

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（Ⅱ）

（動力炉・核燃料開発事業団 研究概要）

1998年2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

## 地層処分ヴァーチャル・エンジニアリング・ システムの開発 (II)

—概要—

浅原進也\*)

### 要 旨

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基に、オブジェクト指向、並列計算法等、革新するソフトウェア工学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。同システムを開発・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)に亘る現象の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9年度は、前年度に整理したソフトウェア工学面の技術と試作システムを基にヴァーチャル・エンジニアリング・システム全体について基本的な設計検討を行った。設計作業に伴い、確認に必要な技術は試作を行い評価した。技術確認は既存のシミュレーション・コード出力をビジュアルに統合するインターフェイスに重点をおいた。そのため既に開発されている核種移行評価コード『RELEASE』、熱・多相流体連成解析コード『TOUGH2』、3次元地下水流れ評価コード『3D-FEMWATER』コードを用いて異なったモデルにより検討した。

検討の結果、対象としたコードは、インターフェイス・プログラムを準備することにより、開発中のヴァーチャル・エンジニアリング・システムの視覚化機能で対応でき、3次元ビジュアル・シミュレーションを行うことが確認できた。研究開発の結果、より詳細に連携したシミュレーション・システム全体の設計検討と、対話機能、データベース機能、ソフトウェア品質管理機能等を提供するサポート・システムの開発が今後の課題として残された。

---

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：090D0293

事業団担当部課室および担当者：環境技術開発推進本部 処分研究グループ 梅木 博之

\*) 日商岩井株式会社

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

## Summary

Shinya ASAHARA \*)

## Abstract

The Virtual Engineering System for Geological Isolation is a dedicated and integrated computer system to provide a basis of argument and synthetic evaluation, from many directions, for the study of safety assessment of radioactive waste geological disposal. For the system, the latest computer technologies such as object oriented programming, parallel computing and other state of the art are applied. 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) proving 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 and interface between scientist/engineer and public through the use of easily understandable visualization technology. The preliminary scope of studies are; 1) interactive Graphic User Interface for future synthetic performance assessment; 2) visualized simulation technology which is combined the coupled model (cooperative, parallel) computation technology, which might be defined as middle range of complexity between individual deterministic analysis and stochastic analysis, and visualization technology; 3) multi-database development technology which inter-relates various objects, models and scenario; 4) software quality assurance technology.

In fiscal year 1997, fundamental design for the total system of Virtual Engineering System was studied based upon the last year achievement of software engineering technologies. During the study, the technologies which needs to be evaluated practically were prototyped. Especially, the interface of the system to visualize output of analytical codes and the visualization management were developed. Existing different models were used for the study, such as nuclear transport modeling code "RELEASE" heat/multi-phase flow analyzing coupled code "TOUGH2" and 3D groundwater flow modeling code "3DFEMWATER". As the result of the studies, the selected codes were confirmed that they can be adopted to the Virtual Engineering System as visualized 3D simulation by developing the interface programs. For the next studies, more detailed simulation system, dialogue function, database function, software quality management function will be needed for the total system design evaluation.

---

Work performed by Nissho Iwai corporation under contract with Power Reactor and Nuclear Fuel Development Corporation.

PNC Liaison : Radioactive Waste Management Project, Isolation System Research Program  
Hiroyuki Umeki

\*) Nissho Iwai corporation



# 【 目 次 】

1	概 要 .....	1
2	対話機能の概念設計 .....	6
2.1	GUI の全体構成 .....	6
2.2	Console GUI 仕様 .....	7
2.3	PSA Palette GUI 仕様 .....	8
2.4	3D Viewer GUI 仕様 .....	9
2.5	Parameter Display GUI 仕様 .....	10
2.6	DB Mng GUI 仕様 .....	12
2.7	S/W QC GUI 仕様 .....	15
2.8	対話機能に係わる認知・知識工学的技術等の応用について .....	16
3	視覚化複合事象シミュレーション機能 統合型シミュレーションの概念設計 .....	20
3.1	視覚化複合事象シミュレーション機能（エンジン）基本設計 .....	20
3.2	関連研究課題1：評価シナリオ・クラス概念設計 .....	30
3.3	関連研究課題2：評価モデルのクラス接続技術研究 .....	31
3.4	関連研究課題3：モデル・共通 3D 動画化プロセス間並列データ通信機能研究 .....	32
3.5	関連研究課題4：人工バリア核種放出シナリオ・シミュレーション試作 .....	33
3.6	関連研究課題5：ニアフィールド核種放出シナリオ・シミュレーション試作 .....	38
3.7	関連研究課題6：地層中核種移行シナリオ・シミュレーション試作 .....	48
4	データベース管理機能の基本設計 .....	66
4.1	データベース管理機能の基本設計 .....	66
4.2	データベース・モデル .....	69
4.3	データベース管理システム機能仕様 .....	71
4.4	オブジェクト指向技術の応用 .....	73
4.5	シミュレーション・シナリオ・データベース試作 .....	78
4.6	シミュレーション・モデル・データベース試作 .....	81
4.7	シミュレーション・オブジェクト・データベース試作 .....	84
4.8	試験データ等、数値・文字データベース試作 .....	85
4.9	品質管理データベース試作 .....	85
5	ソフトウェア開発管理機能の概念設計 .....	86
5.1	ソフトウェア開発管理機能概念設計 .....	87
5.2	プログラム品質管理機能試作 .....	91
5.3	ソフトウェア品質管理機能の試作 .....	94
6	実証・評価用システムの改良整備 .....	98
7	まとめ .....	102

## 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（以下、本システムと記す）の開発目標は以下のとおりである。

### —目標—

本システムを開発・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関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シナリオを複雑に組み合わせ、地層処分システムの長期的挙動の評価を行い、地層処分の安全性に関わる評価技術を判断しながら処分システムの設計の最適化を図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を本プロジェクトの目標とする。また、将来の科学技術として考えられている計算科学技術分野の基盤技術開発に資するものとして、本研究開発の知見を整備することもあわせて目的とする。

平成9年度（以下、本年度と記す）は、前年度に整備した基盤技術を基に、地層処分ヴァーチャル・エンジニアリング・システムの概念設計を行い、次年度に実施する詳細設計の準備として以下の分野の設計開発を実施した。作業にあたっては、前年度に試作した評価用プロトタイプ・システムを改良整備し、設計技術確認等に利用した。

### 分野1：対話機能の概念設計

前年度に引き続き、地層処分システムの長期的な挙動を提示するための作図、動画、画像処理、対話機能などといったコンピュータと人間との対話（情報交換）を司る機能を研究し、本システムの対話機能に係る部分の設計を行った。（図1-1参照）



図1-1 対話画面の例

## 分野2：視覚化複合事象シミュレーション機能の概念設計

前年度に引き続き、地層処分システムの安全性評価に関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などといった、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開発を行い、本システムのシミュレーション機能に係る部分の設計を行った。また、別途開発している地層処分FEP・データベース・システムとのインターフェースの概念を設計した。（図1-2参照）

