and diversions all drawn from the universe of mathematics).
- Flanigan, F. (1983) *Complex Variables*. Dover: New York. (A good introduction to complex variables.)
- Gardner, M. (1986) *Knotted Doughnuts and Other Mathematical Entertainments*. Freeman: New York.
- Hardy, G. A *Mathematician's Apology*. Cambridge Univ. Press: New York.
- Hobson, E. (1965) *Squaring the Circle*. Chelsea Publishing Co.: New York.
- Lawrence, J. (1972) *A Catalog of Special Plane Curves*. Dover: New York.
- Peterson, I. (1987) Pictures worth a thousand numbers. *Washington Post*, July 19th (Sunday). section C3, first page.
- Peterson, I. (1987) Portraits of Equations. *Science News* 132(12): 184-186 (and cover picture).
- Rivlin, R. (1986) Computer graphics: the arts. *Omni Magazine*. 8: 30.
- Rivlin, R. (1986) *The Algorithmic Image*. Microsoft Press: Redmond, WA. (also see excerpt in *Computer Graphics World*, August 1986).
- Rucker, R. (1988) *Mind Tools. The Five Levels of Mathematical Reality*. Houghton Mifflin Co.: Boston.
- Sorenson, P. (1989) Pickover's strange attractor. *Special Effects Business*. December/ January I (3): 6-7; 53.
- Voelcker, J. (1988) Picturing randomness. *IEEE Spectrum*. August 25(8): 13.
- Anderson, B. (1988) Graphics high end: war of words. *Computer Graphics Today*. July Issue. pg. I. (This article is not about fractals, but rather it describes the new class of graphics supercomputers which can be used to create fractals with startling speed.)
- *The Beauty of Fractals* (1987 photo essay) *Computers in Physics*. Nov/Dec. pp 26-31.
- Aqvist, J., Tapia, O. (1987) Surface fractality as a guide for studying protein-protein interactions. *J. Molec. Graph.* 5(1): 30-34.
- Barnsley, M. (1988) *Fractals Everywhere*. Academic Press: New York. 15
- Briggs, J., Peat, D. (1989) *The Turbulent Mirror*. Harper and Row: New York. (Good for beginners.)
- Boyd, D. (1973) The residual set dimensions of the Apollonian packing. *Mathematika* 20: 170-174. (Very technical reading).
- Brooks, R., Matelski, J. P. (1981) The dynamics of 2-generator subgroups of  $PSL(2, C)$ . In *Riemann Surfaces and*

Related Topics.' Proceedings of the 1978 Stony Brook Conference.

- Kyra, I. and Maskit, B. (eds.) Princeton University Press: New Jersey. (Note: this 1978 paper contains computer graphics and mathematical descriptions of both Julia and Mandelbrot sets).
- Devaney, R., Krych, M. (1984) Dynamics of  $\exp(z)$ . *Ergod. Th. & Dynam. Sys.* 4: 35-52.
- Dewdney, A. K. (1985) Computer Recreations. *Scien. Amer.* 253: 16-24.
- Denning, P. (1985) Computing in the frontiers of science and engineering, cover page description. *Commun. ACM.* Vol 28.
- Douady, A., Hubbard, J. (1982) Iteration des polynomes quadratiques complexes. *Comptes Rendus (Paris)* 2941 : I 23-1 26.
- Foley, J., and van Dam, A. (1984) *Fundamentals of Interactive Computer Graphics.* Addison-Wesley: Massachusetts.
- Family, F. (1988) Introduction to droplet growth processes: simulation theory and experiments. In *Random Fluctuations and Pattern Growth. Experiments and Models.* Stanley, H., Ostrowsky, N. (eds.) Kluwer: Boston.
- Family, F., Meakin, P. (1988) Droplet growth. *Phys Rev. Lett.* 61 :428.
- Family, F., Landau, D., eds. (1984) *Kinetics of Aggregation and Gelation.* North-Holland: Amsterdam.
- Feder, J. (1988) *Fractals.* Plenum: New York.
- Fatou, P. (1919/1920) Sur les equations fonctionelles. *Bull. Soc. Math. Fr.* 47: I 6 1 -27 1 .
- Gardner, M. (1968) Packing of circles and spheres. *Scien. Amer.* 21 8: I 30-1 25.
- Hamilton, W. R. (1969) *Elements of Quaternions, Vol. I and II,* reprinted by Chelsea Publishing Co.: New York.
- Hirsch, M. (1989) Chaos, Rigor, and Hype. *Mathematical Intelligencer.* II (3):6-9. (Pages 8 and 9 include James Gleick's response to the article.)
- Hubbard, J. (1986) Order in chaos. *Engineering: Cornell Quarterly.* 20(3): 20-26 (Winter Issue).
- Julia, G. (1918) Memoire sur l'iteration des fonctions rationnelles, *J. Math. Pure Appl.* 4: 47-245.
- Kadanoff, L. (1986) Fractals: where's the physics? *Physics Today* February, 6-7.
- Kausch-Blecken, H. Schmeling, V., Tschöegl, N. (1970) Osculatory packing of finite areas with circles. *Nature.* March 225: I I 19-1 121.
- Kaye, B. (1989) *A Randomwalk Through Fractal Dimensions.* VCH Publishers: New York.
- La Brecque, M (1985) Fractal Symmetry. *Mosaic* 16: 10-23.
- Lakhtakia, A., Vasundara, V., Messier, R., Varadan, V. (1987) On the symmetries of the Julia sets for the process  $z - zp + c$ . *J. Phys. A: Math. Gen.* 20: 3533-3535.
- Lakhtakia, A., Vasundara, V., Messier, R., Varadan, V. (1987) The generalized Koch Curve. *J. Phys. A: Math. Gen.* 20: 3537-3541.
- Lakhtakia, A., Vasundara, V., Messier, R., Varadan, V. (1986) Self-similarity versus self-affinity: the Sierpiński gasket revisited. *J. Phys. A: Math. Gen.* 19: L985-L989.
- Mandelbrot, B. (1983) *The Fractal Geometry of Nature,* Freeman, San Francisco.
- Mandelbrot, B. (1983) On the quadratic mapping for complex and  $z$ : The fractal structure of its  $M$  set, and scaling. *Physica I 7D:* 224-239.
- Musgrave, K. (1989) The synthesis and rendering of eroded fractal terrains. *Computer Graphics (ACM-SIGGRAPH).* July 23(3): 41-50.
- Norton, A. (1982) Generation and display of geometric fractals in 3-D, *Computer Graphics (ACM-SIGGRAPH)* 16: 61-67.
- Peitgen, H., Richter, P. (1984) *Die Unendliche Reise.* *Geo (German Edition)* 6: I 00- I 24.
- Peitgen, H., Richter, P. (1986) *The Beauty of Fractals.* Springer: Berlin.
- Saupe, D. (editors) (1988) *The Science of Fractal Images.* Springer: Berlin.
- Peterson, I. (1984) Ants in the labyrinth and other fractal excursions, *Science News.* 125: 42-43.
- Richardson, L. (1960) The problem of contiguity. In *Statistics of Deadly Quarrels.* (Wright, Q., Lienau, C., eds.) Boxwood Press: Pittsburgh.
- Robinson, A. (1985) Fractal fingers in viscous fluids. *Science* 228: 1077-1080.
- Roux, S., Guyon, E., Sornette, D. (1988) Hull percolation. *J. Phys. Math. Gen.* 2 1 : L47 5-L482.
- Schroeder, P. (1986) Plotting the Mandelbrot Set. *Byte* December 207-21 1 .
- Sorenson, P. (1984) Fractals. *Byte.* Sept. 9: I 57-172 (a fascinating introduction to the subject).
- Thomsen, D. (1982) A place in the sun for fractals. *Science News.* 121 : 28-32.
- Ushiki, S. (1988) Phoenix. *IEEE Trans. Circuits and Syst.* July 35(7): 788-789.
- Voss, R. (1985) Random fractal forgeries. In *Fundamental Algorithms in Computer Graphics.* R. Earnshaw, ed. Springer-Verlag: Berlin.

- Voss, R. (1988) Fractals in nature: from characterization to simulation. In *The Science of Fractal Images*, Peitgen, H., Saupe, D. (editors).
- West, S. (1984) The new realism. *Science* 84, 5: 31-39.
- West, B., Goldberger, A. (1987) Physiology in fractal dimensions. *Amer. Scien.* 75: 354-365. (This article describes the fractal characterization of the lungs's bronchial tree, the Weierstrass function, the fractal geometry of the heart, and "fractal time").
- Witten, T., Sander, L. (1981) *Phys. Rev. Lett.* 47: 1400.
- Aono, M., Kunii, L. (1984) Botanical Tree Image Generation. *IEEE Computer Graphics and Appl.* 4: 10-34.
- Bloomenthal, J. (1985) Modeling the mighty maple. *Computer Graphics (ACM SIGGRAPH)*. San Francisco. 19(3) 305-311.
- Braun, S. (1986) Botany with a twist. *Science* 86. 7: 63-64.
- Cohen, L. (1964) Random walk with transition probabilities that depend on the direction of motion. *Math. Magazine* 37(4): 248-250. (This work was used for designing plants).
- Dixon, R. (1983) The mathematics and computer graphics of spirals in plants. *Leonardo* 16: 86-90.
- Jena, R. (1984) *Mathematical Approach to Pattern and Form in Plant Growth*. John Wiley and Sons: New York.
- Kappraff, J. (1986) The geometry of coastlines. *Comp. and Maths. with Appls.* 1 2B: 655-671.
- Kappraff, J. (1986) A course in the mathematics of design. *Comp. and Maths. with Appls.* 12B: 913-948.
- Kawaguchi, Y. 1982. A morphological study of the form of nature. *SIGGRAPH July '82 Proc.*: Boston. 16(3): 223.
- Kolata, G. (1984) Esoteric math has practical result. *Science* 225: 494-495.
- Prusinkiewicz, P., Lindenmayer, A., Hanan, J. (1988) Developmental models of herbaceous plants for computer imagery purposes. *Computer Graphics (ACM-SIGGRAPH)* · August 22(4): 141-150.
- Reeves, W., Blau, R. (1985) Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Computer Graphics (ACM SIGGRAPH)* 19(3): 313-322.
- Rivlin, R. (1986) *The Algorithmic Image*. Microsoft Press, WA.
- Thompson, D. (1961) *On Growth and Form*. Cambridge: Cambridge, UK.
- Viennot, X., Eyrolles, G., Janey, N., Arques, D. (1989) Combinatorial analysis of ramified patterns and computer imagery of trees. *Computer Graphics (ACM-SIGGRAPH)* July 23(3): 31-40.
- Yessios, C. (1979) Computer drafting of stones, wood, plant and ground materials. *Computer Graphics (ACM-SIGGRAPH)* August '79 Proc.: Chicago 13(2): 190.
- Brown, D. (1987) Competition of cellular automata rules. *Complex Systems*. 1: 169-180.
- Conway, J., Berlekamp, E., Guy, R. (1982) *Winning Ways for Your Mathematical Plays*. Academic Press: New York.
- Levy, S. (1985) The portable universe: getting to the heart of the matter with cellular automata. *The Whole Earth Review Magazine*. Winter issue, 42-48.
- Maeder, D. (1987) The free energy concept in cellular automaton models of solid-solid phase transitions. *Complex Systems* 1: 131-144.
- Peterson, I. (1987) Forest fires, barnacles, and trickling oil. *Science News*. 132: 220-221.
- Poundstone, W. (1985) *The Recursive Universe*. William Morrow and Company: New York.
- Schrandt, R., S. Ulam (1970) On recursively defined geometrical objects and patterns of growth. In *Essays on Cellular Automata*, A. Burks, ed. Univ. of Illinois Press, Chicago.
- Wolfram, S. (1983) Statistical mechanics of cellular automata. *Rev. Modern Physics*. 55, 601-644.
- Abraham, R., Shaw, C. (1985) *Dynamics - The Geometry of Behavior, Part 3.-Global Behavior*. Aerial Press: California. (Actually, the entire book collection of Aerial Press, including the Visual Math Series, is an educational wonderland).
- Abraham, R., Shaw, C. (1983) *Dynamics - The Geometry of Behavior, Part 2.-Chaotic Behavior*. Aerial Press: California.
- Aronson, D., Chory, M., Hall, G. and McGehee, R. (1980) A discrete dynamical system with subtly wild behavior. In *New Approaches to Nonlinear Problems in Dynamics*, P.J. Holmes, ed. Soc. for Industry and Appl. Math.
- Berge, P., Pomeau, Y., Vidal, C. (1984) *Order Within Chaos* Wiley: New York.
- Campbell, D., Crutchfield, J., Farmer, D., Jen, E. (1985) Experimental mathematics: the role of computation in nonlinear science. *Commun. ACM*. 28: 374-389.
- Crutchfield, J., Farmer, J., Packard, N. (1986) *Chaos*. *Scien. Amer.* 255: 46-57;
- Devaney, R. (1986) Chaotic bursts in nonlinear dynamical systems. *Science* 235: 342-345.
- Dewdney, A. K. (1987) Probing the strange attractions of chaos. *Scien. Amer.* July Issue, 108-111.



- Collet, P., J.P. Eckmann. (1980) Iterated Maps on the Interval as Dynamical Systems. Birkhauser: Boston.
- Feigenbaum, M. (1979) The universal metric properties of nonlinear transformations. J. Statistical Physics. 21 : 669-706.
- Feigenbaum, M. (1981) Universal behavior in nonlinear systems. Los Alamos Science. I : 4-27.
- Finney R., D. Ostberg (1976) Elementary Differential Equations with Linear Algebra. Mass: Addison-Wesley.
- Fisher, A. ( 1985) Chaos: The Ultimate Asymmetry. Mosaic 16: 24-30.
- Fischer, P., Smith, W. (1985) Chaos, Fractals, and Dynamics. Marcel Dekker, Inc.: New York.
- Glass, L., Mackey, M. (1988) From Clocks to Chaos.' The Rhythms of Life Princeton Univ. Press: New Jersey.
- Hassell, M. (1974) Insect Populations. J. Anim. Ecol. 44: 283-296.
- Hofstadter, D. (1981) Strange Attractors. Scien. Amer. 245: 16-29.
- Hofstadter, D. (I 986) Metamagical Themes.' Question for the Essence of Mind and Pattern. Bantam: New York.
- Kudrewicz, J., Grudniewicz, J. and Swidzinske, B. (1986) Chaotic oscillation as a consequence of the phase slipping phenomenon in a discrete phase-locked loop. Proc. of the IEEE Int. Symp. on Circuits and Syst. I: 74-78.
- Levi, B. (1981) Period-doubling route to chaos shows universality. Physics Today. March, pp. 17-19.
- Lorenz, E. (1963) Deterministic nonperiodic flow. J. Atmos. Sci. 20: 130.
- May, R. (1976) Simple mathematical models with very complicated dynamics. Nature. 261 : 459-467.
- Moon, F. (1987) Chaotic Vibrations. John Wiley and Sons, New York.
- Nussbaum, R., Peitgen, O. (1984) Special and spurious solutions of  $f^{-1}(t) = -af(x(t-1))$ . Memoirs of the American Mathematical Society. 51 : 1- 129.
- Peterson, I. (I 986) Toying with a touch of chaos. Science News. 129: 277-278.
- Schwenk, T. (I 976) Sensitive Chaos. Schocken Books: New York.
- Shaw, A. ( 1984) The Dripping Faucet as a Model Chaotic System. Aerial Press: California.
- Sinanoglu, O. (1975) Theory of chemical reaction networks. J. Am. Chem. Soc. 97:2309-2320. (The Sinanoglu references are included here because they are helpful for "Pattern Formation and Chaos in Networks" on page 142.)
- Sinanoglu, O. (1981) 1- and 2-topology of reaction networks. J. Math. Phys. 22: I 504- I 5 1 2.
- Stoker, J. ( 1953) Mathematical methods in nonlinear vibration theory, In Proc. Symp. Nonlinear Circuit Analysis, Vol.2. Interscience Publishers: New York. 28-55.
- Taubes, G. (1984) The mathematics of chaos. Discover Magazine. September Issue, 30-45.
- Bidwell, J. (1973) Pascal's triangle revisited. Mathematics Teacher. 66: 448-452.
- Crandall, R. (1978) On the "3x+1" problem. Math. of Computation 32: 1 28 1 - I 292.
- Crypton, D. (1985), Prime numbers and the national security. Science Digest. Oct. 86-88.
- Dodge, C. (1969) Numbers and Mathematics. Prindle, Weber, and Schmidt: Boston (easy reading).
- Gardner, M. (1977) Pascal's triangle. In Mathematical Carnival. Vintage Books: New York.
- Garner, L. (1981) On the Collatz  $3n+1$  problem. Proc. Amer. Math. Soc. 82: 1 9-22.
- Gordon, J., Goldman, A., Maps, J. (1986) Superconducting-normal phase boundary of a fractal network in a magnetic field. Phys. Rev. Let. 56: 2280-2283.
- Hayes, B. (1984) Computer recreations: on the ups and downs of hailstone numbers. Scien. Amer. 250: 10-16.
- Holter, N., Lakhtakia, A., Varadan, V., Vasundara, V. Messier, R. (1986). On a new class of planar fractals: the Pascal-Sierpiński gaskets. J. Phys. A: Math. Gen. 19: 1753-1759.
- Jansson, L. (1973) Spaces, functions, polygons, and Pascal's triangle. Mathematics Teacher, 66: 71-77.
- Lakhtakia, A., Vasundara, V., Messier, R., Varadan, V. (1988) Fractal sequences derived from the self-similar extensions of the Sierpiński gasket. J. Phys. A: Math. Gen. 21 : 1925-1928.
- Legarias, J. (1985) The  $3x+1$  problem and its generalizations. Amer. Math. Monthly, January 3-23. Schroder, M. (1984) Number Theory in Science and Communication. Springer: New York (This book is recommended highly. An interesting book, by a fascinating author).
- Schroder, M. (I 982) A simple function. Math. Intelligencer. 4: I 58-161 .
- Stein, M., Ulam, S., Wells, M. (1964) A visual display of the distribution of primes. Math. Monthly 71:516-520.
- System Product Interpreter User's Guide, Release 4 (1984) IBM manual (SC24-5238-2).
- Trigg, C. (1982-83) Palindromic octagonal numerals. J. Recr. Math. 15(1): 4 1 -46.
- Usiskin, Z. (1973) Perfect square patterns in the Pascal triangle. Math. Magazine Sept.-Oct. 203-208.
- Wagon, S. ( 1985) The Colatz problem. Math. Intelligencer 7: 72-76.
- Wiggin, B. (1988) Wondrous numbers - conjecture about the  $3n+1$  family. J. Recr. Math. 20(I ): 52-56.
- Zhiqing, L. (I 985) Pascal's pyramid. Math. Spectrum I 7(1 ): 1-3.
- Benzinger, H., Burns, S., Palmore, J. (1987) Chaotic complex dynamics and Newton's method. Physics Letters A. I 19: 441-445.

- Grove, W. (1966) *Brief Numerical Methods*. Prentice-Hall: New Jersey (Has an excellent section on root-finding methods).
- Hamming, R. (1973) *Numerical/ Methods for Scientists and Engineers*. Dover Publications, New York.
- May, R. (1976) Simple mathematical models with very complicated dynamics. *Nature*. 261 : 459-467.
- Peterson, I. (1987) Zeroing in on chaos. *Science News* 131 : 137-139;
- Press, W. Flannery, B. Teukolsky, S., Vetterling, W. (1986) *Numerical Recipes*. Cambridge University Press: New York.
- Tritten, D. (1986) Chaos in the swing of a pendulum. *New Scientist*. July Issue: 37-40.
- Wimp J. (1984) *Computation with Recurrence Relations*. Pitman Publishing: Boston.
- A.V. Aho, J.E. Hopcroft, and J.D. Ullman: *Data Structures and Algorithms*. Addison-Wesley, Reading, Massachusetts. 1983.
- O-J. Dahl, B. Myrhaug, and K. Nygaard: *SIMULA Common Base Language*. Norwegian Computing Center S-22, Oslo, Norway. 1970.
- O-J. Dahl and C.A.R. Hoare: Hierarchical Program Construction in "Structured Programming." *Academic Press*, New York. 1972. pp 174-220 . 1
- A. Goldberg and D. Robson: *SMALLTALK-80 The Language and Its Implementation*. Addison Wesley, Reading, Massachusetts. 1983.
- R.E. Griswold et.al.: *The Snob014 Programming Language*. Prentice-Hall, Englewood Cliffs, New Jersey. 1970.
- R.E. Griswold and M.T. Griswold: *The ICON Programming Language*. Prentice-Hall, Englewood Cliffs, New Jersey. 1983. 17]
- Brian W. Kernighan and Dennis M. Ritchie: *The C Programming Language*. Prentice-Hall, Englewood Cliffs, New Jersey. 1978.
- George Orwell: 1984. Secker and Warburg, London. 1949.
- Martin Richards and Colin Whitby Stevens: *BCPL - The Language and Its Compiler*. Cambridge University Press. 1980.
- L. Rosler (Chairman. ANSI X3J1 I Language Subcommittee): Preliminary Draft Proposed Standard - The C Language. X3 Secretariat: Computer and Business Equipment Manufacturers Association, 311 First Street, NW, Suite 500, Washington, DC 20001 , USA.
- L. Rosler: The Evolution of C - Past and Future. *AT&T Bell Laboratories Technical Journal*. Vol.63 NO.8 Part 2. October 1984. pp 1685-1700.
- Ravi Sethi: Uniform Syntax for Type Expressions and Declarations. *Software Practice & Experience*, Vol.II (1981), pp 623-628.
- Bjarne Stroustrup: Adding Classes to C.' An Exercise in Language Evolution. *Software Practice & Experience*, 13 (1983), pp 139-61 .
- P.M. Woodward and S.G. Bond: *Algol 68-R Users Guide*. Her Majesty's Stationery Office, London. 1974.
- UNIX System V Release 2.0. *User's Reference Manual*. AT&T Bell Laboratories, Murray Hill, New Jersey. December 1983.
- UNIX Time-Sharing System, *Programmer's Manual*. Research Version, Eighth Edition. AT&T Bell Laboratories, Murray Hill, New Jersey. February 1985.
- UNIX Programmer's Manual. 4.2 Berkeley Software Distribution University of California, Berkeley, California. March 1984.
- *Human Interface Guidelines: The Apple Desktop Interface*. Addison-Wesley. 1988.
- Cox, Brad. *Object-Oriented Programming*. Addison-Wesley, 1986.
- Date, C. J. *An Introduction to Database Systems*, fourth edition. Addison-Wesley, 1986.
- *Smalltalk/V Tutorial and Handbook*. Digitalk, Inc., 1988.
- Ellis, Margaret, and Bjarne Stroustrup. *The Annotated C++ Reference Manual*. Addison-Wesley, 1990.
- Ezzell, Ben. *Object-Oriented Programming in Turbo Pascal 5.5*. Addison-Wesley. 1989.
- Fox, Edward A., editor. *Resources in Human-Computer Interaction*. ACM Press, 1990.
- Goldberg, Adele, and David Robson. *Smalltalk-80: The Language*, Addison-Wesley, 1989.
- Laurel, Brenda, editor. *The Art of Human-Computer Interaction*. Addison-Wesley, 1990.
- Kernighan, Brian, and Dennis Ritchie. *The C Programming Language*. second edition. Addison-Wesley. 1976.
- Meyer, Bertrand. *Object-Oriented Software Construction*. Prentice Hall. 1988.
- Norman, Don. *The Design of Everyday Things*. Doubleday, 1988.
- Parsaye, Kamran, Mark Chignell, Setrag Khoshafian, and Harry Wong. *Intelligent Data-bases*. Wiley, 1989.
- Rubin, Tony. *User Interface Design for Computer Systems*. John Wiley & Sons, 1988.
- Shneiderman, Ben. *Designing the User Interface*. Addison-Wesley, 1987.
- Tracz, Will. *Tutorial: Software Reuse: Emerging Technology*. IEEE, 1988.
- ANSI/MIL-STD-1815A. *Reference Manual for the Ada Programming Language*. Ada Joint Program Office, U.S.

Department of Defense, 1983.

- Auer, Ken. "Which Object-Oriented Language Should We Choose?" Hotline on Object-Oriented Technology. SIGS Publications, November 1989.
- Gomaa, Hassan. "Structuring Criteria for Real-Time System Design." ICSE '89 Proceedings. IEEE/ACM. 1989.
- Nicholson, Robert T. "Designing a Portable GUI Toolkit," Dr. Dobbs' Journal, January 1991.
- Norman, Don. "Why Interfaces Don't Work." The Art of Human-Computer Interaction, edited by Brenda Laurel. Addison-Wesley, 1990.
- Rheingold, Howard. "An Interview with Don Norman." The Art of Human-Computer Interaction. edited by Brenda Laurel. Addison-Wesley, 1990.
- Boehm, Barry. Software Engineering Economics. Prentice Hall, 1981.
- Bouldin, Barbara. Agents of Change. Prentice Hall, 1989.
- Cherry, George. Software Construction by Object-Oriented Pictures. Dorset House. 1990.
- DeMarco, Tom. Structured Analysis and System Specification. Prentice Hall, 1978.
- Fisher, Alan S. CASE: Using Software Development Tools. John Wiley & Sons, 1988.
- Gardarin, Georges, and Patrick Valduriez. Relational Databases and Knowledge Bases, Addison-Wesley, 1989.
- Gilb, Tom. Principals of Software Engineering Management. Addison-Wesley, 1988. Humphrey, Watts. Managing the Software Process. Addison-Wesley, 1989.
- McClure, Carma. CASE Is Software Automation. Prentice Hall, 1989.
- Connell, John L., and Linda Shafer. Structured Rapid Prototyping. Prentice Hall, 1989.
- "The Future Belongs to OOP." Apple Viewpoints. December 19. 1988.
- "The Power of Object-Oriented Programming." Apple Direct. February 1989.
- Banerjee, Jay. Hong-Tai Chou. Jorge Garza. Won Kim. Darrell Woelk. and Ballou Nat. "Data Model Issues for Object-Oriented Applications," ACM Transactions on Office Information Systems. January 1987.
- Blaha, Michael, William Premerlani, and James Rumbaugh. "Relational Database Design Using an Object-Oriented Methodology." ACM. April 1988.
- Bloom, Tony. and Stanley Zdonik. "Issues in the Design of Object-Oriented Database Programming Languages." ACM OOPSLA 87 Proceedings. October 1987.
- Boehm, Barry. "A Spiral Model of Software Development and Enhancement," IEEE Computer, May 1988.
- Boehm, Barry. "Understanding and Controlling Software Costs." IEEE Transactions on Software Engineering. October 1988.
- Booch, Grady. "Object-Oriented Development." IEEE Transactions on Software Engineering. February 1986.
- Brooks, Fred. "No Silver Bullet." IEEE Software. April 1988.
- Bruce, Thomas. "CASE Brought Down to Earth." Database & Programming Design. October 1988.
- "A New Era for Auto Quality." Business Week. October 22. 1990.
- Cattell, R. and T. Rogers. "Combining Object-Oriented and Relational Models of Data." IEEE International Workshop on Object-Oriented Database Systems Proceedings. 1986.
- Danforth, Scott. and Chris Tomlinson. "Type Theories and Object-Oriented Programming." March 1988. ACM Computing Surveys.
- Embley, David. and Scott Woodfield. "A Knowledge Structure for Reusing Abstract Data Types." Proceedings of the Ninth International Conference on Software Engineering. IEEE. 1987. Also in Tracz [1988a].
- Fischer, Gerhard "Human-Computer Interaction in Software: Lessons Learned, Challenges Ahead." IEEE Software. January 1989.
- Hull, Richard. and Roger King. "Semantic Database Modeling: Survey. Applications. and Research Issues," ACM Computing Surveys, September 1987.
- Ingalls, David. "Design Principles Behind Smalltalk." Byte. August 1981.
- Jacobsen, Ivar. "Object-Oriented Development in an Industrial Environment." ACM OOPSLA '87 Proceedings. October 1987.
- "Functional Refinement and Nested Objects for Object-Oriented Design." Engineering. March 1989. IEEE Transactions on Software
- Kang, Kyo. "Reuse-Based Development Methodology." Proceedings from the Workshop on Software Reusability and Maintainability. National Institute of Software Quality and Productivity. 1987. Also in Tracz [1988a].
- Kim, Ballou. Chou. Garza. Woelk. "Integrating an Object-Oriented Programming System with a Database System." October 1988. ACM OOPSLA '88 Proceedings.
- Ladden, Richard. "A Survey of Issues to Be Considered in the Development of an Object-Oriented Development Methodology for Ada," Ada Letters. March/April 1989.
- Lee, Rissman. D'Ippolito. Plinta. Scoy. "An OOD Paradigm for Flight Simulators," CMU-SEI Technical Report, second edition, September 1988.

- Loomis, M., A. Shah, and J. Rumbaugh. "An Object Modeling Technique for Conceptual Design." European Conference on OOP. June 1987.
- Lubars. Mitchell. "Wide-Spectrum Support for Software Reusability." Proceedings from the Workshop on Software Reusability and Maintainability, National Institute of Software Quality and Productivity, 1987. Also in Tracz [1988a]
- Lubars. Mitchell. "Code Reusability in the Large versus Code Reusability in the Small," Tutorial: Software Reuse: Emerging Technology. IEEE. 1988.
- Miller. G. A. "The Magical Number Seven. Plus or Minus Two: Some Limits on Our Capacity for Processing Information." Psychological Review. March 1963.
- Miller. G. A. "The Magical Number Seven after Fifteen Years," Studies in Long Term Memory, edited by A. Kennedy. Wiley, 1975.
- Millikin. Michael. "Object Orientation: What It Can Do for You." Computerworld. March 13. 1989.
- Norman. Ron. "Object-Oriented Analysis & Design." presented at the DPMA Information Technology Conference. San Diego. October 1990.
- Parnas, David. "On the Criteria for Decomposing Programs into Modules." Communications of the ACM. December 1972.
- Potter, William, and Robert Trueblood. "Traditional. Semantic. and Hyper Semantic Approaches to Data Modeling." IEEE Computer. June 1988.
- Ramamoorthy, C. V., and Phillip Sheu. "Object-Oriented Systems." IEEE Expert. Fall 1988.
- Rumbaugh, James. "Relations as Semantic Constructs in an Object-Oriented Language," ACM OOPSLA'87 Proceedings. October 1987.
- Seidewitz. Ed. "Object-Oriented Programming in Smalltalk and Ada." ACM OOPSLA '87 Proceedings. October 1987.
- Seidewitz, Ed, and Mike Stark. "Towards a General Object-Oriented Software Development Methodology." Ada Letters. vol. 7, no. 4.
- Simos. Mark. "The Domain-Oriented Lifecycle: Towards an Extended Process Model for Reusability." Proceedings from the Workshop on Software Reusability and Maintainability. National Institute of Software Quality and Productivity. 1987. Also in Tracz [1988a].
- Smith. Connie. "Applying Synthesis Principles to Create Responsive Software Systems." IEEE Transactions on Software Engineering. October 1988.
- Smith. Karen. and Stanley Zdonik. "Intermedia: A Case Study on the Differences between Relational and Object-Oriented Database Systems." ACM OOPSLA 87 Proceedings. September 1987.
- Stankovic. John. "Misconceptions about Real-Time Computing." IEEE Computer. October 1988.
- Stroustrup, Bjarne. "A Better C," Byte. August 1988.
- Stroustrup. Bjarne. "What Is Object-Oriented Programming?" IEEE Software. May 1988.
- Teorey. T.. D. Yang. and J. Fry. "A Logical Design Methodology for Relational Databases Using the Extended Entity Relationship Model." ACM Computing Surveys. June 1986.
- Thomas. Dave. "What's an Object?" Byte. March 1989.
- Tracz. Will. "Confessions of a Used Program Salesman." a series of seven articles from various issues of IEEE Software. reprinted in Tutorial: Software Reuse: Emerging Technology. IEEE. 1988.
- Tracz. Will. "RMISE Workshop on Software Reuse Meeting Summary." Tutorial: Software Reuse: Emerging Technology. IEEE. 1988. Also in Tracz [1988a].
- Tracz, Will. "Software Reuse: Motivators and Inhibitors," Proceedings of COMPCON. IEEE, Spring 1987. Also in Tracz [1988a]. Also in Tracz [1988a].
- Tracz. Will. "Software Reuse Myths." ACM SIGSOFT Software Engineering Notes. January 1988. Also in Tracz [1988a].
- Unland, R., and G. Schlageter. "An Object-Oriented Programming Environment for Advanced Database Applications," Journal of Object-Oriented Programming. May/ June 1989.
- Wegner. Peter. "Dimensions of Object-Based Language Design." ACM OOPSLA '87 Proceedings. October 1987.
- Zaniolo, Carlo. Hassan Ait-Kaci. David Beech. Stephanie Cammarata. Larry Kerschberg. and David Maier. "Object-Oriented Database Systems and Knowledge Systems." Expert Database Systems. edited by Larry Kerschberg. 1986.
- Berryman. Gregg. Notes on Graphic Design and Visual Communication. William Kaufmann. 1984.
- Encyclopaedia Britannica. Articles on "Behaviour. Animal." "Classification Theory." and "Mood." Encyclopaedia Britannica, Inc., 1986.
- IEEE Standard Glossary of Software Engineering Terminology (Standard 729). IEEE, 1983.
- Lanham, Richard A. Revising Business Prose. Charles Scribner's Sons, 1981 .

- McKenzie, E. C. 14,000 Quips and Quotes for Writers and Speakers, "Religion." Outlet Book Company, a Random House Company, New York. The "Religion" section is a great source of "method" oneliners. Just substitute the word religion for the word method, and chuckle away! (In a similar fashion, it's also a great source for programming language jokes.)
- Dictionary of Computing. Oxford University Press, 1986.
- Rosenau, Milton. Successful Project Management. Wadsworth, 1981 .
- Webster's New Twentieth-Century Dictionary, Collins World, 1977.
- Coad, Peter. Object-Oriented Analysis. Seminar notes. Object International, 1990.
- Coad, Peter. Object-Oriented Design. Seminar notes. Object International, 1990.
- Coad, Peter, and Edward Yourdon. Object-Oriented Analysis. second edition. Prentice Hall. 1991 .
- Yourdon, Edward. Managing the Structured Techniques, fourth edition. Prentice Hall, 1988.
- Yourdon, Edward. and Larry Constantine. Structured Design. Prentice Hall, 1979.
- Coad, Peter. "Analysis and Design Column," Coad's regular column in Journal of Object-Oriented Programming. SIGS Publications. 1991 .
- Coad, Peter. "OOA: Object-Oriented Analysis." American Programmer. Summer 1989.
- Coad, Peter. and Edward Yourdon. "OOA Object-Oriented Analysis." IEEE Tutorial on System and Software Requirements Engineering. 1990.
- Yourdon, Edward. "More on the Future of CASE," American Programmer. October 1988.
- Yourdon, Edward. "Sayonara. Structured Stuff." American Programmer. August 1988.
- Cherry, George, Software Construction by Object-Oriented Pictures. Dorset House. 1990.
- Cox, Brad, Object-Oriented Programming. Addison-Wesley, 1986.
- Smalltalk/V Tutorial and Hand book. Digitalk, Inc., 1988.
- Meyer, Bertrand, Object-Oriented Software Construction. Prentice Hall, 1988.
- Parsaye, Kamran; Chignell, Mark; Khoshafian, Setrag; and Wong, Harry, Intelligent Databases. Wiley, 1989.
- Shlaer, Sally and Mellor, Steve, Object-Oriented Systems Analysis. Prentice Hall, 1988.
- Tracz, Will, Tutorial: Software Reuse: Emerging Technology. IEEE, 1988.
- Blaha, Michael; Premerlani, William; and Rumbaugh, James. "Relational Database Design Using An Object-Oriented Methodology." Communications of the ACM. April 1988.
- Danforth, Scott and Tomlinson, Chris. "Type Theories and Object-Oriented Programming," ACM Computing Surveys. March 1988.
- Hull, Richard and King, Roger. "Semantic Database Modeling: Survey, Applications, and Research Issues." ACM Computing Surveys, September 1987.
- Kang, Kyo. "Reuse-Based Development Methodology." Proceedings from the Workshop on Software Reusability and Maintainability. National Institute of Software Quality and Productivity, 1987. Also in Tracz, 1988a.
- Ladden, Richard, "A Survey of Issues to be Considered in the Development of an Object-Oriented Development Methodology for Ada." Ada Letters. March/April 1989.
- "Entity Life Histories." Learmonth Burchett Management Systems, Houston and London, 1987.
- Loomis, M.; Shah, A.; and Rumbaugh, J., "An Object Modeling Technique for Conceptual Design." European Conference on OOP. June 1987.
- Lubars, Mitchell, "Code Reusability in the Large versus Code Reusability in the Small." Tutorial: Software Reuse: Emerging Technology. IEEE. 1988.
- Prieto-Dias, Ruben, "Domain Analysis for Reusability," Proceedings of COMPSAC '87. IEEE, 1987. Also in Tracz, 1988a.
- Rumbaugh, James. "Relations as Semantic Constructs in an Object-Oriented Language." ACM OOPSLA'87 Proceedings. October 1987.
- Seidewitz, Ed and Stark, Mike. "Towards a General Object-Oriented Software Development Methodology," Ada Letters, Volume 7. Number 4.
- Stankovic, John, "Misconceptions About Real Time Computing." IEEE Computer. October 1988.
- Teorey, T.; Yang, D.; and Fry, J., "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model." ACM Computing Surveys. June 1986.
- Thomas, Dave. "What's an Object?," Byte. March 1989.
- Tracz, Will. "Confessions of a Used Program Sales-man." a series of seven articles from various issues of IEEE Soft-ware, reprinted in Tutorial: Software Reuse: Emerging Technology, IEEE, 1988.
- Bouldin, Barbara. Agents of Change. Prentice Hall. 1989.
- DeMarco, Tom, Structured Analysis and System Specification. Prentice Hall. 1978.
- Fisher, Alan S.. CASE: Using Software Development Tools. John Wiley & Sons, 1988.
- Flavin, Matt, Fundamental Concepts of Information Modeling. Prentice-Hall. 1990.