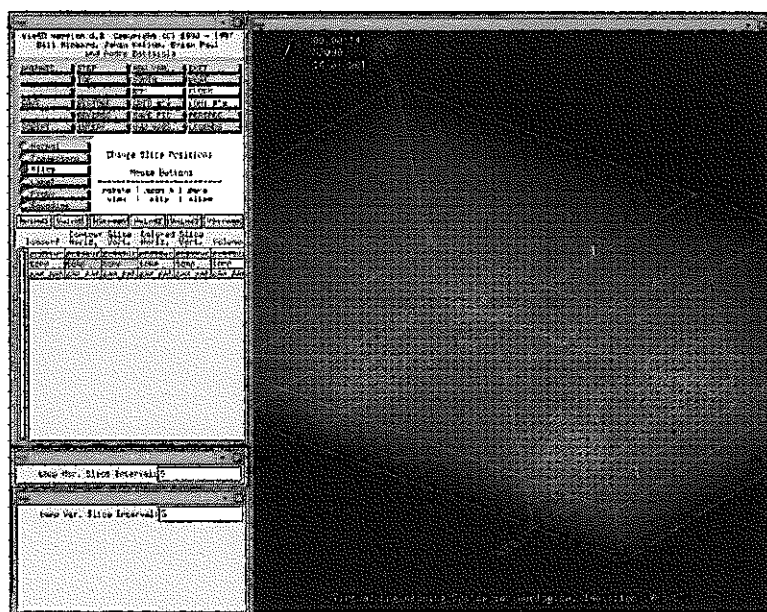


図1-2 廃棄物4体の3次元2層（地下水・ガス）温度シミュレーションの例

## 分野3：データベース管理機能の概念設計

前年度に引き続き、本システム全体の基盤となる、膨大な各種情報モデルを集約した複合化データベースの研究開発を行い、本システムのデータベース管理機能に係る部分の設計を行った。（図1-3参照）

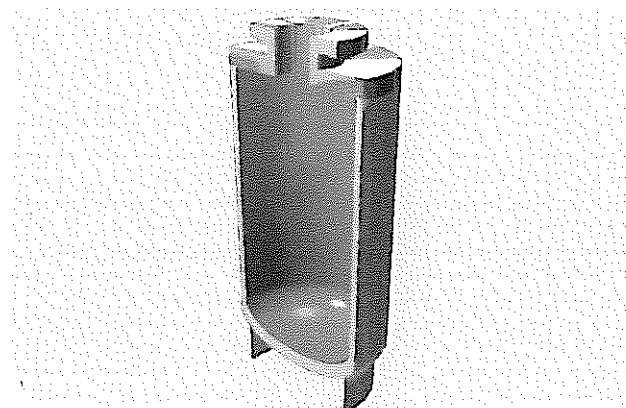


図1-3 データベース管理情報区分1：設計オブジェクトの例

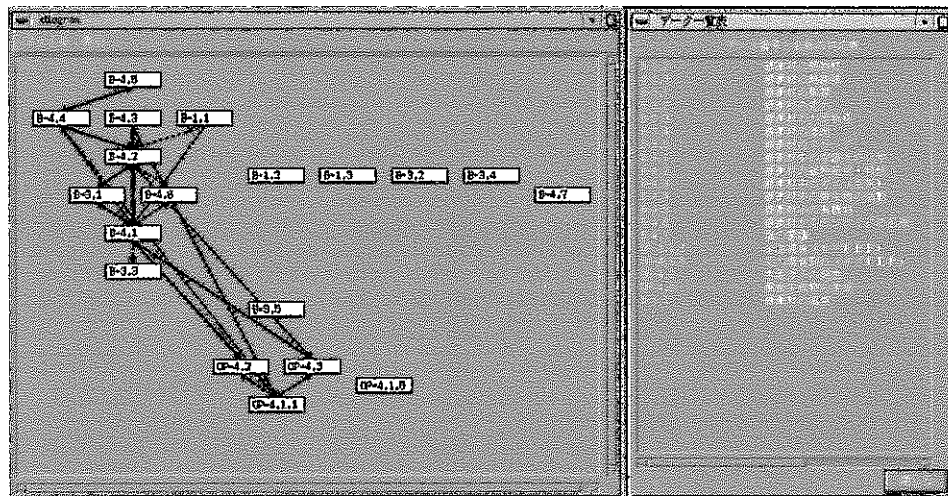


図 1-4 データベース管理情報区分 2 : FEP情報の例

```

4* select des from DES where fepno = 'H-4.1';
des
ニアフィールド母岩中の地下水は、フェアフィールドと同様に地球化学的変遷を経るとともに、他分場の建設や操業時にニアフィールド母岩中に取り込まれた酸堿や二酸化炭素と反応しながら、ペントナイト中に侵入することが想定される。
ニアフィールド母岩中の地下水は、フェアフィールドと同様に地球化学的変遷を経るとともに、他分場の建設や操業時にニアフィールド母岩中に取り込まれた酸堿や二酸化炭素と反応しながら、ペントナイト中に侵入することが想定される。
5*

```

図 1-5 データベース管理情報区分 3 : 文字情報の例

#### 分野 4 : ソフトウェア開発管理機能の概念設計

前年度に引き続き、本システム構築に必要なソフトウェアの設計、開発に関わる管理・品質保証を行う手法の研究開発を行い、本システムのソフトウェア開発管理機能に係る部分の設計を行った。

## 分野5：実証・評価用システムの改良整備

前年度に引き続き、本システムの設計・開発において必要となる、対話機能やシミュレーション機能など要素技術の試作・評価等に利用するため、専用の技術開発・実証用コンピュータ・システムの改良整備・構築を行った。

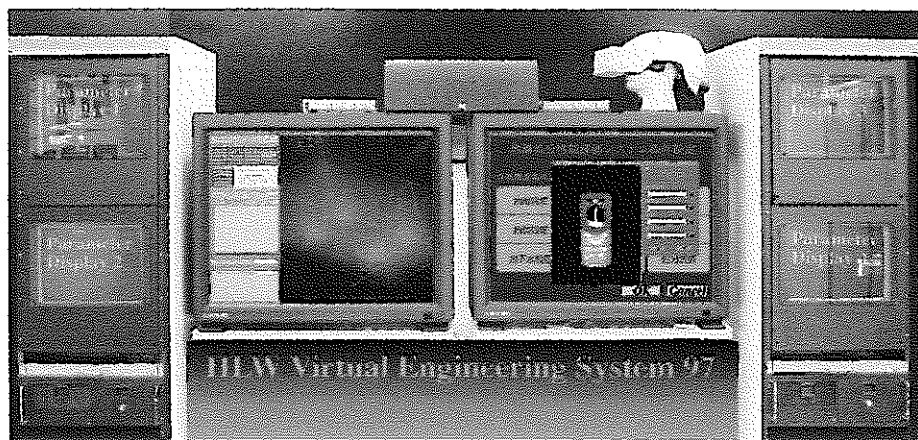


図1-6 平成9年度に改良したシステムの外観

ヴァーチャル・エンジニアリングは、ソフトウェアのインテグレーションと管理の技術であると考えられる。個々の部品は、例えば、個別の現象解明等の研究成果であるソフトウェア（知識、技術、データ、プログラム、文献等）や、それらを複雑に組み合わせてビジュアルなシミュレーションを行うためのグラフィクス、データベースおよびネットワーク技術が基礎となると考えられる。したがって、ヴァーチャル・エンジニアリングは、情報処理システム・インテグレーション技術をその基盤としている。

### ソフトウェア・インテグレーション技術

ソフトウェア・インテグレーションの課題は、従来開発してきたソフトウェアをどのように部品化することが望ましいか、部品化されたソフトウェアを再編成して大規模なシステムを開発するにはどのようにしたらいいのかの2通りと思われる。さらに、『大規模連成コードの新規開発』等の課題がでてくるかも知れないが、その前になぜ新規に開発しなければならないのかを示す必要がある。

また、問題が特定した現象レベルのものであった場合、それは従来型のR&Dの領域に属するものなのか、それともソフトウェア・エンジニアリングを基盤とするヴァーチャル・エンジニアリングの領域に属するものなのかについて明確にしておく必要があると思われる。いずれにしても、当面、ヴァーチャル・エンジニアリングを特徴付けるソフトウェア・インテグレーションとは、ソフトウェアのアセンブリ技術を開発することであり、部品そのもの（個別の解析コード等）を開発することではないと考えられる。

### ソフトウェア管理技術

ヴァーチャル・エンジニアリング研究開発におけるもう一つの重要な特徴として、ソフトウェア

管理工学が挙げられる。一般に、長い期間に亙って数値計算コードを中心に検証・確証や移植を行ってきた来た技術者は、バグの無いソフトウェアは存在しない事を知っている。また、ソフトウェアは、必要以上に手間のかかる対象とも感じているだろう。それらは、皆、野放し状態の開発管理（職人芸）が原因と思われる。したがって、ヴァーチャル・エンジニアリングを実現するための第一歩の作業の一つとして、ソフトウェア管理基盤を確立することは重要な課題と考えられる。