- Gane, Chris and Sarson, Trish. *Structured Systems Analysis: Tools and Techniques*. Prentice Hall, 1979.
- Gardarin, Georges and Valduriez, Patrick. *Relational Databases and Knowledge Bases*. Addison-Wesley. 1989.
- Humphrey, Watts. *Managing the Software Process*. Addison-Wesley. 1989.
- McClure. Carma. *CASE Is Software Automation*. Prentice Hall. 1989.
- McMenamain. Steve and Palmer, John, *Essential Systems Analysis*. Prentice Hall. 1984.
- Scharbach, P. N., *Formal Methods: Theory and Practice*. CRC Press, 1989.
- "The Future Belongs to OOP." *Apple Viewpoints*. December 19. 1988.
- "The Power of Object-Oriented Programming." *Apple Direct*. February 1989.
- Banerjee. Jay; Chou. Hong-Tai; Garza, Jorge; Kim. Won; Woelk. Darrell; and Ballou. Nat. "Data Model Issues for Object-Oriented Applications." *ACM Transactions on Office Information Systems*. January 1987.
- Bloom. Tony and Zdonik. Stanley. "Issues in the Design of Object-Oriented Database Programming Languages." *ACM OOPSLA '87 Proceedings*. October 1987.
- Boehm. Barry. "Understanding and Controlling Software Costs," *IEEE Transactions on Software Engineering*. October 1988.
- Booch. Grady, "Object-Oriented Development," *IEEE Transactions on Software Engineering*. February 1986.
- Brooks, Fred, "No Silver Bullet," *IEEE Software*, April 1988.
- Bruce. Thomas, "CASE Brought Down to Earth," *Database & Programming Design*. October 1988.
- Cattell, R. and Rogers, T., "Combining Object-Oriented and Relational Models of Data." *IEEE International Workshop on Object-Oriented Database Systems Proceedings*. 1986.
- Chen. Peter. "The Entity Relationship Model Toward a Unified View of Data." *ACM Transactions on Database Systems*, March 1976.
- Embley, David and Woodfield. Scott, "A Knowledge Structure for Reusing Abstract Data Types." *Proceedings of the Ninth International Conference on Software Engineering, IEEE, 1987*. Also in Tracz, 1988a.
- Fischer. Gerhard. "Human-Computer Interaction in Software: Lessons Learned. Challenges Ahead," *IEEE Software*, January 1989.
- Gerrard, Christopher; Coleman. Gary; and Gallimore. Robin. "Formal Specification and Design Time Testing." *IEEE Transactions on Software Engineering*. January 1990.
- Ingalls, David. "Design Principles Behind Smalltalk." *Byte*. August 1981.
- Iscoe. Neil. "Domain-Specific Reuse: An Object-Oriented and Knowledge-Based Approach." *Tutorial: Software Reuse: Emerging Technology*. IEEE. 1988.
- Jacobsen, Ivar. "Object-Oriented Development in an Industrial Environment," *ACM OOPSLA '87 Proceedings*. October 1987.
- Jalote, Pankaj. "Functional Refinement and Nested Objects for Object-Oriented Design." *IEEE Transactions on Software Engineering*. March 1989.
- Katz, Shmuel; Richter, Charles; and The. Khe-Sing. "PARIS: A System for Reusing Partially Interpreted Schemas," *Proceedings of the Ninth International Conference on Software Engineering, IEEE, 1987*. Also in Tracz, 1988a.
- Kim. Won; Ballou. Nat; Chou. Hong-Tai; Garza, Jorge; and Woelk, Darrell, "Integrating an Object-Oriented Programming System with a Database System," *ACM OOPSLA '88 Proceedings*, October 1988.
- Lee, Rissman. D'Ippolito. Plinta. and Scoy, "An OOD Paradigm for Flight Simulators." *CMU-SEI Technical Report*. Second Edition. September 1988.
- Lubars. Mitchell. "Wide-Spectrum Support for Software Reusability." *Proceedings from the Workshop on Software Re-usability and Maintainability*. National Institute of Software Quality and Productivity. 1987. Also in Tracz. 1988a.
- Martin. Jim. "Paper: the Parent of Perfidious Practices," *Defense Science*, July 1988.
- Miller, G. A., "The magical number seven. plus or minus two: Some limits on our capacity for processing information," *Psychological Review*, March 1963.
- Miller, G. A., "The magic number seven after fifteen years," *Studies in Long Term Memory*. edited by A. Kennedy. Wiley, 1975.
- Millikin. Michael. "Object-Oriented: What It Can Do for You." *Computerworld*. March 13. 1989.
- Parnas, David. "On the Criteria for Decomposing Programs into Modules." *Communications of the ACM*. December 1972.
- Potter. William and Trueblood. Robert. "Traditional. Semantic, and Hyper-Semantic Approaches to Data Modeling." *IEEE Computer*. June 1988.
- Ramamoorthy. C. V.. and Sheu. Phillip. "Object-Oriented Systems." *IEEE Expert*. Fall 1988.
- Rumbaugh. James. "Controlling Propagation of Operations using Attributes on Relations." *ACM OOPSLA '88 Proceedings*. October 1988.

- Sanden, Bo, "An Entity-Life Modeling Approach to the Design of Concurrent Software." *Communications of the ACM*, March 1989.
- Sanden, Bo, "The Case for Eclectic Design of Real-Time Software." *IEEE Transactions on Software Engineering*, March 1989.
- Seidewitz, Ed. "Notes on Object-Oriented Analysis and Specification." unpublished notes. December 1988.
- Seidewitz, Ed. "Object-Oriented Programming in Smalltalk and Ada." *ACM OOPSLA '87 Proceedings*. October 1987.
- Simos, Mark. "The Domain-Oriented Lifecycle: Towards an Extended Process Model for Reusability." *Proceedings from the Workshop on Software Reusability and Maintainability*, National Institute of Software Quality and Productivity, 1987. Also in Tracz. 1988a.
- Smith, Connie. "Applying Synthesis Principles to Create Responsive Software Systems." *IEEE Transactions on Software Engineering*, October 1988.
- Smith, Karen and Zdonik, Stanley. "Intermedia: A Case Study on the Differences between Relational and Object-Oriented Database Systems," *ACM OOPSLA '87 Proceedings*. September 1987.
- Stroustrup, Bjarne. "A Better C," *Byte*, August 1988.
- Stroustrup, Bjarne. "What is Object-Oriented Programming?." *IEEE Software*. May 1988.
- Tracz, Will. "Reusability Comes of Age." *IEEE Software*. IEEE. July 1987. Also in Tracz. 1988a.
- Tracz, Will, "RMISE Workshop on Software Reuse Meeting Summary." *Tutorial: Software Reuse: Emerging Technology*. IEEE. 1988. Also in Tracz. 1988a.
- Tracz, Will. "Software Reuse: Motivators and Inhibitors," *Proceedings of COMPCON*, IEEE. Spring 1987. Also in Tracz. 1988a.
- Tracz, Will. "Software Reuse Myths." *ACM SIGSOFT Software Engineering Notes*, January 1988. Also in Tracz, 1988a.
- Unland, R. and Schlageter, G. "An Object-Oriented Programming Environment for Advanced Data-base Applications." *Journal of Object-Oriented Programming*, May/ June 1989. · Wing, J. M. and Nixon, M. R.. "Extending Ina JOTM with Temporal Logic." *IEEE Transactions on Software Engineering*, February 1989. (Ina Jo is a trademark of SDC, now a part of Unisys.)
- Wegner, Peter. "Dimensions of Object-Based Language Design." *ACM OOPSLA '87 Proceedings*. October 1987.
- Zaniolo, Carlo; Ait-Kaci, Hassan; Beech, David; Cammarata, Stephanie; Kerschberg, Larry; and Maier, David, "Object-Oriented Database Systems and Knowledge Systems." *Expert Database Systems*. edited by Larry Kerschberg. 1986.
- Berryman, Gregg. *Notes on Graphic Design and Visual Communication*. William Kaufmann. 1984.
- *Encyclopaedia Britannica*. Articles on "Behaviour. Animal." "Classification Theory." and "Mood." *Encyclopedia Britannica, Inc.*, 1986.
- *IEEE Standard Glossary of Software Engineering Terminology (Standard 729)*. IEEE. 1983.
- Lanham, Richard A., *Revising Business Prose*. Charles Scribner's Sons, 1981 .
- *Subjects*. Available in print and on CD-ROM. Library of Congress. 1990.
- *Dictionary of Computing*. Oxford University Press. 1986.
- Rosenau, Milton, *Successful Project Management*. Wadsworth, 1981 .
- Shertzer, Margaret. *The Elements of Grammar*. Macmillan, 1986.
- Strunk Jr., W. and White. E. B.. *The Elements of Style*. Macmillan, 1979.
- *Webster's New Twentieth Century Dictionary*. Collins World. 1977.
- Coad, Peter, *Object-Oriented Analysis*. Seminar notes. Object International (Austin, Texas USA), 1990.
- Coad, Peter, *Object-Oriented Design*. Seminar notes. Object International (Austin, Texas USA), 1990.
- Coad, Peter and Yourdon, Edward, *Object-Oriented Design*. Prentice Hall, 1991 .
- Yourdon, Edward, *Managing the Structured Techniques*. Fourth Edition. Prentice Hall, 1988.
- Yourdon, Edward. *Modern Structured Analysis*. Prentice Hall, 1989.
- Coad, Peter. "OOA: Object-Oriented Analysis." *American Programmer*. Summer 1989.
- Coad, Peter and Yourdon, Edward. "OOA Object-Oriented Analysis." *IEEE Tutorial on System and Software Requirements Engineering*. 1990.
- "More on the Future of CASE." *American Programmer*. October 1988.
- "Sayonara, Structured Stuff," *American Programmer*. August 1988.
- PNC PJ1406 97-008, ベタフロップス・コンピューティングに係わる動向調査、1997年4月
- Wylie B, Cameron G, White M, Smith M, McArthur D. 1993, PARAMICS: Parallel Microscopic Traffic Simulator. *Proceedings of the Second European Connection Machine Users Conference*, Paris, October.
- Cameron G, Smith M., White, M, Wylie B, McArthur D. 1994, The PARAMICS Parallel Microscopic Traffic

Simulator. Proceedings of the JFIT UK IT Forum, Edinburgh.

- Yau, H.W. Cliffe, K.A. Sinclair, J.E. Sumner P.J., 1994, The Parallelisation of the AEA Probabilistic Safety Assessment Program, MASCOT. Proceedings of HPCN '94 Munich.
- Baxter, R.M. Murphy, K.D. Trewin S.M. 1996, Experiences in Parallelising FLITE3D on the Cray T3D.. Concurrency: Practice and Experience, vol 8(10).
- Booch, 1994, Object-Oriented Analysis and Design with Applications (Second Edition). The Benjamin/Cummings Publishing Company, Inc.
- National Science foundation. Grand Challenge : High-Performance Computing and Communications. Report, Committee on Physical, Mathematical, and Engineering Sciences, Washington, D.C., U.S. Office of science and technology Policy, 1992.
- Hwang, K. 1993, Advanced Computer Architecture : Parallelism, Scalability, Programmability, McGraw-Hill Series in Computer Science. McGraw-Hill Inc.
- Buschmann, F., Meurnier, R., Rohnert, H., Sommerlad, P. & Stal, M. 1996, A System of Patterns : Pattern-Oriented Software Architecture. John Wiley & Sons.
- Cook, S. & Daniels, J. 1994, Designing Object Systems : Object-Oriented Modelling with Syntropy. Prentice Hall.
- Coplien, J.O. 1992, Advanced C++ : Programming Styles and Idioms. Addison-Wesley.
- Gamma, E., Helm, R., Johnson R. & Vlissides, J. 1994, Design Patterns : Elements of Reusable Object-Oriented Software. Addison-Wesley.
- Object Management Group. 1992, The Common Object Request Broker : Architecture and Specification. OMG Document Number 91.12.1. Revision 1.1.
- Orfali, R., Harkey, D. & Edwards J. 1996, The Essential Distributed Objects Survival Guide. John Wiley & Sons.
- Zhao, L. & Foster, E. A 1997, Pattern Language of Transport Systems (Point and Route). Accepted for publication in : Pattern Languages of Program Design 3. Editors: Riehle, D., Buschmann, F., & Martin, R.C. Addison-Wesley.
- G.A. 1986, Actors : a model of concurrent computation in distributed systems. Cambridge, Mass. : MIT Press.
- Caromel, D. 1993, Toward a Method of Object-Oriented Concurrent Programming. Communications of the ACM. Vol. 36., No. 9. September.
- Coad, P. with North, D. & Mayfield, M. 1995, Object Models : Strategies, Patterns, & Applications. Yourdon Press.
- Coplien J.O. 1996, Software Patterns : A White Paper. SIGS Publications.
- Ewald, A. & Roy, M. 1997, Are IDL Interfaces Good Business Object Interfaces? Object magazine 6(11).
- Fowler, M. 1996, Analysis Patterns : Reusable Object Models. Addison-Wesley.
- Hay, D.C. 1996, Data Model Patterns : Conventions of Thought. Dorset House Publishing.
- Hennessey, P., Scheid, R. & Kirkley III, R. 1996, An integration framework for distributed systems. Object Magazine 6(1). March
- Jacobson, I., Bylund, S., Jonsson, P. & Ehneboom, S. 1995, Using contracts and plugs to build pluggable architectures. Journal of Object-oriented Programming Vol. 8, No. 2.
- Meyer, B. 1993, Systematic Concurrent Object-Oriented Programming. Communications of the ACM. Vol.36, No.9.
- Object Management Group. 1995, CORBA services: Common Object Services Specification. OMG Document Number 95-3-31.
- Editors : Coplien, J.O. & Schmidt, D.C. 1995, Pattern Languages of Program Design. Addison-Wesley.
- Editors: Vlissides, J.M., Coplien J.O. & Kerth, N.L. 1996, Pattern Languages of Program Design 2. Section 7 : Concurrent Programming / Distributed Systems. Addison-Wesley.
- Pree, W. 1995, Design patterns for Object-Oriented Software Development. Addison-Wesley.
- Stubbs, S.S., Carver, D.L., & Hoppe, A. 1995, IPCC+: A concurrent C++ based on a shared memory model. Journal of Object-oriented Programming Vol. 8, No. 2.
- Wirfs-Brock, R., Wilkerson, B. & Wiener, L. 1990, Designing Object-oriented Software. Prentice Hall.
- V. Ambriola, P. Ciancarini, and M. Danelutto. Design and distributed implementation of the parallel logic language Shared Prolog. In Proceedings Second ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pages 40-49, Seattle, March 1990.
- Ashcraft, N. Carriero, and D. Gelernter. Is explicit parallelism natural? Hybrid db search and sparse ldl factorization using Linda. Research report, Yale University Department of Computer Science, January 1989.
- S. Ahuja, N. Carriero, D. Gelernter, and V. Krishnaswamy. Matching language and hardware for parallel computation in the Linda machine. IEEE Transaction on Computers, 37(8):921-929, August 1988.



- Simulator. Proceedings of the JFIT UK IT Forum, Edinburgh.
- Yau, H.W. Cliffe, K.A. Sinclair, J.E. Sumner P.J., 1994, The Parallelisation of the AEA Probabilistic Safety Assessment Program, MASCOT. Proceedings of HPCN '94 Munich.
  - Baxter, R.M. Murphy, K.D. Trewin S.M. 1996, Experiences in Parallelising FLITE3D on the Cray T3D.. Concurrency: Practice and Experience, vol 8(10).
  - Booch, 1994, Object-Oriented Analysis and Design with Applications (Second Edition). The Benjamin/Cummings Publishing Company, Inc.
  - National Science foundation. Grand Challenge : High-Performance Computing and Communications. Report, Committee on Physical, Mathematical, and Engineering Sciences, Washington, D.C., U.S. Office of science and technology Policy, 1992.
  - Hwang, K. 1993, Advanced Computer Architecture : Parallelism, Scalability, Programmability, McGraw-Hill Series in Computer Science. McGraw-Hill Inc.
  - Buschmann, F., Meurnier, R., Rohnert, H., Sommerlad, P. & Stal, M. 1996, A System of Patterns : Pattern-Oriented Software Architecture. John Wiley & Sons.
  - Cook, S. & Daniels, J. 1994, Designing Object Systems : Object-Oriented Modelling with Syntropy. Prentice Hall.
  - Coplien, J.O. 1992, Advanced C++ : Programming Styles and Idioms. Addison-Wesley.
  - Gamma, E., Helm, R., Johnson R. & Vlissides, J. 1994, Design Patterns : Elements of Reusable Object-Oriented Software. Addison-Wesley.
  - Object Management Group. 1992, The Common Object Request Broker : Architecture and Specification. OMG Document Number 91.12.1. Revision 1.1.
  - Orfali, R., Harkey, D. & Edwards J. 1996, The Essential Distributed Objects Survival Guide. John Wiley & Sons.
  - Zhao, L. & Foster, E. A 1997, Pattern Language of Transport Systems (Point and Route). Accepted for publication in : Pattern Languages of Program Design 3. Editors: Riehle, D., Buschmann, F., & Martin, R.C. Addison-Wesley.
  - G.A. 1986, Actors : a model of concurrent computation in distributed systems. Cambridge, Mass. : MIT Press.
  - Caromel, D. 1993, Toward a Method of Object-Oriented Concurrent Programming. Communications of the ACM. Vol. 36., No. 9. September.
  - Coad, P. with North, D. & Mayfield, M. 1995, Object Models : Strategies, Patterns, & Applications. Yourdon Press.
  - Coplien J.O. 1996, Software Patterns : A White Paper. SIGS Publications.
  - Ewald, A. & Roy, M. 1997, Are IDL Interfaces Good Business Object Interfaces? Object magazine 6(11).
  - Fowler, M. 1996, Analysis Patterns : Reusable Object Models. Addison-Wesley.
  - Hay, D.C. 1996, Data Model Patterns : Conventions of Thought. Dorset House Publishing.
  - Hennessey, P., Scheid, R. & Kirkley III, R. 1996, An integration framework for distributed systems. Object Magazine 6(1). March
  - Jacobson, I., Bylund, S., Jonsson, P. & Ehneboom, S. 1995, Using contracts and plugs to build pluggable architectures. Journal of Object-oriented Programming Vol. 8, No. 2.
  - Meyer, B. 1993, Systematic Concurrent Object-Oriented Programming. Communications of the ACM. Vol.36, No.9.
  - Object Management Group. 1995, CORBA services: Common Object Services Specification. OMG Document Number 95-3-31.
  - Editors : Coplien, J.O. & Schmidt, D.C. 1995, Pattern Languages of Program Design. Addison-Wesley.
  - Editors: Vlissides, J.M., Coplien J.O. & Kerth, N.L. 1996, Pattern Languages of Program Design 2. Section 7 : Concurrent Programming / Distributed Systems. Addison-Wesley.
  - Pree, W. 1995, Design patterns for Object-Oriented Software Development. Addison-Wesley.
  - Stubbs, S.S., Carver, D.L., & Hoppe, A. 1995, IPCC+: A concurrent C++ based on a shared memory model. Journal of Object-oriented Programming Vol. 8, No. 2.
  - Wirfs-Brock, R., Wilkerson, B. & Wiener, L. 1990, Designing Object-oriented Software. Prentice Hall.
  - V. Ambriola, P. Ciancarini, and M. Danelutto. Design and distributed implementation of the parallel logic language Shared Prolog. In Proceedings Second ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pages 40-49, Seattle, March 1990.
  - Ashcraft, N. Carriero, and D. Gelemtter. Is explicit parallelism natural? Hybrid db search and sparse ldt factorization using Linda. Research report, Yale University Department of Computer Science, January 1989.
  - S. Ahuja, N. Carriero, D. Gelemtter, and V. Krishnaswamy. Matching language and hardware for parallel computation in the Linda machine. IEEE Transaction on Computers, 37(8):921-929, August 1988.

- United States Department of Defense. Reference Manual for the Ada Programming Language, July 1982.
- G. Almasi and A. Gottlieb. Highly Parallel Computing. Benjamin/Cummings, Redwood City, CA, 1989.
- R. Abarbanel and A. Janin. Distributed object management with Linda. Research report, Apple Computer, Cupertino, CA, September 1989.
- M. Ben-Ari. Principles of Concurrent and Distributed Programming. Prentice-Hall, Hertfordshire, U.K., 1990.
- R. Babb. Parallel processing with large grain data flow techniques. IEEE Computer, 17:55-61, 1984.
- R. Bjornson, N. Carriero, D. Gelernter, and J. Leichter. Linda, the portable parallel. Research Report 520, Yale University Department of Computer Science, January 1988.
- G. Birtwistle, O. Dahl, B. Myhrhaug, and K. Nygaard. Simula Begin. Chartwell-Bratt, London, 1979.
- P. Bercovitz. TupleScope user's guide. Research report, Yale University Department of Computer Science, 1990.
- L. Borrmann, M. Herdieckerhoff, and A. Klein. Tuple space integrated into Modula-2, implementation of the Linda concept on a hierarchical multiprocessor. In Jesshope and Reinartz, editors, Proc. CONPAR '88. Cambridge University Press, 1988.
- A. Birrel and B. Nelson. Implementing remote procedure calls. ACM Transactions on Computing Systems, 2(1):39-59, February 1984.
- P. Brinch Hansen. The Programming Language Concurrent Pascal. IEEE Transaction on Software Engineering, 1(2):199-206, 1975.
- J. Browne. Software engineering of parallel programs in a computationally oriented display environment. In D. Gelernter, A. Nicolau, and D. Padua, editors, Languages and Compilers for Parallel Computing, pages 75-94. MIT Press, Cambridge, 1990.
- H. Bal, J. Stenier, and A. Tanenbaum. Programming Languages for Distributed Computing Systems. ACM Computing Surveys, 21(3):261-322, September 1989.
- H. Bal and A. Tanenbaum. Orca: A language for distributed objectbased programming. IR 140, Vrije Universiteit, Amsterdam, December 1987.
- N. Carriero. Implementing Tuple Space Machines. PhD thesis, Yale University Department of Computer Science, New Haven, Connecticut, 1987. Department of Computer Science.
- N. Carriero and D. Gelernter. Linda in context. Communications of the ACM, 32(4):444-458, April 1989.
- N. Carriero and D. Gelernter. Tuple analysis and partial evaluation strategies in the Linda pre-compiler. In D. Gelernter, A. Nicolau, and D. Padua, editors, Languages and Compilers for Parallel Computing, pages 114-125. MIT Press, Cambridge, 1990.
- N. Carriero, D. Gelernter, and J. Leichter. Distributed data structures in Linda. In Thirteenth ACM Symposium on Principles of Programming Languages Conf., pages 236-242, St. Petersburg, Florida, January 1986. Association for Computing Machinery.
- M. C. Chen. A parallel language and its compilation to multiprocessor machines or VLSI. In Thirteenth Annual ACM Symposium on Principles of Programming Languages, pages 131-139. Association for Computing Machinery, January 1986.
- K. M. Chandy and J. Misra. Parallel Program Design: A Foundation. Addison-Wesley, Reading, Mass., 1988.
- D. Comer. Operating System Design: The Xinu Approach. Prentice-Hall, Englewood Cliffs, NJ, 1984.
- D. Cheriton and W. Zwaenpoel. Distributed process groups in the V Kernel. ACM Transactions on Computer Systems, 3(2):77-107, May 1985.
- W. Dally. Object-oriented concurrent programming in CST. In Proc. Third Conf on Hypercube Concurrent Computers and Applications, page 33, 1988.
- P. Deane. The First Industrial Revolution. Cambridge University Press, Cambridge, U.K., 1969.
- J. Dongarra, D. Sorenson, and P. Brewer. Tools and methodology for programming parallel processors. In M. Wright, editor, Aspects of Computation on A synchronous Processors, pages 125-138. North-Holland, 1988.
- M. Factor. The Process Trellis Software Architecture for Parallel, Real-Time Monitors. PhD thesis, Yale University Department of Computer Science, 1990. In preparation.
- I. Foster and S. Taylor. Strand: New Concepts in Parallel Programming. Prentice-Hall, Englewood Cliffs, N.J., 1989.
- D. Gelernter, N. Carriero, S. Chandran, and S. Chang. Parallel programming in Linda. In International Conference on Parallel Processing, pages 255-263, August 1985.
- P. Gilmore. Massive parallel processor (MPP): Phase one final report. Technical Report GER-16684, Goodyear Aerospace Co., Akron, 1979.
- D. Gelernter and S. Jagannathan. Programming Linguistics: A first course in the design and evolution of programming languages. MIT Press, Cambridge, 1990.
- D. Gelernter, S. Jagannathan, and T. London. Environments as first-class objects. In Fourteenth ACM Symposium on Principle of Programming Languages Conf., January 1987.

- L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325-348, 1987.
- R. Halstead. Multilisp: A language for concurrent symbolic computation. *Transactions on Programming Languages and Systems*, 7(4):501-538, October 1985.
- P. Henderson. Purely functional operating systems. In J. Darlington, P. Henderson, and D. Turner, editors, *Functional Programming and its Applications*, pages 177-192. Cambridge University Press, 1982.
- P. Horowitz and W. Hill. *The Art of Electronics*. Cambridge Press, 1980.
- C. A. R. Hoare. Monitors: An operating system structuring concept. *Communications of the ACM*, 17(10):549-557, October 1974.
- C. A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):667-677, August 1978.
- W. D. Hillis and G. Steele Jr. Data parallel algorithms. *Communications of the ACM*, 29(12):1170-1183, December 1986.
- E. Jul, H. Levy, N. Hutchinson, and A. Black. Fine-grained mobility in the emerald system. *ACM Transactions on Computer Systems*, 6(1):109-133, Feb. 1988.
- H. Jordan. Structuring parallel algorithms in an MIMD, shared memory environment. *Parallel Computing*, 3:93-110, 1986.
- G. Kahn. The semantics of a simple language for parallel processing. In *Proceedings IFIP Congress*, page 471, 1974.
- L. Kale. The design philosophy of the Chare kernel parallel programming system. Technical Report UIUCDCS-R-89-1555, University of Illinois at Urbana-Champaign Department of Computer Science, 1989.
- H. Kung and C. Leiserson. Systolic arrays (for VLSI). In I. Duff and G. Stewart, editors, *Sparse Matrix Proceedings 1978*, pages 256-282. SIAM, 1979.
- W. Leler. Linda meets Unix. *IEEE Computer*, 23(2):43-55, Feb. 1990.
- B. Lampson and D. Redell. Experience with processes and monitors in Mesa. *Communications of the ACM*, 23(2):105-117, Feb. 1980.
- C. Lee, S. Skedzielewski, and J. Feo. On the implementation of applicative languages on shared-memory, MIMD multiprocessors. In *Proceedings of the ACM/SIGPLAN Symposium on Parallel Programming*, Aug. 1988.
- T. Malone. What is coordination theory? Working Paper 182, MIT Center for Information Systems Research, Feb. 1988.
- D. May. OCCAM. *ACM SIGPLAN Notices*, 18(4):69-79, April 1983.
- T. Marsland and M. Campbell. Parallel search of strongly ordered game trees. *ACM Computing Surveys*, 14(4):533-552, Dec. 1982.
- S. Matsuoka and S. Kawai. Using tuple space communication in distributed object-oriented languages. In *Proceedings OOPSLA '88*, pages 276-284, Nov. 1988.
- S. Mullender and A. Tanenbaum. The design of a capability-based distributed operating system. *The Computer Journal*, 29(4):289-300, Mar. 1986.
- J. Narem. Db: A parallel news database in Linda. Technical memo, Yale University Department of Computer Science, Aug. 1988.
- R. Nikhil, K. Pingali, and Arvind. Id Nouveau. Memo 265, MIT Computation Structures Group, 1986.
- T. Olson. Finding lines with the Hough Transform on the BBN Butterfly parallel processor. Report 10, University of Rochester, Department of Computer Science, Butterfly Project, Sept. 1986.
- G. Ringwood. Parlog86 and the dining logicians. *Communications of the ACM*, 31(1):10-25, Jan. 1988.
- C. Seitz. The Cosmic Cube. *Communications of the ACM*, 28(1):22-33, 1985.
- E. Shapiro, editor. *Concurrent Prolog Collected Papers*, volume 1 and 2. MIT Press, Cambridge, Mass., 1987.
- L. Snyder. The XYZ abstraction levels of poker-like languages. In D. Gelernter, A. Nicolau, and D. Padua, editors, *Languages and Compilers for Parallel Computing*, pages 470-489. MIT Press, Cambridge, 1990.
- A. Tanenbaum. *Operating Systems: Design and Implementation*. Prentice-Hall, Englewood Cliffs, N.J., 1987.
- S. Ward and R. Halstead. *Computation Structures*. MIT Press, Cambridge, 1990.
- N. Wiener. *Cybernetics*. John Wiley and Sons, New York, 1948.
- N. Wirth. Modula: A language for modular multiprogramming. *Software-Practice and Experience*, 7:3-35, 1977.
- R. Whiteside and J. Leichter. Using Linda for supercomputing on a local area network. In *Proceedings of Supercomputing 88*, 1988.
- M. Young. The duality of memory and communication in the implementation of a multiprocessor operating system. In *Proceedings of the Eleventh ACM Symposium on Operating Systems Principles*, pages 63-76, Nov. 1987.

- Selim, G. Akl: The Design and Analysis of Parallel Algorithm, Prentice Hall (1989)
- 渡辺勝正: アルゴリズムと計算機械, pp. 65~68, 93~94, 日本コンピュータ協会 (1978)
- 丸岡章: “セル構造オートマトンとハードウェアアルゴリズム”, 情報処理, 26, 6, pp.622~631 (June, 1985)
- 小林孝次郎: “2次元の一斉射撃問題”, 数理科学, 23, 7 (通巻265), pp.36~42 (July, 1985)
- 高橋馨郎, 早迫完一: “自己増殖オートマトンの設計と試作”, 情報処理学会論文誌, 31, 2, pp. 238~248 (Feb., 1990)
- 吉田紀彦: “シストリックアルゴリズムの関係型表現”, 日本ソフトウェア科学会第3回大会論文集, D-2-1 (1986)
- 吉田典可, 菊野亨, 藤井隆志: “シストリック配列とバス付セル配列上のハードウェアアルゴリズム”, 情報処理, 26, 6, pp.613~621 (June, 1985)
- ハーマン, H. ゴールドスタイン (未包良太, 米口肇, 大代茂之訳): 計算機の歴史, pp. 346~348, 共立出版 (1979)
- 村岡洋一: “並列処理技術の動向”, bit臨時増刊, 21, 4, pp.6~15 (1989)
- 栗田昭平: 90年代のコンピュータウォーズ, 日刊工業新聞社 (1988)
- 石田晴久, 村田健郎: 超大型コンピュータシステム, 産業図書 (1975)
- 山田博: コンピュータアーキテクチャ, 産業図書 (1976)
- George, S. Almasi, Allan Gottlieb: Highly Parallel Computing, The Benjamin/Cummings Publishing Company (1989)
- Wulf, W. A., Cohen, E., Corwin, W., Jones, A., Levin, R., Pierson, C., Pollack, F.: "Hydra: The Kernel of Multiprocessor Operating System". Comm. of ACM, 17, 6, pp.337~345 (June, 1974)
- “Elxsi System 6400 System Introduction”, Elxsi (1984)
- “Elxsi System 6400 System Architecture”, Elxsi (1983)
- “Balance 8000 Guide to Parallel Programming”, Sequent Computer Systems, inc. (1985)
- “並列コンピュータアーキテクチャ”, bit臨時増刊, 21, 4 (1989)
- “並列処理ハードウェアと言語-特集”, 電子情報通信学会論文誌, J71-D, 8 (Aug., 1988)
- Gordon, Bell: “The Future of High Performance Computers in Science and Engineering”, Comm. of ACM, 32, 9, pp.1091~1101 (Sep., 1989)
- 小池汎平, 田中英彦: “並列推論エンジン PIE64”, bit臨時増刊, 21, 4, pp.126~135 (1989)
- 濱崎陽一, 岡田義郎, 鈴木基史: “光電子コンピュータ: Dialog システム”, bit臨時増刊, 21, 4, pp.169~174 (1989)
- Michael, J. Flynn: “Some Computer Organizations and Their Effectiveness”, IEEE Trans. on Computer, C-21, 9, pp.948~960 (Sep., 1972)
- 小野洋彦: “VLIM マシンとそのコンパイラ”, インタフェース, 150, pp.259~266 (Nov., 1989)
- 富田真治: 並列計算機構成論, p.35, 昭晃堂 (1986)
- Kogge, P.M.: The Architecture of pipelined Computers, p.6, McGraw-Hill (1981)
- Colwell, R. P.: “AVLIW Architecture for a Trace Scheduling Compiler”, IEEE Trans. on Computer, C-37, 8, pp.967~979 (Aug., 1988)
- 徳田英幸: “分散/並列 OS: Mach”, bit臨時増刊, 21, 4, pp.34~40 (1989)
- Myers, G.J. (渡辺勝正訳): コンピュータアーキテクチャの設計, pp.108~165, 共立出版 (1981)
- Judy, M. Anderson, et al.: “The Architecture of FAIM-1”, IEEE Computer, 20, 1, pp.55~65 (Jan., 1987)
- Andrew, S. Tanenbaum (大西照代訳): MINIXオペレーティングシステム, アスキー (1989)
- K. Mani Chandy, Jayadev Misra: Parallel Program Design A Foundation. Addison-Wesley (1988)
- 陣崎明, 樋口昌宏: “ネットワーク仮想記憶システムにおける宣言的アクセス制御方式”, 情報処理学会論文誌, 30, 12, pp.1612~1619 (Dec., 1989)
- 白井良明, 辻井潤一: 人工知能, 岩波情報科学講座22, pp.155~160, 岩波書店 (1982)

- 宮澤治編：VLSI用コンピュータアーキテクチャ、昭晃堂（1989）
- 高橋義造編：並列処理機構、丸善（1989）
- Arthur, H. Venn（末吉敏則、富田眞治訳）：“データフロー計算機”、bit別冊 コンピュータサイエンス'86、pp.173～203、共立出版（1988）
- 島田俊夫、平木敬、関口智嗣：“科学技術計算用データフロースーパーコンピュータ SIGMA-1”、bit臨時増刊、21、4、pp.73～80（1989）
- 田中英彦：非ノイマン型コンピュータ、電子情報通信学会（1989）
- 古川康一、溝口文雄編：並列論理型言語 GHC とその応用、共立出版（1987）
- 笈捷彦：“並列処理言語”、bit臨時増刊、21、4、pp.24～33（1989）
- 米澤明憲：“オブジェクト指向計算の現状と展望”、情報処理、29、4、pp.290～294（Apr., 1988）
- フィルマン、フリードマン（雨宮真人、尾内理紀夫、高橋直久訳）：協調型計算システム、pp.167～179、マグロウヒルブック（1986）
- Hewitt, C. e.：“Viewing Control Structures as Patterns of Passing Messages”、Artif. Intell., 8、3、pp.324～364（June, 1977）
- 竹内彰一：“オブジェクト指向の指向するもの”、情報処理、29、4、pp.295～302（Apr., 1988）
- 米澤明憲：“並列オブジェクト指向言語 ABCL/1 による並列処理記述とその枠組の研究”、電子情報通信学会論文誌、J71-D、8、pp.1415～1422（Aug., 1988）
- 丸山勝己、渡部信幸：“既存並列処理言語による実時間オブジェクト指向プログラミング”、情報処理学会論文誌、31、1、pp.88～97（Jan., 1990）
- 渡辺慎哉、原田康徳、三谷和夫、宮本衛市：“場とイベントによる並列計算モデル Kamui88”、コンピュータソフトウェア、6、1、pp.41～55（Jan., 1989）
- 渡辺勝正、都司達夫：“Combination of the Procedural Language and Production Language”、福井大学工学部研究報告、33、1、pp.9～22（1985）
- 渡辺勝正、都司土達夫：“Some Types of parallel Algorithms”、福井大学工学部研究報告、34、1、pp.121～138（1986）
- Hockney. R. W., Josshope, CR（奥川峻史、黒住祥祐訳）：並列計算機、共立出版（1984）
- 福井義成：“汎用マルチプロセッサシステムによる並列処理”、情報処理、28、11、pp.1469～1475（Nov., 1987）
- Garry Rodrigue, ed: Parallel Computations, Academic Press, p.329（1982）
- 佐藤誠：“最適化問題の並列処理化の一方法-双対単体法”、電子通信学会論文誌、J69-D、5、pp.661～666（May, 1986）
- Falkoff, A. D., Iverson, K. E., Sussenguth, E. H.: “A Formal Description of System/360”、IBM journal, 3、3、pp.193～262（1964）
- 長田純一、内山昭：“APL SV-新しいコンピュータ言語”、丸善（1975）
- 山本哲朗、古金卯太郎、野倉久美：“代数方程式を解く Durand-Kerner 法と Aberth 法”、情報処理、18、6、pp.566～571（June, 1977）
- 渡辺勝正、都司達夫：“部分主導型並列アルゴリズムについて”、福井大学工学部研究報告、36、1、pp.9～24（March, 1988）
- Paul Kelly: Functional Programming for Loosely-coupled Multiprocessors, Pitman（1989）
- 前田明：“マルチベクトルプロセッサ VPP における並列処理方式”、bit臨時増刊、21、4、pp.219～224（1989）
- 村岡洋一：“並列処理”、情報処理、20、4、pp.289～294（Apr., 1979）
- 山内幸治、石川聖二、加藤清史：“分散制御方式を用いたロボットマニピュレータ”、日本ロボット学会誌、8、1、pp.44～47（Feb., 1990）
- Jon Kerridge, Dan Simpson: “Communicating Parallel Processes”、Software Practice & Experience, 16、