ここに言う管理とは、『手法に関する工学：メソッド・エンジニアリング』である。このメソッド・エンジニアリングも、前述のインダストリアル・エンジニアリングの一つの領域である。ソフトウェアの管理手法の開発においては、定量的な手法が開発されることが重要と思われる。定量的手法はアルゴリズムがあり、そのような対象は、プログラム可能でヴァーチャル・エンジニアリング・システムの機能として実現可能と考えられる。

## 2 対話機能の概念設計

前年度に引き続き、地層処分システムの長期的な挙動を提示するための作図、動画、画像処理および対話機能などといったコンピュータと人間との対話（情報交換）を司る機能を研究開発し、本システムの対話機能に係る部分の設計を行った。設計した機能部分は以下の通りである。

以下、設計した対話プログラムをGUI（Graphic User Interface）と記す。

### 2.1 GUIの全体構成（準拠標準：OSF/Motif、IRIS4D/GL）

図2. 1-1 にシステム全体の機能セグメント構造を示す。図2. 1-1 に示す通り本システムのユーザーインターフェイスは、次のセグメントから構成される。それらは

- ・モデル選択セグメント
- ・オブジェクト（ステージ、シーン）選択セグメント
- ・シナリオ・キーワード選択セグメント
- ・シナリオ階層マトリクス選択セグメント

である。

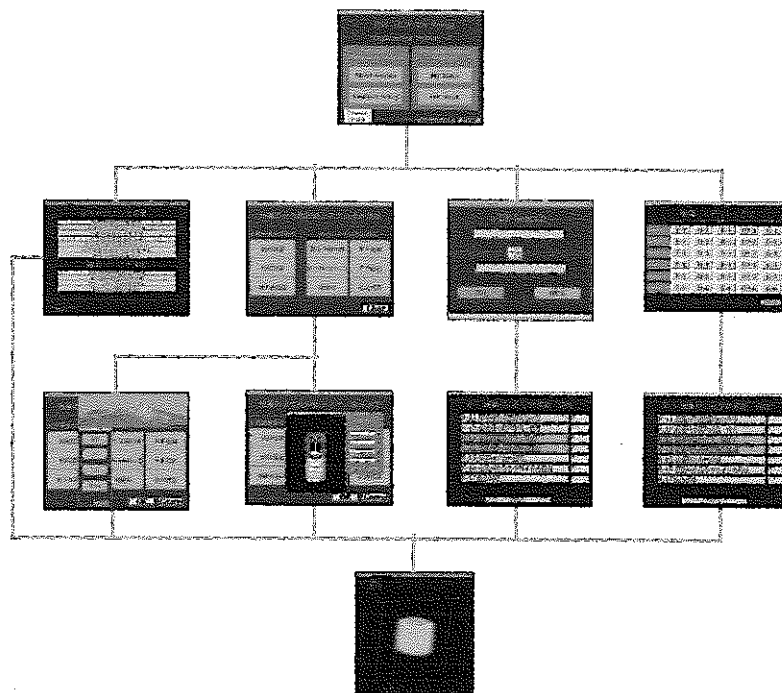


図2. 1-1 GUI全体構成概要

## 2. 2 ConsoleGUI仕様 (準拠標準 : OSF/Motif、IRIS4D/GL)

本システム全体を操作するためのGUIセグメントとした。図2. 2-1に起動画面を示す。

### (1) 画面配置

図2. 2-1に起動画面を示す。図2. 2-1に示すように起動画面は2分野の情報について下記の機能選択ボタンを設定した。それらは、

- ・性能評価解析  
モデル選択ボタン  
オブジェクト (ステージ、シーン) 選択ボタン
- ・地層処分シナリオデータベース  
階層マトリクス選択ボタン  
地層処分キーワード選択ボタン

である。

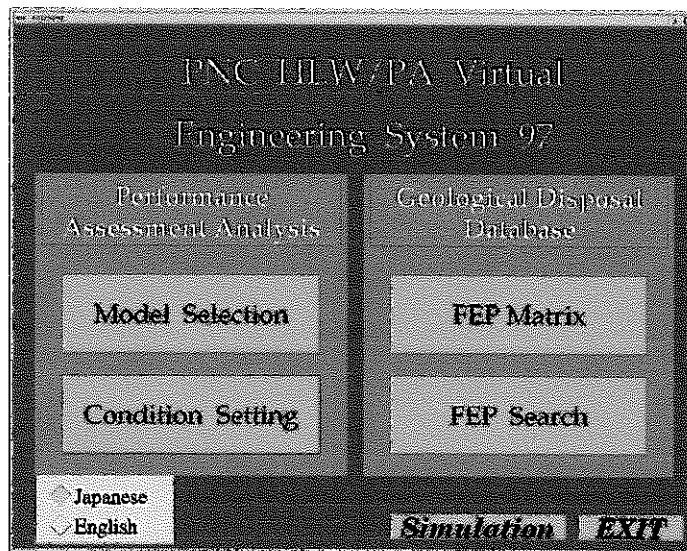


図2. 2-1 起動画面

### (2) GUIプログラム仕様

図2. 2-1に示す画面制御プログラムを実装するシステムの開発環境と制御条件は下記のとおりである。

- |          |   |
|----------|---|
| プラット・ホーム | : Silicon Graphics Indigo2 Extreme、96MB Memory  |
| 開発言語     | : C Compiler version 7.2  |
| 開発環境     | : IRIX 6.2、X window 6   |
| 協調制御     | : Windows-95と連携 (Parameter Display) と連携<br>Silicon Graphics OCTANE (3DViewer) と連携<br>Silicon Graphics O2 (Data Base System) と連携 |



## 2. 3 PSA PaletteGUI仕様 (準拠標準 : OSF/Motif, IRIS4D/GL)

確率論的評価を行う上で必要なネットワークやモデルの作図・更新・保管および計算を操作するためのGUIセグメントとした。図2. 3-1に起動画面を示す。

### (1) 画面配置

図2. 3-1にPSA操作画面の例を示す。図2. 3-1に示すようにPSA操作画面は頻度評価のための事象のツリーに個別の規模評価を関連させるネットワーク形式の作図機能を想定した。動作は、このようにして作図したボタンのコールバック制御で個別のコードの起動を行う仕様とした。

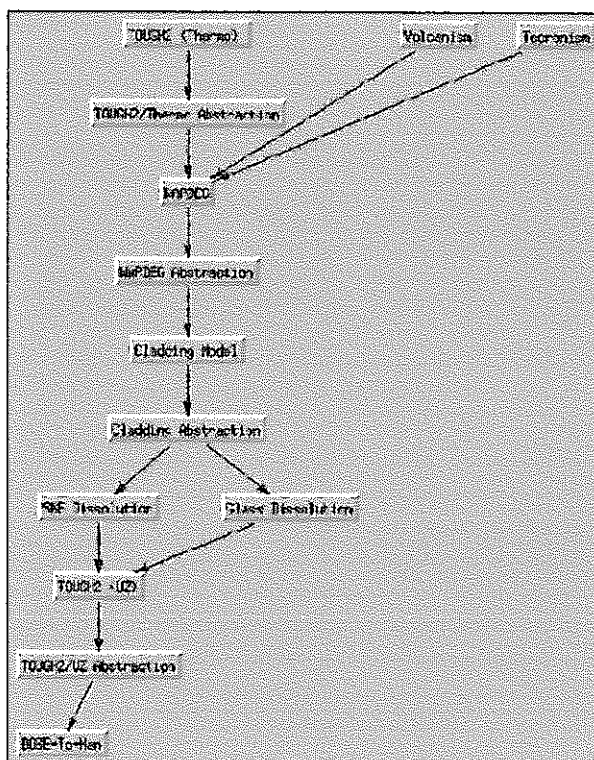


図2. 3-1 PSA操作画面例

### (2) GUIプログラム仕様

図2. 3-1に示す画面制御プログラムを実装するシステムの開発環境と制御条件は下記のとおりである。

プラットフォーム	: Silicon Graphics Indigo2 Extreme、96MB Memory
開発言語	: C Compiler version7.2
開発環境	: IRIX 6.2、X window 6
協調制御	: Silicon Graphics O2 (Data Base System) からデータ取得

## 2. 4 3DViewerGUI仕様（準拠標準：OSF/Motif、IRIS4D/GL、IRIS VRML）

数値計算を実行している解析コードから、シミュレーション結果を即座にネットワークを介して並列的に受信し、3次元動画像として再現するためのGUIセグメントとした。また物理量などの表示については、ボリューム・レンダリング、ベクトル、等高線等、理解しやすい表示形式の選択が行えるように設計した。（図2. 4-1）

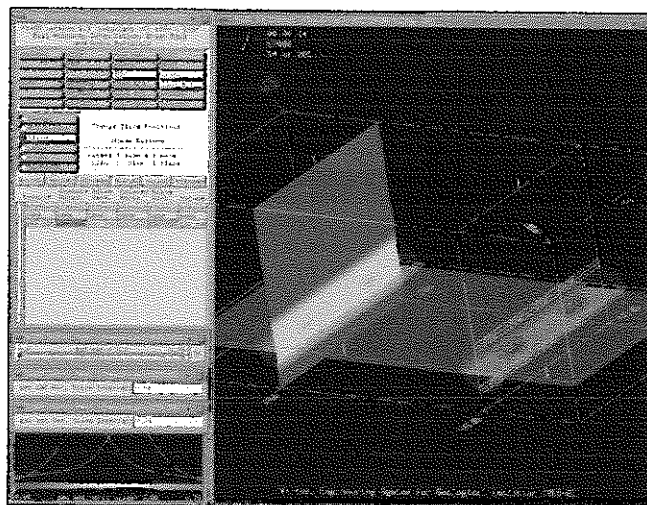


図2. 4-1 熱・二相地下水連成シミュレーション視覚化処理例

また、設計オブジェクトの景観シミュレーション等を行うための機能も設計評価した。（図2. 4-2）



図2. 4-2 景観シミュレーション（設計施設）例

それにより、本システムのシミュレーションの範囲を、景観シミュレーション、動作シミュレーションおよび現象シミュレーションとして扱い、設計オブジェクトから評価まで一貫した可視化技術の基盤を整備した。図2. 4-3に動作シミュレーションの例を示す。

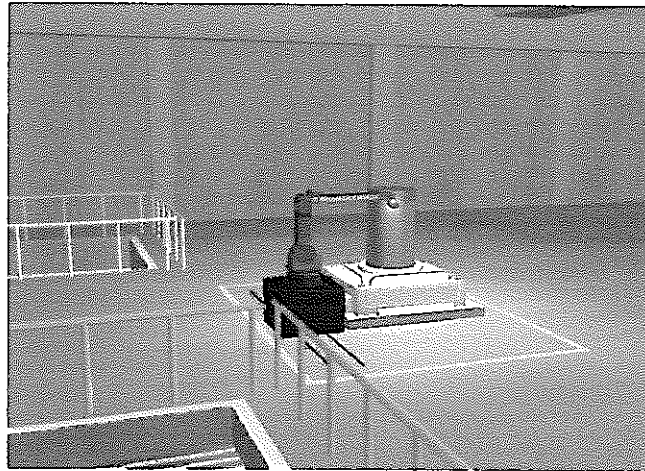


図 2. 4-3 動作シミュレーション（溶接ロボット）例

## 2. 5 Parameter DisplayGUI仕様（準拠標準：Windows-95）

計算中の現在時刻、評価対象のシナリオやシミュレーション条件など、シミュレーションの実行に伴い表示が必要なパラメータを表示するための対話機能セグメントとした。本セグメントは、UNIX Systemで実行されるシミュレーション・プロセスのスレーブとして、TCP/IPソケット経由でシミュレーション・パラメータを受信し、ローカルなWindows-95の画面サーバーに表示する設計とした。

### (1) メニュー画面配置

図 2. 5-1 に、ステージパラメータの表示例を示す。

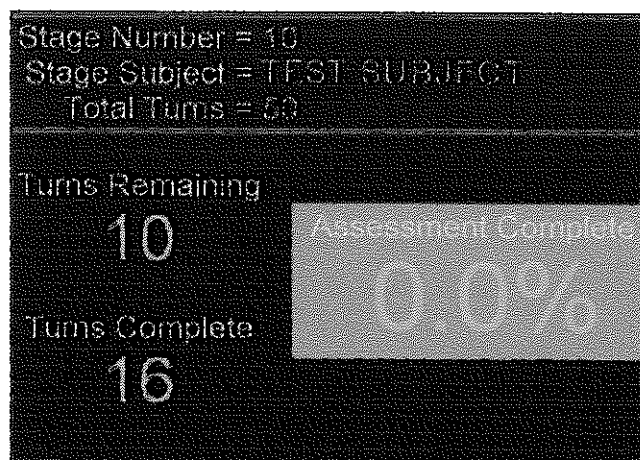


図 2. 5-1 ステージ・パラメータ表示例

図 2. 5-2 に、時刻パラメータの表示例を示す。

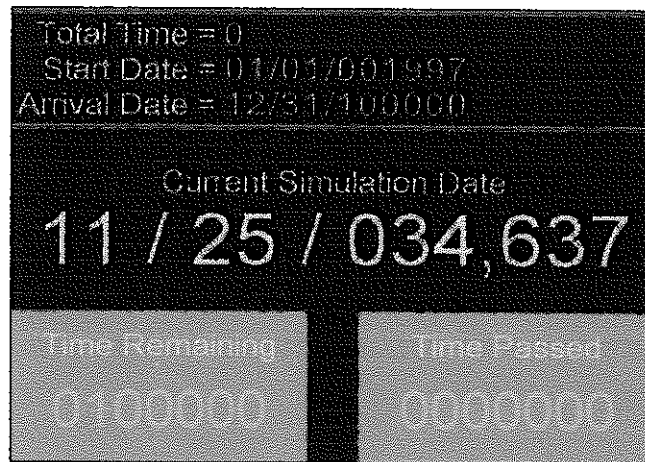


図 2. 5-2 時刻パラメータの表示例

図 2. 5-3 に、選択シナリオの表示例を示す。

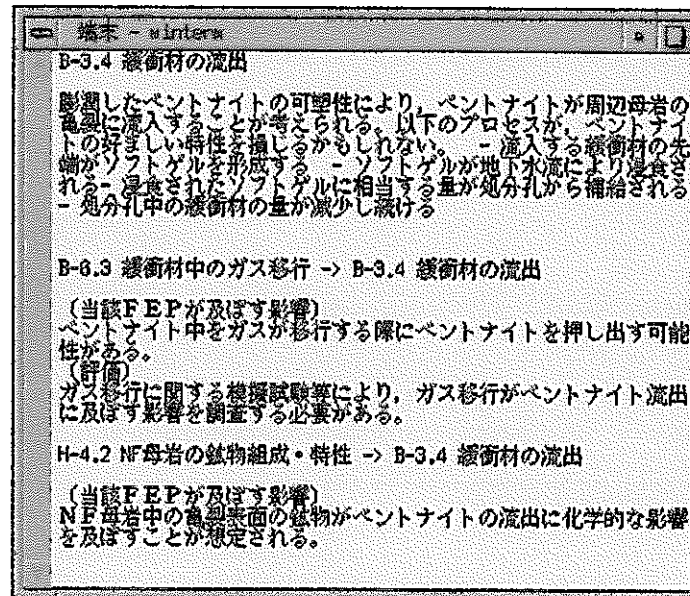


図 2. 5-3 選択シナリオの表示例

## (2) GUIプログラム仕様

図 2. 5-1 から図 2. 5-3 に示す画面制御プログラムを実装するシステムの開発環境と制御条件は下記のとおりである。

- プラット・ホーム : Silicon Graphics OCTANE、512MB Memory
- 開発言語 : C Compiler version 7.2
- 開発環境 : IRIX 6.4、X window 6
- 協調制御 : Silicon Graphics Indigo2 Extreme (Main System) から起動・停止制御  
 Silicon Graphics OCTANE (3D Viewer) からデータ表示制御  
 Silicon Graphics O2 (Data Base System) からデータ表示制御