- 1, pp.63~86 (Jan.,1986)
- 近藤利夫、杉山吉、中島孝利：“2次元アレイプロセッサ(AAP2)とプログラミング言語”、電子情報通信学会論文誌、J71-D、8、pp.1399~1406 (Aug.,1988)
  - 近藤利夫、中島孝利、渡辺琢美、杉山吉：“1ビット2次元SIMDプロセッサAAP”、bit臨時増刊、21、4、pp.202~212 (Mar.,1989)
  - 渡辺勝正、榎本好晴、郡司達夫：“非手続き的な表現の一つとしての条件式のPascalへの導入”、情報処理学会論文誌、25、6、pp.980~989 (June,1984)
  - 相場亮：“これからの高レベル言語は？制約ロジックプログラミング言語”、数理科学、27、7、pp50~57 (July,1989)
  - 村岡洋一、丸島敏一：“Fortran用並列処理解析プログラムParaphrase”、コンピュータソフトウェア、5、3、pp.41~48 (July, 1988)
  - 笠原博徳：“最適化並列コンパイラ技術の現状”、電子情報通信学会誌、73、3、pp.258~266 (March,1990)
  - Anthony E. Terrano, Stanley M. Dunn and Joseph E. Peters: “Using an Architectural Knowledge Base to Generate Code for Parallel Computers”、Comm. of ACM. 32、9、pp.1065~1072 (Sep.,1989)
  - 笠原博徳、成田誠之助、橋本親：“OSCAR (Optimally Scheduled Advanced Multiprocessor) のアーキテクチャ”、電子情報通信学会論文誌、J71-D、8、pp.1440~1445 (Aug.,1988)
  - 出口弘：“並列画像生成システムLINKS”、bit臨時増刊、21、4、pp.192~201 (March, 1989)
  - Anne Dinning: “A Survey of Synchronization Methods for Parallel Computers”、IEEE Computer、22、7、pp.66~77 (July, 1989)
  - Allan Gottlich, et al: “The NYU Ultracomputer-Designing an MIMD Shared Memory Parallel Computer”、IEEE Trans. on Computer、C-32、2、pp.25~36 (Oct.,1989)
  - 石川勉、桃井茂晴、島田茂夫、小林正光、小川良夫：“階層化2次元アレー計算機HAPのアーキテクチャ”、電子情報通信学会論文誌、J71-D、8、pp.1433~1439 (Aug.,1988)
  - 自川友紀：“並列計算機PAX”、bit臨時増刊、21、4、pp.66~72 (1989)
  - Nicholas Carriero、David Gelernter: Linda and Friends”、IEEE Computer、19、8、pp.26~34 (Aug.,1986)
  - Charles, J. Fleckenstein, David Hemmendinger: “Using a Global Name Space for Parallel Execution of Unix Tools”、Comm. of ACM, 32、9、pp.1085~1090 (Sep.,1989)
  - 宮野悟：“並列化とその限界-理論的な側面から”、コンピュータソフトウェア、7、1、pp.2~15 (Jan.,1990)
  - 梅尾博司：“並列計算機のためのアルゴリズムとアーキテクチャ1”、bit、22、6、pp.617~623 (June,1990)
  - 奥川峻史：情報処理 情報通信 ハードウェア論、p.177、共立出版 (1988)
  - 近山隆：“並列推論マシンのOS PIMOS”、数理科学、27、7、pp.26~33 (July,1989)
  - Wan-Hong, S. Cheng and Virgil, E. Wallentine: “DEBL: A Knowledge-Based Language for Specifying and Debugging Distributed Programs”、Comm. of ACM, 32、9、pp.1079~1084 (Sep.,1989)
  - Leblanc, T. J., Mellor-Crummey, J. M.: “Debugging Parallel Programs with Instant Reply”、IEEE Trans. on Computer、C-36、4、pp.471~482 (Apr.,1987)
  - Charles, E. McDowell and David, P. Helmbold: “Debugging Concurrent Programs”、ACM Computing Survey、21、4、pp.593~622 (Dec.,1989)
  - 平尾延夫：“汎用並列計算機 Symmetry/Balance”、bit臨時増刊、21、4、pp.235~243 (Mar.,1989)
  - 松本一教、内平直志、本位田真一：“時相論理とその応用”、情報処理、30、6、pp.651~657 (June,1989)
  - 内田俊一、瀧和男、近山隆：“並列推論と知識情報処理”、数理科学、27、7、pp.5~10 (July,1989)
  - 奥川峻史：“超並列コンピュータ向きアーキテクチャ”、bit、22、7、pp.704~715 (July,1990)
  - 向殿政男編：フォールト・トレラント・コンピューティング、丸善 (1989)
  - A. L. De Cegama: The Technology of Parallel Processing-Parallel Processing Architectures and VLSI Hardware、

Volume 1, Prentice Hall Inc. (1989)

- R. W. Hockney, C. R. Jesshope: Parallel Computers, Adam Hilger (1981) (奥川峻史他訳：並列計算機、共立出版 (1984) )
- R. W. Hockney, C. R. Jesshope: Parallel Computers, 2nd Ed., Adam Hilger (1988)
- 富田真治：並列計算機構成編、昭晃堂 (1986)
- 富田真治、末吉敏則：並列処理マシン、オーム社 (1989)
- 高橋義造編：並列処理機構、丸善 (1989)
- 村岡洋一：VLSI コンピュータアーキテクチャ、オーム社 (1988)
- 村岡洋一他編：“並列コンピュータ・アーキテクチャ”、bit 臨時増刊、21、4 (1989)
- 雨宮真人、田中譲：コンピュータアーキテクチャ”、オーム社 (1988)
- H. S. Stone: High-Performance Computer Architecture, Addison-Wesley (1987)
- K. Hwang, D. DeGroot: Parallel Processing for Supercomputers and Artificial Intelligence, McGraw-Hill (1989)
- A. Dinning: “A Survey of Synchronization Methods for Parallel Computers”、IEEE Computer、22、7、pp.66~77 (1989)
- R. Duncan: “A Survey of Parallel Computer Architectures”、IEEE Computer、23、2、pp.5~16 (1990)
- D. B. Skillicorn: “A Taxonomy for Computer Architectures”、IEEE Computer、21、11、pp.46~57 (1988)
- S. Dasgupta: “A Hierarchical Taxonomic Systems for Computer Architectures”、IEEE Computer、23、3、pp.64~74 (1990)
- D. J. Lilja: “Reducing the Branch Penalty in Pipelined Processors”、IEEE Computer、21、7、pp.47~55 (1988)
- H. Cheng: “Vector Pipelining, Chaining, and Speed on the IBM 3090 and Cray X-MP”、IEEE Computer、22、9、pp.31~46 (1989)
- J. A. B. Fortes, B. W. Wah: “Systolic Arrays-From Concept to Implementation”、IEEE Computer、20、7、pp.12~17 (1987)
- S. Y. Kung, et al.: “Wavefront Array Processors-Concept to Implementation”、ibid. pp.18~33 (1987)
- I. Koren, et al.: “A Data-Driven VLSI Array for Arbitrary Algorithms”、IEEE Computer、21、10、pp.30~43 (1988)
- 曾和将容：データフローマシンと言語、昭晃堂 (1986)
- S. Bhatia: “On Modeling Data-flow Supercomputers with Modified Stochastic-Timed Petri Nets”、IEEE、pp.534~540 (1985)
- D. Ghosal, L. N. Bhuyan “Performance Evaluation of a Dataflow Architecture”、IEEE Trans. Comput., C 39、5、pp.615~627 (1990)
- L. N. Bhuyan: “Interconnection Networks for Parallel and Distributed Processing”、IEEE Computer、20、6、pp.9~12 (1987)
- W. J. Dally: “Performance Analysis of karyn-cube Interconnection Networks”、IEEE Trans, Comput., C 39、6、pp.775~785 (1990)
- S. P. Dandamudi, D. L. Eager: “Hierarchical Interconnection Networks for Multicomputer Systems”、ibid., pp. 786~797 (1990)
- D. A. Reed, D. C. Crunwald: “The Performance of Multicomputer Interconnection Networks”、IEEE Computer、20、6、pp.63~73 (1987)
- D. P. Agrawal, et al.: “Evaluating the Performance of Multicomputer Configurations”、IEEE Computer、19、5、pp.23~37 (1986)
- L. N. Bhuyan, D. P. Agrawal: “Generalized Hypercube and Hyperbus Structures for a Computer Network”、IEEE Trans. Comput., C 33、4、pp.323~333 (1984)
- 梅尾博司：“ピラミッド。アーキテクチャとそのアルゴリズム”、bit、22、4、pp.408~417 (1990)
- K. Hwang, j. Ghosh: “Hypernet: A Communication Efficient Architecture for Constructing Massively Parallel

- Computers” , IEEE Trans. Comput., C 36, 12, PP.1450~1466 (1987)
- 石川勉：“直径と接続数が小さい高並列計算機向きネットワーク“CCT cube””、電子情報通信学会論文誌、J73-D-I、6、pp.599~602 (1990)
  - A. El-Amawy, S. Latifi: “Properties and Performance of Folded Hypercubes”, IEEE Trans. Parallel and Distributed Systems, 2, 1, pp.31~42 (1991)
  - J-C Bermond, C. Peyrat: De Bruijn and Kautz Networks: A Competitor for the Hypercube, pp.279~293, in F. Andre, J. P. Verjus (Eds.) : Hypercube and Distributed Computers, North-Holland (1989)
  - D. K. Pladhan: “Fault-Tolerant Multiprocessor Link and Bus Network Architectures” , IEEE Trans. Comput., C-34, 1, pp.35~45 (1985)
  - M. R. Samatham, D. K. Pladhan: “The De Bruijn Multiprocessor Network: A versatile Parallel Processing and Sorting Network for VLSI” , IEEE Trans. Comput., C-38, 4, pp.567~581 (1989)
  - A. H. Esfahanian, S. L. Hakini: “Fault-Tolerant Routing in De Bruijn Communication Networks” , IEEE Trans. Comput., C 34, 9, pp.503~511 (1985)
  - Z. Du, F. K. Hwang: “Generalized de Bruijn Digraph” , Networks, 18, 1, pp.27~38 (1988)
  - M. Imase, M. Itoh: “Design to Minimize Diameter on Building Block Network” , IEEE Trans., Comput., C 30, 6, pp.439~443 (1981)
  - M. Imase, M. Itoh: “A Design for Directed Digraph with Minimum Diameter” , IEEE Trans. Comput., C 32, 9, pp.782~784 (1983)
  - 今瀬真、他：“障害耐力のあるプロセッサ相互結合ネットワーク”、電子通信学会論文誌、J68-D、8、pp.1449~1456 (1985)
  - S. B. Akers, et al.: “The Star Graph: An Attractive Alternative to the n-cube” , Proc. or Int. Conf on Parallel Processing, pp. 393~400 (1987)
  - L. N. Bhuyan, D. P. Agrawal: “Design and Performance of Generalized Interconnection Networks” , IEEE Trans. Comput., C 32, 12, pp.1081~1090 (1983)
  - L. N. Bhuyan, et al.: “Performance of Multiprocessor Interconnection Networks” , IEEE Computer, 22, 2, pp.25~37 (1989)
  - V. P. Kumar, S. M. Reddy: “Augmented Shuffle-Exchange Multistage Interconnection Networks” , IEEE Computer, 20, 6, pp.30~40 (1987)
  - V. P. Kumar, A. L. Reibman: “Failure Dependent Performance Analysis of a Fault-Tolerant Multistage Interconnection Network” , IEEE Trans. Comput., C-38, 12, PP.1703~1713 (1989)
  - A. Varma, C. S. Raghavendra: “Fault-Tolerant Routing in Multistage Interconnection Networks” , IEEE Trans. Comput., C-38, 3, pp.385~393 (1989)
  - W. Lin, et al.: “A Conflict-Free Routing Scheme on Multistage Interconnection Networks” , IEEE Trans. Comput., C-38, 8, pp.1086~1096 (1989)
  - 渋谷進：“いくつかの基本的な相互結合ネットワークにおける経路の選択と経路の衝突”、電子通信学会論文誌、J-69-D、3、pp.312~323 (1986)
  - G. Lee: “A Performance Bound of Multistage Combining Networks” , IEEE Trans. Comput., C-38, 10, pp.1387~1395 (1989)
  - D. M. Dias, J. R. Jump: “Analysis and Simulation of Buffered Delta Networks” , IEEE Trans. Comput. C-30, 4, pp.273~282 (1981)
  - H. Y. Kyungsook, et al.: “Performance Analysis of Multibuffered Packet-Switching Networks in Multiprocessor Systems” , IEEE Trans. Comput., C-39, 3, pp.315~327 (1990)
  - T. N. Mudge, et al.: “Multiple Bus Architectures” , IEEE Computer, 20, 6, PP.42~48 (1987)
  - 清水茂則、山内長承：“マルチプロセッサ・ワークステーションTOP-1”、bit、22、7、pp.767~774 (1990)



- C. R. Das, L. N. Bhuyan: "Bandwidth Availability of Multiple-Bus Multiprocessor", IEEE Trans. Computer., C-34, 10, pp.918~926 (1985)
- F. E. Guibaly: "Design and Analysis of Arbitration Protocol", IEEE Trans. Comput., C-38, 2, pp.161~171 (1989)
- B. L. Bodnar, A. C. Liu: "Modeling and Performance Analysis of Single-Bus Tightly Coupled Multiprocessors" IEEE Trans. Comput., C-38, 3, pp.464~470 (1989)
- P. Stenstroem: "Reducing Contention in Shared-Memory Multiprocessors", IEEE Computer, 21, 11, pp.26~37 (1988) D. Chaiken, et al: "Directory-Based Cache Coherence in Large-Scale Multiprocessors", IEEE Computer, 23, 6, pp.49~58 (1990)
- H. Cheong, A. V. Veidenbaum: "Compiler-Directed Cache Management in Multiprocessors", ibid., pp.39~47 (1990)
- Q. Yang, et al: "Analysis and Comparison of Cache Coherence Protocols for a Packet-Switched Multiprocessors", IEEE Trans. Comput., C-38, 8, pp.1143~1153 (1989)
- S. Thakkar, et al: "Scalable Shared-Memory Multiprocessor Architectures", ibid., pp.71~83 (1990)
- J.-L. Baer, C. Girault: A Petri Net Model for a Solution to the Cache Coherence Problem, Proc. 1st Jnt. Conf Super Computer System, pp.680~689 (1985)
- M. A. Marsan, et al.: Generalized Stochastic Petri Net Models of Multiprocessors with Cache Memories, ibid., pp.690~696 (1985)
- M. K. Vernon, M. A. Holliday: "Performance Analysis of Multiprocessor Cache Consistency Protocols Using Generalized Timed Petri Nets", Proc. Performance '86, pp.9~17 (1986)
- 元岡達、他: VLSI コンピュータ II、岩波書店 (1985)
- 石井光雄: 映像化マシン、オーム社 (1989)
- 小池誠彦: CAD マシン、オーム社 (1989)
- 村岡洋一、古谷立美: 知的連想メモリマシン、オーム社 (1989)
- 田中英彦、他: "並列処理マシン特集"、情報処理、28、1、pp.3~105 (1987)
- 青山友紀、小野定康: 信号処理プロセッサ、オーム社 (1990)
- 田丸啓吉、他: "専用 VLSI プロセッサ特集"、情報処理、31、4、pp.446~499 (1990)
- 長沼次郎、他: "連想メモリを用いた Prolog マシン (ASCA) のアーキテクチャ"、Proc. of the Logic Programming Conference '86, pp.185~191 (1986)
- 湯浅太一: "コネクションマシン"、bit、19、11、pp.36~44 (1987)
- 中田登志之、他: "並列回路シミュレーションマシン Cenju"、情報処理、31、5、pp.593~601 (1990)
- 村岡洋一: "超並列プロセッサ"、電子情報通信学会誌、72、7、pp.733~741 (1989)
- 村岡洋一: 超並列処理コンパイラ、コロナ社 (1990)
- 奥川峻史: "超並列コンピュータ向きアーキテクチャ"、bit、22、7、pp.704~714 (1990)
- 特集: "10年後の主役を目指す超並列マシン"、日経エレクトロニクス、No.478、pp.123~146 (1989)
- 木村博昭、福島実: トランスピュータによる並列処理、海文堂 (1990)
- 一岡芳樹、谷田純: "並列デジタル光コンピュータ"、別冊サイエンス 86、pp.90~101 (1988)
- 甘利俊一、後藤栄一編: "人工ニューラルシステム"、bit 臨時増刊、21、11 (1989)
- 阿江忠: "VLSI ニューロコンピュータのアーキテクチャ 1~10"、bit、21、10~22、6 (1989~1990)
- "特集、夢の次世代コンピュータ"、Computer Today、No.37 (1990.5)
- Abt, Clark. 1974. Serious Games. New York: Viking.
- Armstrong, Robert and Margaret Hobson. 1969. "Games and Urban Planning." Surveyor 31 (October):32-34.
- Becker, Henk A. and Alan L. Porter, eds. 1986a. Impact Assessment Today. 2 Vol. Utrecht, Netherlands: Uitgeverij Jan van Arkel.
- Becker, Henk A. and Alan L. Porter, eds. 1986b. Methods and Experiences in Impact Assessment. Dordrecht, Netherlands, and Boston/Lancaster/Tokyo: D. Reidel.