## 2. 6 DB Mng GUI仕様 (準拠標準 : OSF/Motif、IRIS4D/GL)

本システムの情報基盤となるデータベースの保守や、直接アクセスに供するGUI機能セグメントとした。設計にあたっては、オブジェクト指向データベースとリレーショナル・データベースを統合的に管理できるものとして検討した。

### (1) 画面配置

図2. 6-1に設計したオブジェクト指向データベース画面の例を示す。同様に、図2. 6-2に設計したリレーショナル・データベース画面を示す。

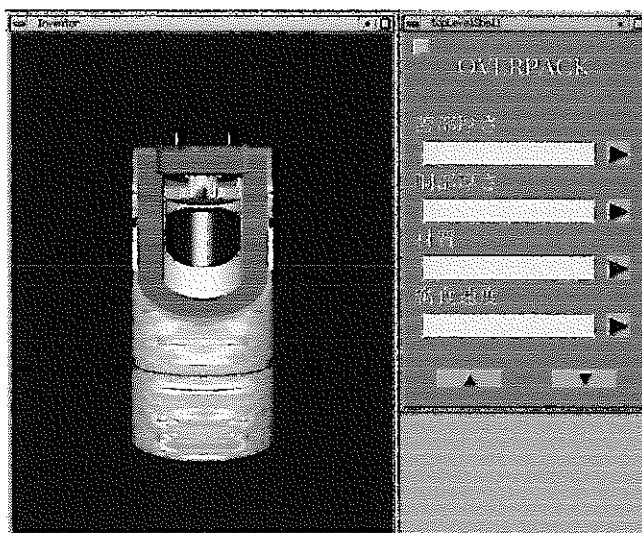


図2. 6-1 オブジェクト指向データベースの例

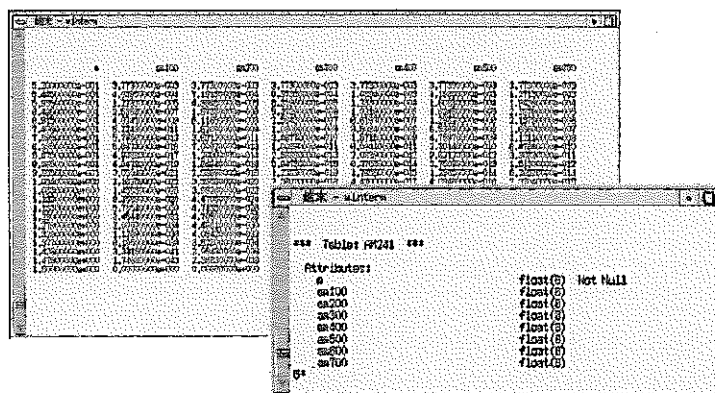


図2. 6-2 リレーショナル・データベースの例

図2. 6-3に、データベースに試験的に登録した衛星観測データ管理画面を示す。

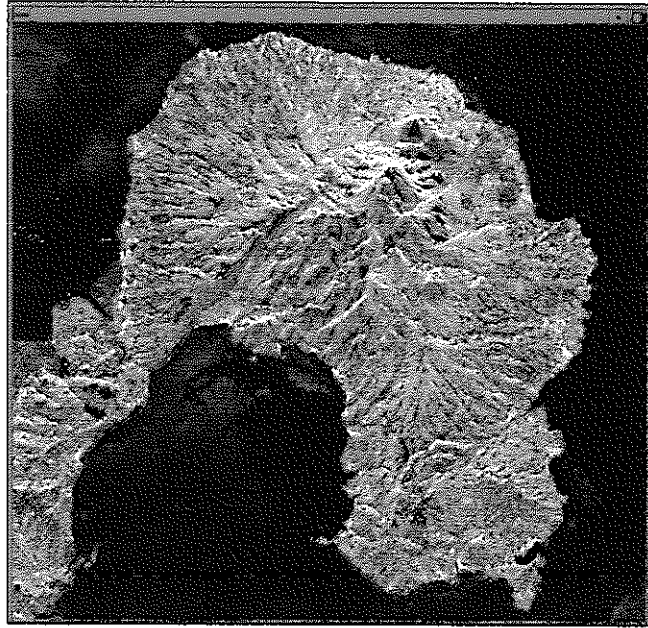


図 2. 6-3 衛星観測データ（写真）の管理画面例

図 2. 6-4 に、データベースに試験的に登録した地上観測データ管理画面を示す。

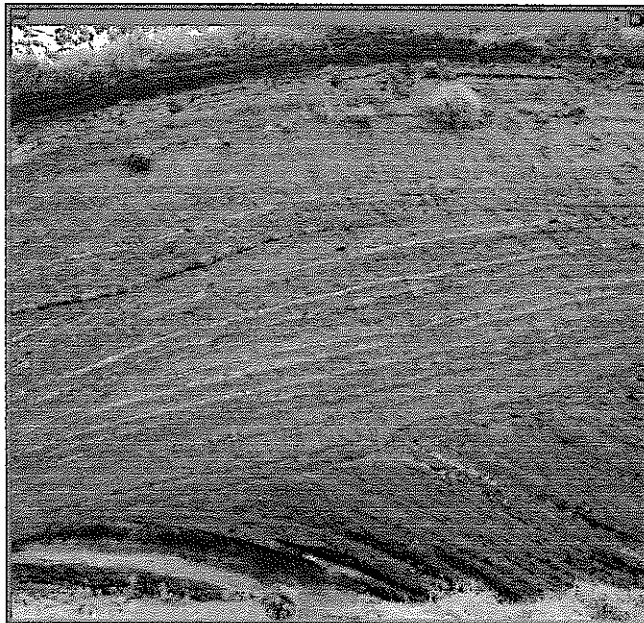


図 2. 6-4 地上観測データ（写真）の管理画面例

図 2. 6-5 に、解析コード管理画面を示す。

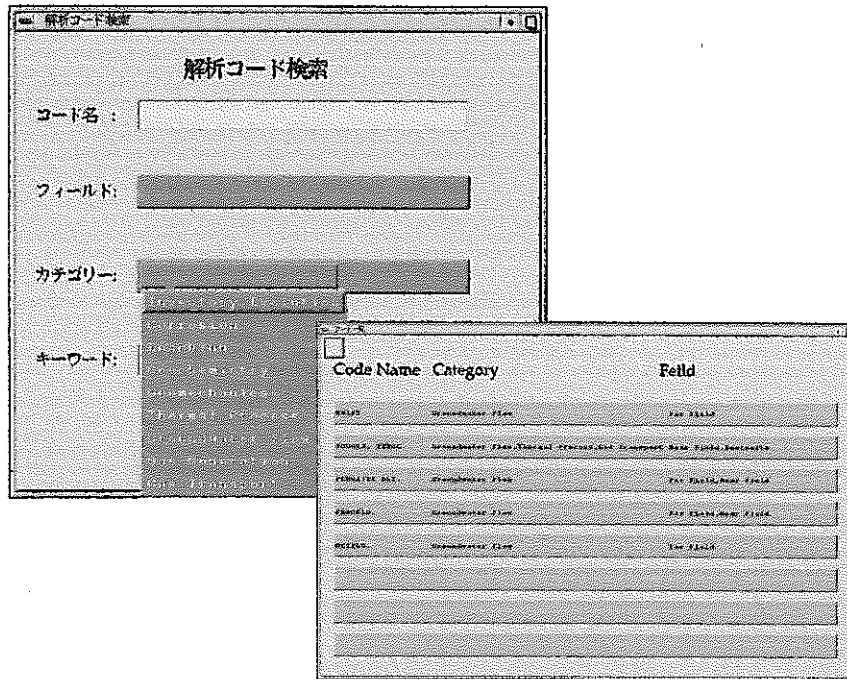


図 2. 6-5 解析コード管理画面

## (2) GUI プログラム仕様

図 2. 6-1 から図 2. 6-5 に示す画面制御プログラムを実装するシステムの開発環境と制御条件は下記のとおりである。

- プラットフォーム : Silicon Graphics Indigo2 Extreme, Memory 96MB
- 開発言語 : C Compiler version 7.2
- 開発環境 : IRIX 6.2、X window 6
- 協調制御 : Silicon Graphics Indigo2 Extreme (Main System) から起動・停止制御  
Silicon Graphics OCTANE (3D Viewer) からデータ表示制御  
Silicon Graphics O2 (Data Base System) からデータ表示制御



## 2. 7 S/W QC GUI仕様 (準拠標準 : OSF/Motif, IRIS4D/GL)

本システム開発に必要な、ソフトウェアの品質管理に関わる情報管理について操作するためのGUIセグメントとした。図2. 7-1に設計画面を示す。

### (1) 画面配置

図2. 7-1に、プログラムのコールバック解析図を示す。

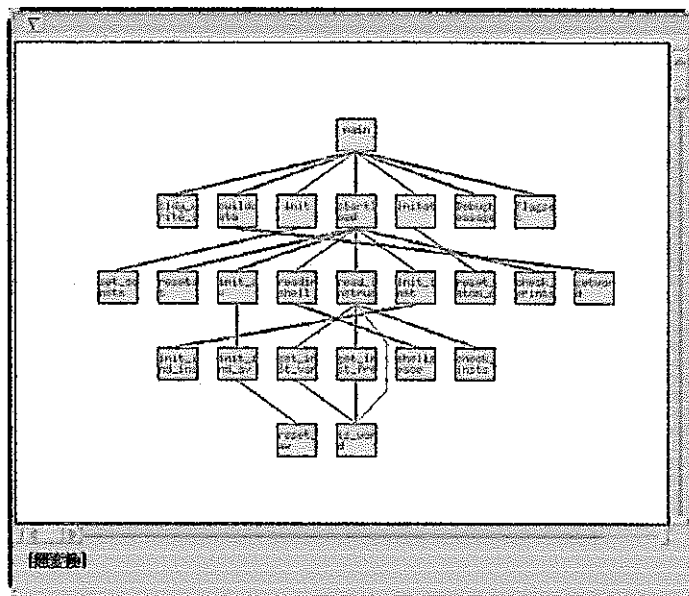


図2. 7-1 プログラムのコールバック解析図

図2. 7-2に、品質解析ツールの操作画面を示す。

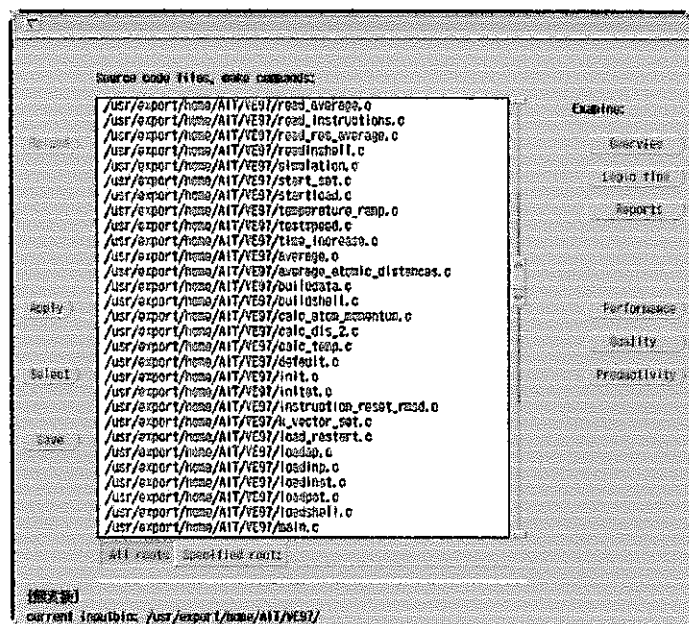


図2. 7-2 プログラムのコールバック解析図



図 2. 7-3 に、複雑度評価画面を示す。

Function	Complexity
average_reset	1
calc_aton_momentum	1
calc_dis_2	1
sqn_dis_2	1
calc_aton_temp	1
init	1
reset_aton_dist	1
set_time_vars	1
set_energy_vars	1
instruction_zero_total_momentum	1
instruction_reset_variables	1
Debug/message	1
inst_set_next_time	1
start_set	1
temperature_ramp	1
kelvin_to_joules	1
main	1
start_outside_end	1
set_consts	2
init_av	2
reset_av	2
inst_set_next	2
calc_temp	2
free_insts	2
init_inst	2

図 2. 7-3 複雑度評価画面

## (2) GUIプログラム仕様

図 2. 7-1 から図 2. 7-3 に示す画面制御プログラムを実装するシステムの開発環境と制御条件は下記のとおりである。

- プラットフォーム : Silicon Graphics Indigo2 Extreme、Memory 96MB
- 開発言語 : C Compiler version 7.2
- 開発環境 : IRIX 6.2、X window 6
- 協調制御 : Silicon Graphics Indigo2 Extreme (Main System) から起動・停止制御  
Silicon Graphics OCTANE (3D Viewer) からデータ表示制御  
Silicon Graphics O2 (Data Base System) からデータ表示制御

## 2. 8 対話機能に関わる認知・知識工学的技術等の応用について

対話機能を認知科学の領域と捉える研究は、GUI設計にとって重要な課題と考えられた。なぜならば、一般にGUIは試行錯誤的な操作を利用者に要請するため、問題解決の手段としての情報システムを考えた場合に、人とコンピュータの対話により進められる一連の認知の過程（知識の蓄積と判断の過程、さらに学習、そういった過程全て）が適切に進められることが重要であると思われるためである。従って、性能評価シミュレーションという活動を考えた場合でも、各種のシナリオを評価し、最終的に生物圏が漏洩した放射性核種から保護されることを確認するための認知過程であると考えられた。そのような観点から考えると、性能評価シミュレーションという学習活動は、

諸々の情報に関して次のように取り組むとよいと思われた。

研究にあたって、認知の過程に於ける情報の種類を視覚・パターンの情報と、いわゆる論理・知識の情報というように分けることにする。パターンの情報の代表なものはグラフィックス（表象）である。従って、これらを絵や画面に表示されたパターン情報と捉えることにした。一方、知識の情報は、文字によって記録され文章といった論理情報として考えることにした。そのため、性能評価は、研究者がこの2つの領域の情報を学習認識手段としながら、目標とする最終的な評価結果を得る知的活動と考えた。そのような活動をヴァーチャル・エンジニアリング・システムの利用者の認知過程のモデルとして捉えることが今後の研究課題と思われた。

### 認知学習活動としての性能評価

対話システムの開発においては、認知のモデルを組み立て、感覚と知覚、言語と思考および注意と記憶といった一つの精神活動をベースにした個人の認知過程を捉えていくことが重要と思われた。

たとえば、ある特定の色、パターンにより、どのように理解度が変化するかが対話機能の使いやすさ、さらに正確さに直接寄与していると思われた。特に一つの認知モデルをメニューに対応させることは興味ある課題と思われた。FEPシナリオを例に取った場合に、個別FEPマトリクス及び全体FEP階層マトリクス、そしてインフルエンス・ダイアグラムという情報の領域があり、それらが個別に表示されるわけであるが、その表示の善し悪し（理解しやすさ）により、FEPを利用したシナリオ開発という活動が左右される可能性があると思われた。なぜならば、シナリオの開発という一つの学習課程は、複数の人間の知的過程を一人の人間が一つのシナリオ置き換えるという、非常に複雑な認識活動を含むものだからである。そこには、論理的な手法より、むしろ臨床的な認知の過程が多分に含まれていると考えられた。また、個人の主観が入り込む領域でもあると考えられる。