- Barthes, Roland. 1964. *Elements of Semiology*. London: Jonathan Cape.
- Biggs, William D. 1986. "Computerized Business Management Simulations for Tyros." In *Development in Business Simulation and Experiential Exercises*, Vol. 13, edited by Alvin C. Burns and Lane Kelly. Little Rock: ABSEL, University of Arkansas.
- Boocock, Sarane S. and E. O. Schild. 1968. *Simulation Games in Learning*. Newbury Park, CA: Sage.
- Branwyn, Gareth. 1986. "Gaming: Simulating Future Realities." *Futurist* (January-February):29-35.
- Bredemeier, Mary E. and Cathy Stein Greenblat. 1981. "The Educational Effectiveness of Simulation Games: A Synthesis of Recent Findings." *Simulation and Games* 12 (September).
- Bredemeier, Mary E., Naomi G. Rotter, and Ron Stadskev. 1981. "'The Academic Game' as a Frame Game." *Journal of Experiential Learning and Simulation (JELS)* 3(2):73-83.
- Bruin, Klaas, Jan de Haan, Cees Teijken, and Wijbren Veeman. 1979. *How to Build a Simulation/Game: Proceedings of the 10th ISAGA Conference*. Leeuwarden, Netherlands: Ubbo Emmius Teacher Training Institute.
- Burns, Alvin C. and Lane Kelley. 1986. *Developments in Business Simulation and Experiential Exercises*. Vol. 13. Stillwater: Oklahoma State University.
- Crookall, David, Allan Martin, Danny Saunders, and Alan Coote. 1986. "Human and Computer Involvement in Simulation." *Simulation and Games* 17(3):345-375.
- Crookall, David and Rebecca Oxford 1987 "Gaming Context, Communication, Reality and Future: An Introduction." In *Simulation/Gaming: State of the Art in the Late 1980's*, edited by D.
- Crookall, C. S. Greenblat, A. Coote, J. H. Kladders, and D. R. Watson. Oxford: Pergamon.
- Crookall, David, Danny Saunders, and Allan Coote. 1987. "The SIMULATION DESIGN GAME: An Activity for Exploring the Simulation Design Process." To be published in the *Proceedings of the SAGSET 1986 meetings*.
- Dillman, Duane. 1970. "The Design and Development of Simulation Exercises." Paper presented at the meetings of the American Educational Research Association, Minneapolis, MN, March.
- Duke, Richard D. 1974. *Gaming: The Future's Language*. New York: Halsted.
- Duke, Richard D. 1980. "A Paradigm for Game Design." *Simulation and Games* 11(September):364-377.
- Duke, Richard D. and Cathy S. Greenblat. 1979. *Game-Generating-Games*. Newbury Park, CA: Sage.
- Elgood, Chris. 1984. *Handbook of Management Games*. 3rd Ed. Brookfield, VT: Gower.
- Elliott, Charles and Brian McKeown. 1985. *It's Not Fair*. Dublin: Christian Aid.
- Ellington, Henry, Eric Addinall, and Fred Percival. 1983. *A Handbook of Game Design*. London: Kogan Page.
- Feldt, Allan and Frederick L. Goodman. 1975. "Observations on the Design of Simulations and Games." In *Principles and Practices of Gaming-Simulation*, edited by C. S. Greenblat and R. D. Duke. Newbury Park, CA: Sage.
- Fuhs, F. Paul. 1986. "The Design of Simulation Models Using Thought Organizers." In *Development in Business Simulation and Experiential Exercises*, Vol. 13, edited by Alvin C. Burns and Lane Kelly. Little Rock: ABSEL, University of Arkansas.
- Gagnon, John H. and Cathy S. Greenblat. 1978. "Pregnancy Before Marriage: Contraception, Abortion, and Out-of-Marriage Births." *Life Designs: Individuals, Marriages, and Families*. Glenview, IL: Scott, Foresman.
- Gamson, William. 1975. "SIMSOC: Establishing Social Order in a Simulated Society." In *Principles and Practices of Gaming-Simulation*, edited by C.S. Greenblat and R. D. Duke. Newbury Park, CA: Sage.
- Gamson, William. 1978. *SIMSOC: Simulated Society*, 3rd ed. New York: Free Press. Gamson, William. 1984. *WHAT'S NEWS: A Game Simulation of TV News, Coordinator's Manual*. New York: Free Press.
- Glazier, Ray. 1970. *How to Design Educational Games*. 2nd Ed. Cambridge, MA: Abt.
- Goodman, Fred L. and Community Systems Foundation. 1970. *The Policy Negotiations Simulation: Leader's Notebook*. Ann Arbor, MI: Learning Activities and Materials, Inc. Goodman, Fred L. and Larry Coppard, eds. 1979, *Urban Gaming-Simulation*. Ann Arbor, MI: School of Education.
- Goosen, Ken R. 1982. *A Comprehensive Guide to ABSEL's Conference Proceedings (1974-1981)*. Little Rock: University of Arkansas.
- Gordon, Amy K. 1970. *Games for Growth*. Palo Alto, CA: Science Research Associates. Gray, Paul and Israel Borovits. 1986. "The Contrasting Roles of Monte Carlo Simulation and Gaming in Decision Support Systems." *Simulations* 47(6):233-239.
- Greenblat, Cathy Stein. 1973. "Teaching with Simulation Games: A Review of Claims and Evidence." *Teaching Sociology* 1(October).
- Greenblat, Cathy Stein. 1974. "Sociological Theory and the 'Multiple Reality' Game." *Simulation and Games* 5(March).
- Greenblat, Cathy Stein. 1975. "Simulating Marital Decision Making." In *The Marriage Game: Understanding Marital Decision Making*, edited by Cathy S. Greenblat, Peter J. Stein, and Norman F. Washburne. New York:

Random House.

- Greenblat, Cathy Stein. 1980. "Group Dynamics and Game Design: Some Reflections." *Simulation and Games* 11 (March):35-58.
- Greenblat, Cathy Stein and Richard D. Duke. 1975. *Gaming-Simulation: Rationale, Design, Applications*. Newbury Park, CA: Sage.
- Greenblat, Cathy Stein and Richard D. Duke. 1981. *Principles and Practices of Gaming-Simulation*. Newbury Park, CA: Sage.
- Greenblat, Cathy Stein and John H. Gagnon. 1976. "Specifications for Design: Blood Money-Report to the National Heart and Lung Institute." New Brunswick, New Jersey. (unpublished)
- Greenblat, Cathy Stein, Linda Reich Rosen, and John H. Gagnon. 1978. "Sex Education and Pre-Marital Pregnancy: An Innovative Approach." Paper presented at the 3rd International Congress on Medical Sexology, Rome, Italy, October.
- Greenblat, Cathy Stein, Peter J. Stein, and Norman F. Washburne. 1977. *The Marriage Game: Understanding Marital Decision Making*. New York: Random House.
- Henderson, Thomas A and John L. Foster. 1978. *URBAN POLICY GAME: A Simulation of Urban Politics*. New York: John Wiley.
- Hoffman, Therese Lemire and Susan Dempsey Reif. 1978. *INTO AGING: A Simulation Game*. Thorofare, NJ: Charles B. Slack.
- Hope, Joanne and Peter McAra. 1984. *Games Nurses Play*. New York: Pergamon.
- Horn, Robert and Anne Cleaves. 1980. *The Guide to Simulations/Games for Education and Training*. Newbury Park, CA: Sage.
- Inbar, Michael and Clarice Stoil. 1972. *Simulation and Gaming in Social Science*. New York: Free Press.
- Jones, Ken. 1980. *Simulations: A Handbook for Teachers*. London: Kogan Page.
- Jones, Ken. 1982. *Simulations in Language Teaching*. Cambridge: Cambridge University Press.
- Jones, Ken. 1986. *Designing Your Own Simulations*. New York: Methuen.
- Kenner, Joseph and Myron Uretsky. 1986. "The Management Decision Laboratory at New York University." In *Developments in Business Simulation and Experiential Exercises*, Vol. 13, edited by Alvin C. Burns and Lane Kelly. Little Rock: ABSEL, University of Arkansas.
- Klabbers, Jan H. G. 1984. "Design Characteristics of the Simulation-Game PERFORM." *Simulation and Games* 15 (June).
- Klabbers, Jan H. G. 1985. "Instruments for Planning and Policy Formation: Some Methodological Considerations." *Simulation and Games* 16(June): 135-161.
- Lederman, Linda. 1983. "Differential Learning Outcomes in an Instructional Simulation: Exploring the Relationship between Designated Role and Perceived Learning Outcome." *Communication Quarterly* 32: 198-204.
- Lederman, Linda. 1984. "Debriefing: A Critical Re-examination of the Post-Experience Analytic Process with Implications for Its Effective Use." *Simulation and Games* 15:415-431.
- Lederman, Linda and Lea P. Stewart. 1983. *The SIMCORP Simulation Participant's Manual*. Princeton, NJ: Total Research Corp.
- Livingston, Samuel A. 1972. "How to Design a Simulation Game." Baltimore, MD: Academic Games Associates.
- Livingston, Samuel A. 1973. "Six Ways to Design a Bad Simulation Game." *Simulation/Gaming/News* (March): 15.
- McClellan, Larry A. 1971. "Simulation Game Development: An Instructional Packet." Chicago: Governors State University.
- Meadows, Dennis L. 1985. "User's Manual for STRATEGEM-1, A Microcomputer-Based Management Training Game on Energy-Environment Interactions." Report DSD #509, Thayer School of Engineering, Hanover, New Hampshire 03755.
- Meadows, Dennis, William W. Behrens III, Donella H. Meadows, Roger F. Naill, Jorgen Randers, and Erich K.O. Zahn. 1974. *Dynamics of Growth in a Finite World*. Cambridge: MIT Press.
- Moore, Omar Khayyam and Alan Ross Anderson. 1975. "Some Principles for the Design of Clarifying Educational Environments." In *Principles and Practices of Gaming-Simulation*, edited by C. S. Greenblat and R. D. Duke. Newbury Park, CA: Sage.
- Philliber, Susan and Elaine M. Gutterman. 1982. "Playing Games about Teenage Pregnancy." *Evaluation and Program Planning* 3(4 1).
- Raser, John R. 1969. *Simulation and Society*. Boston: Allyn & Bacon. Rhyne, R. F. 1975. "Communicating Holistic Insights." In *Principles and Practices of Gaming-Simulation*, edited by C. S. Greenblat and R. D. Duke.

Newbury Park, CA: Sage.

- Richardson, George P. and Alexander L. Pugh III. 1981. "Chapter 2: Problem identification and System Conceptualization." *Introduction to System Dynamics Modeling with DYNAMO*. Cambridge: MIT Press.
- Roberts, Nancy, David Anderson, Ralph Deal, Michael Garet, and William Shaffer. 1983. *Introduction to Computer Simulation: A System Dynamics Modeling Approach*. Reading, MA: Addison-Wesley.
- SAGSET. 1986. *Debriefing Simulation/Games*. *Simulation/Games for Learning*, guest edited by Morry van Ments, 16, 4 (December).
- Sanoff, H. 1979. *Design Games*. Los Altos, CA: William Kaufmann.
- Sarason, Seymour. 1971. *The Culture of the School and the Problem of Change*. Boston: Allyn & Bacon.
- Schellenberger, Robert E. and Lance A. Masters. 1986. "An Example: The Use of Management Games on Microcomputers by Computer Novices." In *Developments in Business Simulation and Experiential Exercises*, Vol. 13, edited by Alvin C. Burns and Lane Kelly. Little Rock: ABSEL, University of Arkansas.
- Sherrell, Daniel L., Kenneth R. Russ, and Alvin C. Burns. 1986. "Enhancing Mainframe Simulations via Microcomputers: Designing Decision Support Systems." In *Developments in Business Simulation and Experiential Exercises*, Vol. 13, edited by Alvin C. Burns and Lane Kelly. Little Rock: ABSEL, University of Arkansas.
- Shirts, R. Garry. 1975. "Ten 'Mistakes' Commonly Made by Persons Designing Educational Simulations and Games." Del Mar, CA: SIMILE II.
- Shubik, Martin. 1968. "Gaming: Costs and Facilities." *Management Science* 14(July):629-660.
- Shubik, Martin. 1975. *Games for Society, Business and War: Towards a Theory of Gaming*. New York: Elsevier.
- Stahl, Ingolf, ed. 1983. *Operational Gaming: An International Approach*. Laxenburg, Austria: International Institute For Applied Systems Analysis.
- Sterman, John D. and Dennis Meadows. 1985. "STRATEGEM-2: A Microcomputer Simulation Game of the Kondratiev Cycle." *Simulation and Games* 16(2): 174-202.
- Suits, Bernard. 1978. *The Grasshopper: Games, Life and Utopia*. Toronto: University of Toronto Press.
- Suits, Bernard. 1982. "Games and Utopia: Posthumous Reflections." Keynote address at the 1982 NASAGA meetings, Ann Arbor, MI, October.
- Taylor, John and Rex Walford. 1978. *Learning and the Simulation Game*. Milton Keynes.
- Thatcher, Don and June Robinson. 1980. *A Game Workshop for the Design of Games and Simulations for Teaching* Portsmouth, England: Mimeographed.
- Thiagarajan, Sivasailam and Harold D Stolovich. 1980. "Frame Games: An Evaluation." Pp. 98-107 in *The Guide to Simulations/Games for Education and Training*, edited by Horn and Cleaves. Newbury Park, CA: Sage.
- Van Ments, Morry. 1983. *The Effective Use of Role-Play: A Handbook for Teachers*. London: Kogan Page.
- Wolfe, Joseph. 1985. "The Teaching Effectiveness of Games in Collegiate Business Courses: A 1973-1983 Update." *Simulation and Games* 16(2):251-288.
- Zelmer, Amy E. and A. C. Lynn Zelmer. 1982. *Simulation/Gaming for Health Teaching*. Canadian Health Education Society Technical Publications.
- Designed by A. N. O'Connell, J. L. Alpert, M. E. Bredemeier, M. S. Richardson, N. G. Rotter, and R. K. Unger (Task Force on Career Development Simulation of the American Psychological Association's Division on the Psychology of Women). Published by Institute of Higher Education Research and Services, Box 6293, University, AL 35846.
- Designed by Susan Ebel and Jean L. Easterly. Published by SIMILE II, P.O. Box 910, Del Mar, CA 92014.
- Designed by Richard D. Duke, John Morris, and Patrick L. Sweet. Published by The Center for Social and Economic Issues, Industrial Technology Institute, P.O. Box 1485, Ann Arbor, MI 48106.
- Designed by Richard D. Duke and Cathy Stein Greenblat. Published in *Game-Generating-Games*. Newbury Park, CA: Sage.
- Designed by R. Garry Shirts. Published by SIMILE II, P.O. Box 910, Del Mar, CA 92014.
- Designed by Georgeann Wilcoxson. Published by John Knox Press, 341 Ponce de Leon Avenue, N.E., Atlanta, GA 30308.
- Designed by Cathy Stein Greenblat and John H. Gagnon. Published by National Heart, Lung, and Blood Institute, POCE, Bethesda, MD.
- Designed by Richard V. Cotter and David J. Fritzsche. Published by Prentice-Hall, Inc., Englewood Cliffs, NJ 07632.
- Designed by Drew Mackie. Available from Drew Mackie, 64, The Causeway, Duddingston, Edinburgh, EH15 3P2, Scotland.
- Designed by Cathy Stein Greenblat, with Philip Langley, Jacob Ngwa, Saul Luyumba, Ernest Mangesho, and

Foday MacBailey. Information available from Cathy Stein Greenblat, Professor of Sociology, Rutgers University, New Brunswick, NJ 08903.

- CARIBBEAN FISHERMAN, Designed by Rex Walford. Published by Longman Resource Units, London, UK. Also available from Cambridge Publishing Services, Oatlands, High Street, Conington, Cambridgeshire, CB3 8LT, Elsworth (09547) 349, England.
- CITY 1 , Designed by Peter House et al. Published by National Technical Information Service, 5285 Port Royal Road, Springfield, VI 22151.
- CLASSROOM EXPERIENCING , Designed by Alice S. Jackson. Published by Academic Games Associates, Baltimore, MD (game now out of print).
- COLLECTIVE BARGAINING SIMULATED , Designed by Michael R. Carrell and Jerald R. Smith. Published by Charles E. Merrill Publishing Company, 1300 Alum Creek Drive, P.O. Box 508, Columbus, OH 43216-0508.
- THE COMMONS GAME , Designed by Richard Powers, Richard E. Duus, and Richard D. Norton. Suite 4H, 10 West 66th Street, New York, NY 10023.
- COMPETE-A DYNAMIC MARKETING SIMULATION , Designed by Anthony J. Faria, Ray Nulsen, and Phillip Roussos. Published by McGraw-Hill Book Company, Avenue of the Americas, New York, NY.
- THE CONCEPTUAL MAPPING GAME , Designed by Richard D. Duke and Cathy Stein Greenblat. Published in Game-Generating-Games. Newbury Park, CA: Sage.
- THE CONDENSED BUSINESS EXPERIENCE PROGRAM. , Designed by Myron Uretsky. Published by Business Simulations, Inc., 283 Hicks St., Brooklyn, NY 11301.
- THE CONFERENCE GAME , Designed by Chris Brand and Terry Walker. Published by Maxim Consultants, 6 Marlborough Place, Brighton, Sussex, BNI 3XA, England.
- CRISIS , Designed by Western Behavioral Sciences Institute. Published by SIMILE II, P.O. Box 910, Del Mar, CA 92014.
- CUSTOD-E , Designed by Dorothy Derzog and Kathleen Lupo. Information available from Dorothy Derzog, Fair Oaks Hospital, 19 Prospect Street, Summit, NJ 07901 or Kathleen Lupo, Middlesex University Hospital, New Brunswick, NJ 08901.
- DANGEROUS PARALLEL , Designed by Foreign Policy Association. Published by Scott, Foresman and Company, 1900 E. Lake Avenue, Glenview, IL 60025.
- DEMOCRACY , Designed by Academic Games Project, Johns Hopkins University. Published by Western Publishing Company, Inc., School and Library Department, 850 Third Avenue, New York, NY 10022.
- EDGE CITY COLLEGE , Designed by Noel Callahan, Dwight Caswell, Larry McClellan, Robert Mullen, and William N. Savage. Published by Urbandyne, Chicago, IL (game now out of print).
- END OF THE LINE , Designed by Frederick L. Goodman. Published by Institute of Higher Education Research and Services, Box 6293, University, AL 35846.
- THE EXECUTIVE SIMULATION , Designed by Bernard Keys and Howard Leftwich. Published by Kendall/Hunt Publishing Company, Dubuque, IO.
- THE FARMING GAME (formerly THE POVERTY GAME) , Adapted from a game by Jim Dunlop. Published by Oxfam Education Department, 274 Banbury Road, Oxford, OX2-7D ZEB, England.
- FISH BANKS, LTD. , Designed by Dennis Meadows and Diana Shannon. Available from Resource Systems Group, Box 250, Plainfield, NH 03781.
- GENERATION GAP , Designed by Erling O. Schild and Sarane S. Boocock. Published by Western Publishing Company, Inc., School and Library Department, 850 Third Avenue, New York, NY 10022.
- GHETTO , Designed by Dove Toll. Published by Bobbs-Merrill Company, Educational Division, 4300 W. 62nd St., Indianapolis, IN 46268.
- THE GREEN REVOLUTION GAME , Designed by Graham Chapman. Published by The Green Revolution Partnership, 2A Green End Road, Cambridge CB4 1RX, U.K.
- HALLOWEEN GAME , Designed by Cathy Greenblat, John Gagnon, Kevin Greenblat, and Leslie Greenblat. Information available from CSG Enterprises, 34 Bayard Lane, Princeton, NJ 08540.
- THE HEMOPHILIA PLANNING GAME , Designed by Cathy Stein Greenblat and John Gagnon. Published by National Heart, Lung, and Blood Institute, OPCE, Bethesda, MD.
- THE HEXAGON GAME , Designed by Richard D. Duke, Jeffrey K. Harris, Nancy Steiber, and Jo H. Webb. Published by Multilogue Inc., 321 Park Lake, Ann Arbor, MI 48103.
- HORATIO ALGER , Designed by Ann Kraemer, Bob Preuss, and Helen Howe. Published by Citizens for Welfare Reform, 305 Michigan Ave., Detroit, MI 48226.
- ICONS (INTERNATIONAL COMMUNICATIONS AND NEGOTIATIONS SIMULATIONS) , Designed by Jonathan Wilkenfeld and Richard Brecht, based on POLIS, by Robert Noel. Information available from Jonathon

Wilkenfeld, Department of Government and Politics, University of Maryland, Lefrak Hall, College Park, MD 20742.