従って、本システムの利用者が対話機能を相手にした場合に、どのように操作を進めるかということは興味ある課題であると思われた。それは、ある程度の個人の主観に沿った関心事や、好みに沿って対話機能を操作していくことが一般的であると思われた。そのようにして、システムが好みに沿って、もしくはそのような利用者のアクションに対して期待した通りの結果を表示する場合に利用者の満足が得られる。以上のことから、本システムの利用者が、それを操作する上で期待した情報が表示されるかという、主観的な側面を含む問題を取り扱っていかなければならないことが対話システムの設計において設計作業自身を複雑にする一つの要因であると考えられた。そのような臨床的な認識過程に関わる個人の主観的な活動を設計に考慮しないで、純粋に工学的な見地のみに沿って対話機能を設計することが可能であるとも考えられた。そのような対話機能は、たとえば、CADといった世界で既に実現されていると考えられた。なぜならば、CADシステム等においては、それが一つの目標とする設計をとり行い、理論的にその論拠を固めることを目標にするためである。

### 主観を内在する個別シナリオ（原データ）

一方、性能評価シミュレーションにおいては、対話システム設計の取り扱う情報の領域がシナリオ評価を含むために、臨床的（個人的主観）な側面を含んでいると思われる。性能評価においては、評価の枠組みは与えられるが、評価のシナリオは専門家（個人）の考えとして提出されている。その場合、シナリオに沿ってシミュレーションを進めるという考え方を推進する上で大き

な問題となるのは、シナリオの組み合わせが爆発し、最終的には発散してしまう可能性であると考えられる。性能評価においては、シミュレーションの目的は極めて単純である。それは漏出した核種が与えられた時間の間、多重バリアにより確保され、固定され、最終的に生物圏に至った時の線量が、天然自然の線量以下にあることを示すことが基本と思われる。たとえ漏出した放射性核種が生活圏に到達しても、その放射線量はクリアランスレベル以下にあることを示すことである。そのように、問題自身は非常に単純であるが、深地層中で漏出した核種が生活圏に到達するまでの現象は非常に複雑になると考えられる。そのため、核種移行のシナリオが必要となる。そのような背景から、評価においては、第一に評価の枠組みの整理が推進されている。整理された評価の枠組みに沿って考えられるシナリオが開発されると思われる。

地層処分ヴァーチャル・エンジニアリング・システムにおいては、システム設計作業においてシナリオが重要な問題になると考えられた。シナリオは、かなり創造的な、あくまで個人の発想に基づくものであるため、総合的な環境影響評価としての性能評価の枠組みを考えた場合に、広範な要因（核種の移行に関わる影響の要因モデル、それとともに個人の主観を含むシナリオ）を取り扱わなければならないと思われるためである。そのため、対話システムの設計においては、前節にてGUIメニューとして示した天然地形や人工バリアの各ステージなどのように、計画する処分システムの工学的な評価のための一つの枠組みとなる情報の区分を扱うこととした。従って、核種移行にかかわる要因をどのように枠組みづけるかは、一つの物理化学的な活動の範囲で、ある程度の形は作り出せるものと思われた。ただし、経済的な問題、さらに社会心理的な問題は別であると思われる。地層処分ヴァーチャル・エンジニアリング・システムの現在の設計は、前述したように核種移行という考え方から発祥して、最終的に多重バリアシステムが正しく機能するかという純工学的な関心事に沿って評価することを目的としている。

そのように、純粋な工学的観点に沿って核種移行を評価する場合においても、評価の枠組みはある程度確定することはできるが、評価のシナリオではどのように確定するかが非常に重要な課題となってくると思われた。なぜならば、シナリオは個人から出されるため、完全に客観的なものにはならないためである。現在、評価シナリオ開発は、基本的にFEP手法と呼ばれる評価シナリオの原案となる意見の収集により開始するという方法がとられている。しかし、FEP手法は専門領域を記述したものである。そのような個別の専門領域から性能評価の枠組み全体を見渡すことは困難であると思われる。また、前述のとおり、個別の専門領域には専門家の主観が（心理面から）内在していると考えられる。そのように、個別シナリオは全体の記述に欠けることと、主観という不確実性を内在していることを前提として参考にすべきと思われる。そのため、地層処分ヴァーチャル・エンジニアリング・システムでは、収集された個別シナリオを原データ（情報）として評価し、より客観的なシナリオをシステムにより組み上げる手法の研究が必要であると考えられた。また、それは対話機能の一部として組み込むことが適当であると考えられた。

#### FEPシナリオの扱い

FEPシナリオは、ある特定した現象を記述し、その現象に影響する要因となる他の現象及びその現象が寄与する他の現象という一つの影響関連を含めて、文章型で記述した情報に基づくものである。一般に、それはFEPテキストと呼ばれる。たとえば、地下水の地球化学的な特性といったものである。そして、その地球化学的な変化に寄与する要因、並びに地球化学的な特性の変化により変化する要因の関連を示した文章が提出される。このような文章は、特定の個人、いわゆる専門家といわれる個人により、創造的に発想記述される部分を含んでいると考えられる。

従って、シナリオ開発において、FEP情報は個人的な主観を含むものであることを前提としな

ければいけない。そして、それらの提出されたシナリオの中から、どのようなシナリオが最も重要であるかを検討し、最終的に最も支配的と考えられるシナリオを抽出する作業がシナリオ開発であると考えられる。そのため、シナリオ開発の作業は、他の工学分野の研究開発と多少性格が異なっており、一部臨床的な情報の整理が要求されると考えられる。また、性能評価シミュレーションにおいては、シナリオが非常に重要な位置づけである。対話システムの研究では、このシナリオ自身が、どのように対話システムの中に取り込まれているかを重視しなければいけない。なぜならば、前述したようにシナリオにはかなりのレベルでの個人的主観が不確実性として含まれているからである（ただし、それは厳密には不確実性ではなく、心理作用と考えられる）。それは、純粹に工学的な評価の枠組みである理論の世界にそのまま取り込むことは非常に難しいと考えられる。従って、性能評価においては、その間をどのように埋めていくかという非常に重要な課題が生じてくる。そのような、いわゆる臨床的な情報のシナリオから理論的な純工学的な情報の評価モデルの間を橋渡しする活動を考えた場合、それはまた新たな研究課題であると考えられた。

対話システムは、そのような境界領域を上手に橋渡しし、個人的主観を含む、いわば個人の認識過程や精神過程という活動から、より一般的（客観的）な物理工学の理論という、アセスメントの枠組みに結びつけるための一つの手段を提供することが機能に望まれることであると考えられた。特に、それは対話システム研究の中でも重要な認知科学レベルでの研究であると考えられた。

ヴァーチャル・エンジニアリング・システムに於ける研究に期待するところは、単純にデータを管理するだけのソフトウェアの開発ではなく、性能評価に於ける最も複雑で議論を呼ぶ話題である、この評価シナリオの組立と評価の枠組みの制定という作業にどのように役に立つのかということであると思われた。恐らく、対話システムは視覚的な活動に訴えるために、一つの視覚認識レベルでの手段は効率的であると思われた。しかし、シナリオのような、いわゆる知識の蓄積は文章によって行われるため、そのような文章情報のデータベースを上手くリンクして、それを補う形でのパターンによる抽出という機能が重要視されてくると思われた。一般に、認知活動においては、理解はパターンにより行うことが容易であり、知識の獲得は言語により実践することが基本であると考えられる。従って、今後の対話システム研究においては、与えられたシナリオの中から、必要となるシミュレーションを組み立てるためのシナリオのパターンを抽出する作業を支援する機能が将来必要と思われた。それにより、専門家個人により発行された個別シナリオを総合的にコンピュータが整理し、機械による、より客観的な評価シナリオの開発がなされることが期待できる。