- IMPASSE , Designed by Richard D. Duke and Cathy Stein Greenblat. Published in Game Generating Games. Newbury Park, CA: Sage.
- INTER-NATION SIMULATION (INS), Designed by Harold Guetzkow and Cheo Cherryholmes. Published by Science Research Associates, Inc., 155 North Wacker Drive, Chicago, IL 60006.
- INTO AGING , Designed by Therese Lemire Hoffman and Susan Dempsey Reif. Published by Charles B. Slack, Inc., Thorofare, NJ 08086.
- THE MANAGEMENT DECISION LABORATORY (MDL) , Designed by Myron Uretsky et al. Information available from Myron Uretsky, Graduate School of Business, New York University, 100 Trinity Place, New York, NY 10006.
- MANSYM (4th Edition), Designed by Robert E. Schellenberger and Lance A. Masters. Published by John Wiley and Sons, 605 Third Avenue, New York, NY 10158.
- THE MARBLE COMPANY I, Designed by Linda Costigan Lederman and Lea P. Stewart. Published by the Department of Communication, School of Communication, Information and Library Studies, Rutgers, The State University of New Jersey, New Brunswick, NJ 08903.
- THE MARRIAGE GAME: UNDERSTANDING MARITAL DECISION MAKING , Designed by Cathy Stein Greenblat, Peter J. Stein, and Norman F. Washburne. Published by Random House, Inc., 201 E. 50th St., New York, NY.
- MAXIMASIA , Designed by Maxim Training Systems, Ltd. Available from Maxim Training Systems, Ltd., 6 Marlborough Place, Brighton, Sussex BN1 1UB, England.
- ME, THE SLOW LEARNER , Designed by Don Thatcher and June Robinson. Published by Solent Simulations, 80, Miller Drive, Fareham, Hants, P016 7LL, England.
- METRO-APEX , Designed by Richard D. Duke. Published by Mark James, Director of Computing Services, COMEX, Davidson Conference Center, University of Southern California, Los Angeles, CA 90007.
- MODERN BUSINESS DECISIONS , Designed by Richard V. Cotter and David J. Fritzsche. Published by Prentice -Hall, Inc., Englewood Cliffs, NJ 07632.
- PERFORM , Designed by Jan H. G. Klabbers. Information available from Professor Jan N. G. Klabbers, c/o Faculty of Sciences, P.O. Box 80140, 3508 TC Utrecht, The Netherlands.
- PLANAFAM , Designed by Harold Thomas and Katherine Finseth. Published by ERIC, 855 Broadway, Boulder, CO 80302.
- POLICY NEGOTIATIONS , Designed by Frederick L. Goodman. Published by Institute of Higher Education Research and Services, Box 6293, University, AL 35846.
- POMP AND CIRCUMSTANCE , Designed by Cathy Stein Greenblat, John H. Gagnon, and Linda ReichRosen. Published by CSG Enterprises, 34 Bayard Lane, Princeton, NJ 08540.
- PRODUCTION LINE, INC. Designed by Dennis Meadows, Elzbieta Naumienko, Diana Shannon, and Paula Antunes. Available from Resource Systems Group, Box 250, Plainfield, NH 03781.
- RADIO COVINGHAM, Designed by Ken Jones. Published by Max Hueber Verlag, Max Hueber Strasse 4, 8045 Ismaning, West Germany.
- RED DESERT , Designed by Ken Jones. Published by Max Hueber Verlag, Max Hueber Strasse 4, 8045 Ismaning, West Germany.
- RELOCATION , Designed by SIMILE II. Available from SIMILE II, P.O. Box 910, Del Mar, CA 92014.
- SHIPWRECKED , Designed by Ken Jones. Published by Max Hueber Verlag, Max Hueber Strasse 4, 8045 Ismaning, West Germany.
- SIMCORP , Designed by Linda Costigan Lederman and Lea P. Stewart. Published by the Department of Communication, 4 Huntington Street, New Brunswick, NJ 08903.
- SIMSOC (Third Edition) , Designed by William A. Gamson. Published by the Free Press, 866 Third Avenue, New York, NY 10022.
- SNUS-SIMULATED NUTRITION SYSTEM , Designed by Richard D. Duke. Published by the Certificate Program in Gaming/Simulation, University of Michigan, Ann Arbor, MI 48109.
- STARPOWER , Designed by R. Garry Shirts. Published by SIMILE II, P. O. Box 910, Del Mar, CA 92014.
- ST. PHILIP: A SIMULATION ABOUT THE DEVELOPMENT OF A CARIBBEAN ISLAND , Designed by Rex Walford. Published in Journal of Geography, Volume 82, Number 4, July-August 1983:pp. 170-175. Also available from Cambridge Publishing Services, Oatlands, High Street, Conington, Cambridgeshire, CB3 8LT, Elsworth (09 547) 349, England.
- STRATEGEM-1 , Designed by Dennis Meadows, Donella Meadows, Diana Shannon, and Ferenc Toth. Available from Resource Systems Group, Box 250, Plainfield, NH 03781.

- STRATEGEM-2 , Designed by John D. Sterman and Dennis Meadows. Published in Simulation and Games, 16,2 (June 1985): 174-202. Also available from John Sterman, System Dynamics Group, E40-294, Massachusetts Institute of Technology, Cambridge, MA 02139 or from Resource Systems Group, Box 250, Plainfield, NH 03781.
- STRATPLAN , Designed by Roy W. Hinton and Daniel C. Smith. Published by Prentice-Hall, Inc., Englewood Cliffs, NJ 07632.
- THEY SHOOT MARBLES, DON'T THEY? , Designed by Frederick L. Goodman. Information available from Frederick L. Goodman, School of Education, University of Michigan, Ann Arbor, MI.
- THE TRADING GAME , Published in It's Not Fair, Trocaire, 169 Booterstown Avenue, Blackrock, County Dublin, Ireland.
- URBAN POLICY GAME , Designed by Thomas A. Henderson and John L. Fox. Published by MacMillan Publishing Company, 866 Third Avenue, New York, NY 100 22.
- WHAT HAPPENS WHEN THE GAS RUNS OUT? , Designed by Henry Ellington, Eric Addinall, and Fred Percival. Published by the Chemical Society, Education Division, Burlington House, Piccadilly, London W1V 0BN, England.
- WHAT'S NEWS? Designed by William A. Gamson. Published by The Free Press, 866 Third Avenue, New York, NY 10022.
- YOUR NUMBER IS UP , Designed by Joanne Hope and Peter McAra. Published in Joanne Hope and Peter McAra, Games Nurses Play: Experimental Games for Behavioural Science Programmes. New York and Oxford: Pergamon Press, 1984.
- ARTHUR, D J W and TYRER, M. (1996) PHREEQE 96 USER GUIDE: 'Consolidation of HMIP Geochemical Speciation-Solubility Model'. UK DOE, Report DOE/HMIP/CPR2/4 1/1/98 .
- ASHTON, J, BROYD, T W, JONES, M A, KNOWLES, N C, LIEW, S K, MAWBEY, C S, READ, D and SMITH, S L (1993) A directory of computer programs for assessment of radioactive waste disposal in geological formations. CEC Report EUR14201 .
- BENNETT, D G, LIEW, S K, MAWBEY, C S and READ, D (1991) CHEMTARD Theoretical Overview. UK DOE Report DoE/HMIP/RR/92.036.
- BERGSTROM, U, EDLUND, O, EVANS, S and ROJDER, B (1982) BIOPATH - A computer code for calculation of the turnover of nuclides in the biosphere and the resulting doses to man. Studsvik AB, Sweden (STUOSVIK /NW-82/26 1 ) .
- BICKNELL, B R, IMHOFF, J C, KITTLE, J L Jr, DONIGAN, A D Jr, and JOHANSON, R C ( 1993) Hydrological Simulation Program Fortran: User 's manual for release I O: Environmental Research Laboratory. EPA/600/R-93/174, Athens, Ga.
- BIOMOV5 2 Reports ( 1996) Biosphere Modelling for Dose Assessments of Radioactive Waste Repositories: Final Report of the Complementary Studies Working Group.[NL] Sept 1996, ISBN 91-972958-1-7 (TR12).
- BROWN, P L, HAWORTH, A, SHARLAND, S M and TWEED, C H (1991) HARPHRQ.' An extended version of geochemical code PHREEQE. Nirex Safety Studies Report, NSS R1 88.
- COX, J D, WAGMAN, D D and MEDVEDEV, V A (1989) CODATA Key Values for Thermodynamics. -27 Ip., New York (Hemisphere Publ. Corp.).
- CRANWELL, R M, GUZOWSKI, R V, CAMPBELL, J E and ORITZ, N R (1982) Risk methodology for geologic disposal of radioactive waste: Scenario selection procedure. Sandia National Laboratories Report SAND 8 1 -2573 (NUREG/CR-2452), reprinted in I 987.
- CROFF, A G. (1980) A User's Manual for the ORIGEN2 Computer Code. ORNL/TM-7175 (July 1980).
- CROFF, A G (1983) ORIGEN2: A Versatile Computer Code for Calculating the Nuclide Compositions and Characteristics of Nuclear Materials. Reprint from Nuclear Technology, 62, 335-351 (September, 1983).
- DAMES & MOORE (1991) User Guide for TIME4 (Version 1.0). UK DOE Disposal Assessments User Document no-D&M-2. Report DoE/HMIP/RR/9 1 .
- ESRI White Paper Series. (March 1995a) ARC/INFO.' The World's GIS.
- ESRI White Paper Series (July 1995b). Spatial Database Engine (SDE).
- ESRI White Paper Series. (November 1996) ArcView GIS Version 3.0.
- EWEN, J (1990) Basis for the subsurface contaminant migration components of the catchment water flow, sediment transport and contaminant migration modelling system.
- SHETRAN -UK. UK Nirex Ltd., Harwell. NSS/R229.
- JONES, M A. (1990) Gas Generation in Deep Radioactive Waste Repositories: A Review of Processes, Controls and Models. UK DOE Report DOE/HMIP/ RR/90/086.
- LEE, J.H and ATKINS, J.E (1995) Chapter 5. Waste Package Degradation Abstraction in the Total System

Performance Assessment - 1995 Civilian Radioactive Waste Management System Management and Operating, Contractor. BOOOOOO00-01 717-2200-00136 REV 1. Las Vegas NV.

- MCCRINDLE, R and DOGGETT, S (1996) The Multimedia Maintenance Interface (MuMMI) System. Paper presented at BCS Software Creativity 1996.
- MARSH, G.P. (1988a) Progress in the assessment of the corrosion of low and intermediate level waste containers under repository conditions. NSS/RI 26.
- MARSH, G.P. (1988b) Predicting the long term corrosion of metal canisters for nuclear waste disposal. In: 'Scientific Basis for Nuclear Waste Management XI'. M J Apted and R E Westerman (Eds.) Mat. Res. Soc. Symp. Proc., 112, 85-97.
- MURI-AY, K and MAY, P M (1995) JESS User's Manual. July 1995.
- NEA (1996) Summary of thermodynamic and kinetic databases NENNSC/DOC(96)27.
- PARKHURST, D L, THORSTENSON, D C and PLUMMER, L N (1980)
- PHREEQE - A Computer Program for Geochemical Calculations. US Geological Survey, Water Resources Investigations Report 80-96.
- PETCH, J, MOSS, A, JOHNSTON A, YIP, Y J, BASDEN, A, LI, C S, COLE, K and KITMITTO, K (1995), Knowledge-based Interfaces for National Data Sets: The KINDS Project. Pre-proceedings of 1995 Seminar on New Techniques and Technologies for Statistics (NTTS), 20-22 November 1995, Bonn, Germany, pp 395-402.
- PONTING, D K, FOSTER, B A, NACCADE, P F, NICHOLAS, M D, POLLARD, R K, RAE, J, BANBO, D and WALSH, S K (1983) An Efficient Fully Implicit Simulator. Society of Petroleum Engineers Transactions, 275, 544-552.
- PRUESS, K (1991) TOUGH2- A General-Purpose Numerical Simulator for Multiphase Fluid and Heat Flow. LBL-29400 ; UC-25 1 .
- PURDOM, G and AGG, P J (1993) GAMMON: A computer program addressing gas generation in radioactive waste repositories - Part A : Overview. NSS/R338; AEA-D&R-0455.
- READ, D (1990) CHEMVAL: Report on stage 2 laboratory and field data sets.. CEC Report EURI 3 124.
- READ, D (1991) CHEMVAL project. Report on stages 3 and 4. Testing of coupled chemical transport models. Nuclear Science and Technology, CEC Report EUR I 3675 EN, 234p.
- READ, D (1994) CHEMVAL-2 project. Report on stage 1: Status of six technical areas. Nuclear Science and Technology, CEC Report EUR I 5 1 61 EN, 238p.
- READ, D and BROYD, T W (1989) CHEMVAL project. Report on stage 1.-Verification of speciation models. Nuclear Science and Technology, CEC Report EUR 12237 EN, 365p.
- READ, D and FALCK, W E (1996) CHEMVAL2. A coordinated research initiative for evaluating and enhancing chemical models in radiological risk assessment. Final Report. CEC Report EUR 16648 EN.
- REEVES, M, WARD, D S , JOHNS, N D and CRAIVELL, R M (1986) Theory and Implementation for SWIFT II the Sandia Waste-Isolation Flow Transport Model for Fractured Media. Release 4.84, NUREG/CR-3328 (SAND83-1 1 59).
- SCOTT, T (1990) PORES - General Purpose Black Oil Simulator. AEA Petroleum Services.
- SILVA, R J, BIDIGLIO, G, RAND, M H and ROBOUCH, P B (1995) Chemical Thermodynamics Series. Volume 2. 'Chemical Thermodynamics of Americium'. - Paris (OECD-NEA).
- SPENCE, R., BHOGAL, R., TWEEDIE, L and SU, H (1995) Responsive Visualisation - a Tool for Analog Designers Proc. Int. Conf. on Circuits and Systems (ISCAS), Seattle, May 1995.
- SVALSTAD, D K (1991) Documentation for SPECTROM-41: A finite element Heat transfer analysis program SAND 88-7122, Sandia National Laboratories, Albuquerque, NM 87 1 85.
- TWEEDIE, L., SPENCE, R., WILLIAMS, D and BHOGAL, R (1994) The Attribute Explorer. Video Proc. CHI '94, Boston 24-28 April, 1994 ACM Press.
- WEBSTER, E B (1995) FISPIN 7 Introductory Guide. AEA Technology, The ANSWERS Software Service, Technical Services Division, Winfiith Technology Centre.
- WOLERY, T J (1979) Calculation of chemical equilibrium between aqueous solutions and minerals: The EQ3/6 Software Package. Lawrence Livermore Laboratory Report UCRL-52658.
- WOLERY, T J (1983) EQ3NR, A computer program for geochemical aqueous speciation-solubility calculations: User's Guide and documentation. Lawrence Livermore Laboratory Report UCRL-534 1 4.
- WOLERY, T J (1992) EQ3/6 Software Package for Geochemical Modelling of Aqueous Systems: Package Overview and Installation Guide (Version 7. 0). Lawrence Livermore Lab. Report UCRL-MA- I I0662 PT I .
- YEH, G T (1994) User's manual for LEHGC: A Lagrangian-Eulerian finite element model of HydroGeoChemical transport through saturated-unsaturated media - Version 1.0. Sandia National Laboratories report SAND93-7081, I 57 pp .



- YEH, G T and TRIPATHI, V S (1991) HYDROGEOCHEM: A coupled model of HYDROlogic transport and GEOCHEMical equilibria in reactive multicomponent systems. ORNL-6371, Oak Ridge National Laboratory, TN, 312 p.

## ソフトウェア・エンジニアリング関連の参考文献

- Deardorff, E., Presentation to HP's Third Annual Software Metrics Council Meeting, Palo Alto, Calif (Nov. 1985).
- Basili, V., Presentation to IEEE Working Group for Software Productivity Metrics, Nashua, N. H. (Sept. 1984).
- Peters, T. and R. Waterman, In Search of Excellence. New York: Harper & Row 1982, pp. 240, 242.
- Conte, S., H. Dunsmore, and V. Shen, Software Engineering Metrics and Models. Menlo Park, Calif: Benjamin/Cummings Publishing Co., Inc., 1986, p. 173.
- DeMarco, T., Controlling Software Projects. New York: Yourdon Press, 1982.
- Balza, J., "Improving the Methods of Software Project Estimation at CNO," HP Software Productivity Conference Proceedings (April 1986), p. 1-12.
- Christensen, K., G. Fitsos, and C. Smith, "A Perspective on Software Science," IBM System Journal, Vol. 20, no.4 (1981).
- Rambo, R., P. Buckley, and E. Branyan, "Establishment and Validation of Software Metric Factors," Proceeding of the International Society of Parametric Analysts Seventh Annual Conference (May 1985), pp. 410-416.
- Arthur, L. J., Measuring Programmer Productivity and Software Quality. New York: John and Sons, 1985, pp. 65-73.
- Curtis, B., S. Sheppard, and P. Milliman, "Third Time Charm: Stronger Prediction of Programmer Performance by Software Complexity Metrics," IEEE Proceedings of the Fourth International Conference on Software Engineering (1979), pp. 356-360.
- DeMarco, T., Controlling Software Projects. New York: Yourdon Press, 1982, pp. 80-91, 104-112. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- BIRCH M and WHITELEY, K (1989) A Knowledge Base Approach to the Specification of Real Time System Requirements Second International Conference on Software Engineering in Real Time Systems IEE ConfPub no 309, 21 - 25.
- BOEHM, B L, BROWN, J R, KASPAR, H, LIPOW, M, MACLEOD, G J and MERRITT, M J (1978) Characteristics of Software Quality. North Holland, Amsterdam.
- BOEHM, B W (1986) A Spiral Model of Software Development and Enhancement ACM SIGSOFT. 11, 14-24.
- BOEHM, B W (1988) A Spiral Model of Software Development and Enhancement IEEE Computer. 21, 61-72.
- BOWEN, J P and HINCHLEY, M G (1995) Seven More Myths of Formal Methods IEE Software. 12, 34 - 41.
- CHUDLEIGH, M (1989) Developing Software for Safety-Critical Applications The Nuclear Engineer, 30 (1), 14 - 25.
- CHUDLEIGH, M (1990) Software and safety.' how compatible are they. Information and Software Technology, 32(5), 323 - 329.
- CRAIGEN, B, GERHART, S and LSTON, I (1995) Formal Methods Reality Check.' Industrial Usage IEEE Trans Software Eng. 21, 90 - 98.
- DAI, H and SCOTT C H (1995) A VAT. a CASE Tool for Software Verification and Validation Proc 7th International Workshop on Computer Aided Software Engineering, Toronto, 10 - 14 July 1995, 358 - 367.
- DEVLIN, M AND ROYCE, W (1996) Improving Software Economics in the Aerospace and Defense Industry. WWW document at URL: <http://www.rational.com/htd/ocs/pst/tech-papers/tp46.html>
- DROMEY, R G (1991) A Model for Software Product Quality IEEE Trans Software Eng 21(2), 146 - 162.
- FROOME, P and MONAHAN, B (1988) The role of mathematically formal methods in the development and assessment of safety-critical systems Microproc Microsys. 12, 539 - 546.
- GERHART, S, CRAIGEN, B and RALSTON, T (1994) Experience with Formal Methods in Critical Systems IEEE Software 11, 21 - 28.
- GOWEN, L D (1993) Developing and Analysing High-Level Designs for Safety-Critical Software Systems Proc SOUTHEASTCON I 993, (Charlotte, NC, USA, 4-7 April 1993) 505 - 512.
- GOWEN, L D and COLIOFELLO, J S (1994) Assessing Traditional Verification b Effectiveness on Safety-Critical Software Systems J Systems Software, 26, 103 - 115.
- GRADY, R B AND CASWELL, D L (1987) Software Metrics.' Establishing a Company Wide Program. Prentice Hall, 288pp.
- GUILAIN, S and VIGNES, J (1994) Validation of numerical software results - Application to the computation of apparent heat release in direct-injection diesel engines Mathematics and Computers in Simulation, 37, 73 - 92.
- HAMBLING, B (1996) Managing Software Quality. McGraw Hill. 193pp. HAYLEY A, J and PIRBHAI, I A (1987) Strategy for Real Time System Specification, New York, Dorset House.
- HEIL J H (1992) Chapter 15 in Handbook of software Quality Assurance, 2nd edition (editors Schulmeyer, G.L

- and McManus, I.J) Van Nostrand Reinhold, New York, 343 - 397.
- HINLEY, D S (1996) Software evolution management.' a process-oriented perspective *Information and Software Technology* 38, 723 - 730.
  - HUGUE, MH and SCALZO, R C (1995) Speaking Fault Tolerance in Large Complex Computing Systems Proc First IEEE International Conference on Engineering of Complex Computer Systems (FortLauderdale, Florida, USA 6-10 November 1995), 369-372.
  - ICHIYEN, N M and JOANNAU, P K (1995) Safety critical software design approaches developed for Canadian nuclear power plants *Kerntechnik*, 60, 232 - 237.
  - IDA, N (1996) Verification of Computational Electromagnetic Programs *Int J, Numerical Methods: Electronic Networks, Devices and Fields*, 9, 159 - 165.
  - JACOBSON, I (1992) *Object-oriented Software Engineering - A Use Close Driven Approach*, Addison Wesley, 524pp.
  - JOHNSON, C W (1995) Using Z to support the design of interactive safety-critical systems *Software EngJ, I O*, 49 - 60.
  - KANG, B H, KIM, H B, CHANG, H S, JEON, J S and PARK, S J, (1995) A Software Engineering Process for Safety- Critical Software Applications *J Korean Nucl Soc*, 27, 85 - 95.
  - KIM, H B and HAN, J B (1995) Validation Testing of Safety-critical Software *J Korean Nucl Soc*. 27, 385-392.
  - KITCHENHAM, B (1987) Towards a constructive quality model, Part I: Software quality modelling, measurement and prediction *Software Eng J*, 2, 105 -113.
  - KITCHENHAM, B and PICKARD, L (1987) Towards a constructive quality model. Part II: Statistical techniques for modelling software quality in the ESPITREQUEST project *Software Eng J*, 2, I 14 - 126.
  - KOTONYA, G and SOMMERVILLE, I (1992) Viewpoints for Requirements Definition *BCS/IEEE Software Eng J*, 1(6), 375 - 387.
  - LEE, Y M, CHO, W J, HAN, K W and PARK, H H (1990) Verification of Nuclide Migration Model (MIMOSA) by Comparing with Other Models *J Korean Nucl Soc*, 22, 304 - 313.
  - LUTZ, R.R (1993) Targeting Safety-Related Errors During Software Requirements Analysis *SIGSOFT Software Eng News*, 18(5), 99-106.
  - MARTIN, T and TSAI, W J (1990) N-Fold Inspection.' A Requirements Analysis Technique *Comm ACM*, 33(2), 221 - 232.
  - MCCALL, J A, RICHARDS P G C and WALTERS, G F (1979) Factors in Software Quality US Rome Air Development Center reports, NTIS Report no AD/A-049-014, AD/A-049-01 5 and ADA-049-059.
  - MUNSON, J C (1990) Software faults, software failures and software reliability modelling *Information and Software Technology* 38, 687 -699.
  - ORBSKES, N, SHI-DER-FRECHETTE, A and BELITZ, K. (1994) : Verification, Validation and Confirmation of Numerical Models in the Sciences *Science*, 203, 641 - 646.
  - OSTROFF, J S (1995) A CASE Tool for the Design of Safety-Critical Software Proc 7th International Workshop on Computer Aided Software Engineering, Toronto, 10 - 14 July 1995, 370 - 380.
  - OULD, M A (1990) Software development under DefStan O0-55: a guide *Information and Software Technology*, 32, I 70 - I 75.
  - REES, C and ODDY, G (1991) Safety Critical Software for Defense Systems: Requirements of interim Defense Standard O0-55 *GEC J Research*, 12(1), 42 -60.
  - ROCHKIND, M J (1975) The Source Code Control System *IEEE Trans Software Eng*, 4(4), 364 - 370.
  - ROYCE, W (1970) Managing the Development of Large Software Systems: Concepts and Techniques Proc IEEE WESTCON (LOS Angeles) 1-9.
  - SCHULMEYER, G L (1992). Chapter 14 in *Handbook of Software Quality Assurance*, 2nd edition (editors Schulmeyer, G.L and McManus, I.J) Van Nostrand Reinhold, New York, 318 - 347.
  - SCHULMEYER, G L and MCMANUS, I J (1992) (editors), *Handbook of Software Quality Assurance*, 2nd edition, Van Nostrand Reinhold, New York, 562 p p .
  - SMITH, D J (1995) *Achieving Software Quality - Including its Application to Safety-related Systems*, 3rd edition. Chapman and Hall, London, 285pp.
  - SOMMERVILLE, I (1996) *Software Engineering*, 5th edition, Addison Wesley, 742pp.
  - THOMAS, M (1993) The industrial use of formal methods *Microproc Microsys*, 17, 31 - 36.
  - TICHY, W (1985) RCS - A System for Version Control *Software Practice and Engineering*, 15(7), 637 - 654.
  - VOAS, J M and MILLER, K W (1995) Using Fault Injection to Assess Software Engineering Standards Proc Second IEEE International Software Engineering Standards Symposium, 21 -25 August 1995, Montreal Canada, pp 139 - 145.
  - VOAS, T M and MILLER, H W (1996) The Avalanche Paradigm.' An Experimental Software Programming

Technique for Improving Fault-tolerance Proc IEEE Symposium and Workshop on Engineering of Computer Based Systems, I I - 15 March 1996, Friedrichshafen, Germany, pp 1421: 147.

- WARD, N J (1993) Integrated Formal Verification and Validation of Safety Critical Software Aerospace Software Eng for Advanced Systems Architectures, AGARD Conf Proceedings No 545, pp 23/1 - 23/1 1 .
- WARDLE, P J (1990) Methodology and Tools for Requirements Capture, Traceability, and Verification 3rd International Conference on Software Engineering in Real Time Systems, IEE (London) ConfPub No 346, pp 46 -50.
- WATKINS, P and NEAL, M ( 1992) Why and How of Requirements Tracing IEEE Software II, 104 - 106.
- Anderson, M.P. 1979. Using models to simulate the movement of contaminants through ground water flow systems. CRC Critical Review in Environmental Control, November 1979.
- Baetsle, L.H. 1969. Migration of radionuclides in porous media. Progress in Nuclear Energy Series XII, Health Physics, ed. A. Duhamel, Elmsford, New York, Pergamon Press.
- Bear, J. 1979. Hydraulics of Groundwater. McGraw-Hill, New York, N.Y.
- Bear, J. and A. Verruijt. 1987. Modeling Groundwater Flow and Pollution.
- Bear, J. and A. Bachmat. 1990. Introduction to Transport Phenomena in Porous Media. D. Reidel Publishing Company, Dordrecht, Holland.
- Beljin, M. 1985. SOLUTE: Analytical Models for Solute. Transport in Porous Media. IGWMC, Indianapolis, Indiana.
- Carnahan, C.L. and J.S. Remer. 1984. Nonequilibrium and equilibrium sorption with a linear sorption isotherm during mass transport through an infinite porous medium: Some analytical solutions. J. of Hydrology, vol. 73, pp. 227-258.
- Chambre, P.L., T.H. Pigford, A. Fujita, T. Kanki, A. Kobayashi, H. Lung, D. Ting, Y. Sato, and S.J. Zavoshy. 1982. Analytical performance models for geological repositories. LBL-14842, Lawrence Berkeley Laboratory, Berkeley, California.
- Chambre, P.L., T.H. Pigford, W. W. Lee, J. Ahn, and S. Kajiwara. 1985. Mass transfer and transport in a geologic environment. LBL-19430, Lawrence Berkeley Laboratory, Berkeley, California.
- Clearly, R.W., and D.D. Adrian. 1973. New analytical solutions for dye diffusion equations. J. of the Environmental Eng. Div., ASCE, vol. 99, no. EE3, pp. 213-227.
- Clearly, R. W. 1978. Final report 208 project. Nassau-Suffolk Regional Planning Board, Hauppauge, New York.
- Clearly, R. W. and M.J. Unga. 1978. Groundwater pollution and hydrology, mathematical models and computer programs. Report 78-WR-15, Princeton University, Princeton, New Jersey.
- Codell, R.B. and D.W. Shreiber. 1979. NRC models for evaluating the transport of radionuclides in groundwater. In Management of Low Level Radioactive Waste, M.W. Carter, A.A. Moghissi, and B. Kahn editors, Pergamon Press.
- Codell, R.B., K. T. Key, and G. Whelan. 1982. A collection of mathematical models for dispersion in surface water and groundwater. NUREG-0868, U.S. Nuclear Regulatory Commission, Washington, D.C.
- Cole, C.R. et al. 1987. Lessons learned from the HYDROCOIN experience. Presented at GEOVAL-87, April 7-9, Stockholm, Sweden.
- Collette, R. 1985. A three dimensional model for simulating flow and mass transport in ground-water systems. M.S. Thesis, Dep. Geol., Univ. of Alberta, Edmonton, Alberta, 106 pp.
- de Marsily, G. 1986. Quantitative Hydrogeology: Groundwater Hydrology for Engineers. Academic Press, Inc., San Diego, California.
- Dolman, E.A. and P.C. Robinson. 1983. UKAEA Report AERE-R.10882, Harwell, Oxfordshire.
- Domenico, P.A. and V.V. Palciauskas. 1982. Alternative boundaries in solid waste management. Ground Water. vol. 20, pp. 303-311.
- Domenico, P.A. and G.A. Robbins. 1984. A dispersion scale effect in model calibrations and field tracer experiments. J. of Hydrology. vol. 70, pp. 123-132.
- Domenico, P.A. and G.A. Robbins. 1985. A new method of contaminant plume analysis. Ground Water. vol. 23, no. 4, pp. 476-485.
- Domenico, P.A. 1987. An analytical model for multidimensional transport of a decaying contaminant species. J. of Hydrology. vol. 91, pp. 49-58.
- Domenico, P.A. and F.W. Schwartz. 1990. Physical and chemical hydrogeology. John Wiley and Sons, New York.
- El-Kadi, A., S.A. Williams, and P.K.M. van der Heijde. 1987. Applicability of the Screening Level Groundwater Model to Site-Specific Settings.
- IGWMC, Holcomb Research Institute, Indianapolis, Indiana. 90 pp. EPRI. 1984. Geohydrochemical models for solute migration, Vol. 2: preliminary evaluation of selected computer codes. EPRI EA-3417, Electric Power

Research Institute, Palo Alto, California.

- Freeze, R.A. and J.A. Cherry. 1979. Groundwater. Prentice-Hall, Englewood Cliffs, N.J. 604 pp.
- Galya, D.P. 1987. A horizontal plane source model for ground-water transport. Ground Water. vol. 25, no.6, pp. 733-739.
- Goltz, M.N. 1986. Three-Dimensional Analytical Modeling of Diffusion Limited Solute Transport. Ph.D. Thesis, Stanford University, Stanford, California. 171 pp.
- Goltz, M. and P. Roberts. 1986. Three-dimensional solutions for solute transport in an infinite medium with mobile and immobile zones. Water Resources Research. vol. 22, no. 7, pp. 1139-1148.
- Gupta, S.K. et al. 1987. Coupled fluid, energy, and solute transport (CFEST) model: Formulation and user's manual. BMI/ONWI-660, Battelle Memorial Institute, Columbus, Ohio.
- Gureghian, A.B. 1987. Analytical solutions for multidimensional transport of a four-member radionuclide decay chain in ground water. Battelle Office of Crystalline Repository Development, Columbus, Ohio.
- Gureghian, A.B. 1988. MASCOT: Analytical solutions for multidimensional transport of a four-member radionuclide chain in ground water. Battelle Memorial Institute, Columbus, Ohio.
- Herbert, A.W. 1984. Analytical solutions to the three-dimensional radionuclide transport equation for computer code verification. AERE Harwell, Oxfordshire.
- Prakash, A. 1984. Ground-water contamination due to transient sources of pollution. J. of the Hydraulics Div., ASCE, vol. 110, no. 11, pp. 1642-1658.
- Robbins, C.A. and P.A. Domenico. 1984. Determining dispersion coefficients and sources of model dependent scaling. J. of Hydrology, vol 75, pp. 195-211.
- Sagar, B. 1982. Dispersion in three-dimensions: approximate analytical solutions. J. of the Hydraulics Div., ASCE. vol. 108, no. HY1, pp. 47-62.
- Shen, H.T. 1976. Transient dispersion in uniform porous media flow. J. of the Hydraulics Div., ASCE. vol. 102, no. HY6, pp. 707-716.
- Sinunons, C.S. and C.R. Cole. 1985. Guidelines for selecting codes for ground-water transport modeling of low-level waste burial sites. Pacific Northwest Laboratory PNL-4980, Richland, Washington.
- U.S. EPA. 1984. AT123D execution using the data management supporting systems AT123DIN and AT123DOUT, User's Guide. PB86-140634, U.S. EPA, OSW, Washington, D.C.
- U.S. EPA. 1985. Background document for the ground water screening procedure to support 40 CFR part 268, Land Disposal Restrictions, U.S. EPA, OSW, Washington, D.C.
- van der Heijde, P.K.M. 1989. EPA- VHS: Analytical Solute Transport Model. International Ground Water Modeling Center, Indianapolis, Indiana.
- van der Heijde, P.K.M. 1990. PLUME 2.0: A Three-dimensional Analytical Solute Transport Model. International Ground Water Modeling Center, Indianapolis, Indiana.
- van Genuchten, M.Th. and W.J. Alves. 1982. Analytical solutions of one-dimensional convective dispersive solute transport equation. USDA Technical Bulletin No. 1661, Riverside, California.
- Wagner, J., S. Watts, and D.C. Kent. 1985. PLUME3D: Three-dimensional plumes in uniform ground water flow. EPA Report EPA/600/2-85/067,
- Robert S. Kerr Environmental Research Laboratory, Ada, Oklahoma.
- Walton, W. 1979. Progress in analytical groundwater modeling. In: W.Back and D.A. Stephenson (Editors), Contemporary Hydrogeology, J. Hydrology, vol. 4, pp. 149-159.
- Walton, W. 1985. WALTON35: Microcomputer programs. International Ground Water Modeling Center, Indianapolis, Indiana.
- Walton, W. 1988. Practical Aspects of Ground Water Modeling. National Water Well Association (NWWA), Columbus, Ohio.
- Wexler, E.J. 1989. Analytical solutions for one-, two-, and three-dimensional solute transport in ground-water systems with uniform flow. U.S. Geological Survey, Open-File Report 89-56, Tallahassee, Florida.
- Wolanski, E.J. 1973. Discussion of a Dispersion from pit in uniform seepage, by B.W. Hunt, J. of the Hydraulics Div., ASCE, vol. 99. no. HY7, pp. 1198-1199.
- Yeh, G.T. and Y.J. Tsai. 1976. Analytical three-dimensional transient modeling of effluent discharges. Water Resources Research. v. 20, pp. 533-546.
- Yeh, G.T. 1981. AT123D: Analytical transient one-, two-, and three-dimensional simulation of waste transport in the aquifer system. Oak Ridge National Lab., Report No. ORNL-5602. 83 pp.