以上のように、平成9年度は対話システムの研究という課題について、その研究の領域を単純な対話機能プログラムの開発といった活動だけではなく、対話システムそのものが、シナリオ開発や評価の区分を整理するためにどのように応用できるか、特に認知過程をどのように支援していくかといった、認知科学的な分野の研究を予備的に執り行った。

### 3 視覚化複合事象シミュレーション機能統合型シミュレーションの基本設計

地層処分ヴァーチャル・エンジニアリング・システム全体の動作仕様を設計した。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、処分の性能評価シミュレーションに係わる評価シナリオと、それら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、複数から構成するコンピュータシステムとした。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)

システム・ハードウェアの構成は、UNIX系とPC系に大別し、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・ウインドウ制御を統一するとともに、Zバッファを画像フレーム・メモリとして有するシリコン・グラフィクス社が開発したIRIS4D/GLプロトコルを採用した。また、X11のウインドウ管理については、OSF/Motifxm intrinsicを採用した。これは、システム構成上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。PCは、仕様の公開されているIBM/PC系を採用し、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。

#### 3. 1 視覚化複合事象シミュレーション機能 (エンジン) 基本設計

前年度の検討から、ネットワークを通じ互いのハードウェアで実行中のコードが情報を交換しながら計算処理・動画像処理を行えるための技術基盤の確認ができた。本年度は、同技術を基盤として、視覚化複合事象シミュレーション機能を提供するための汎用の実時間視覚化処理モジュール (シミュレーション・エンジン) の設計を行った。

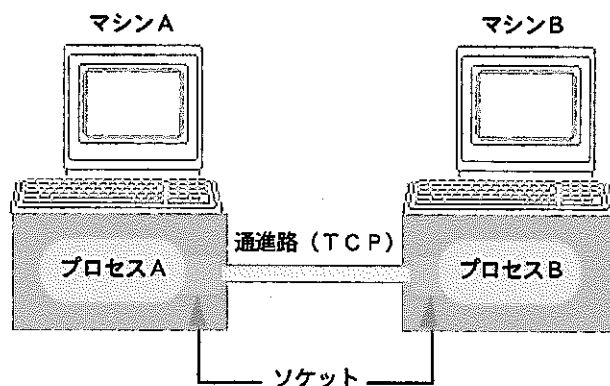


図3. 1-1 TCPソケットインターフェイス概念

このシミュレーション・エンジンには、ネットワークに接続された高速な3次元画像処理フレーム・バッファを搭載しているマルチ・プロセッサ・システムをハードウェアとして利用することを念頭に設計を行った。また、同エンジンと個別の解析コードの出力とのインターフェイスには、TCPソケットインターフェイスを利用した。

(1) 基本内部処理仕様概要

対象とするシミュレーション・エンジンは、個別のシミュレーション・プログラムを除いたシミュレーションに必要なソフトウェアである。それらは、例えばGUIよりコールバックされ、データベースや個別シミュレーション・プログラム等をコールバックする処理を行う。従って、シミュレーション・エンジンは個別のソフトウェアを関連づけ、一連のシミュレーションを行う操作を制御する中核となるソフトウェアとして開発する。

操作概念1：ステージ

シミュレーションの基本となる考え方を図3. 1-2に示す。これは、ステージという概念を試験的に導入している。ステージの概念は、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である。

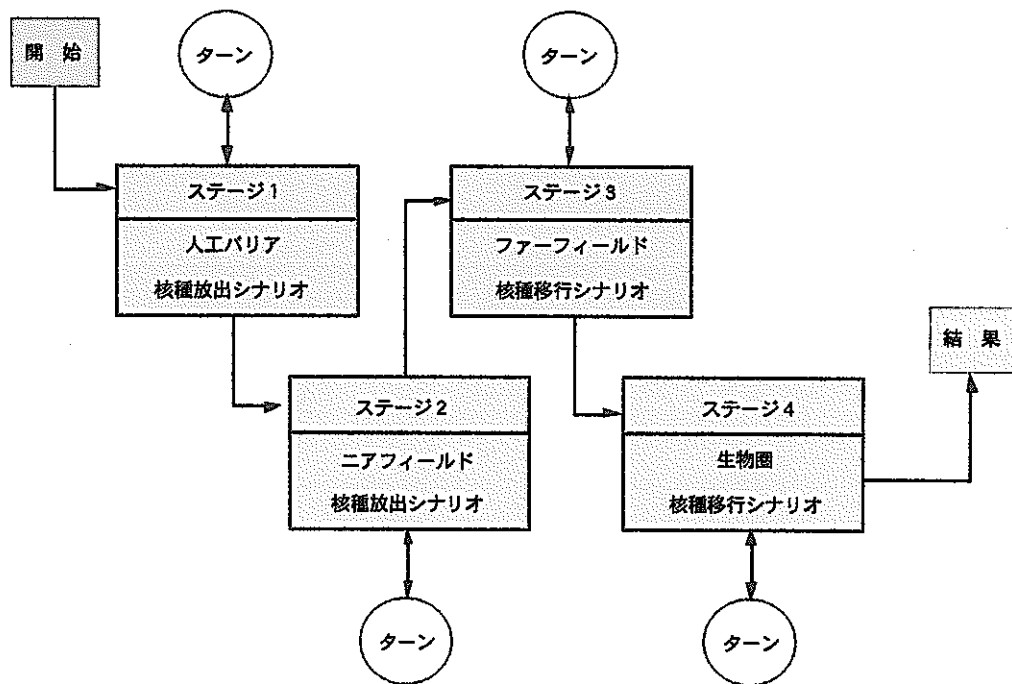


図3. 1-2 ステージの概念

図3. 1-2に試作システムの4段階ステージを示した。図に示すように、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。これによりシミュレーションは、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うものとした。このような考え方は、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問題と思われるが、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ている。シミュレーションは、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い、結果を以下に示す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、ステージごと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。

### 操作概念 2 : ターン

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とした。また、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。

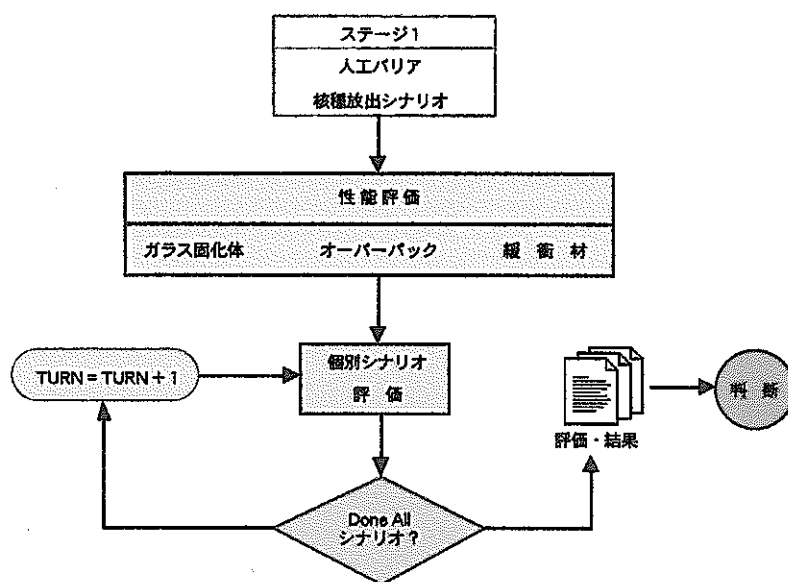


図 3. 1-3 ターンの概念

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、オーバーバックおよび緩衝材についての個別シナリオが、階層 FEP 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、緩衝材中の化学的現象・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、緩衝材中の化学的現象・特性というサブ・グループについてはターンの値は 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せら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

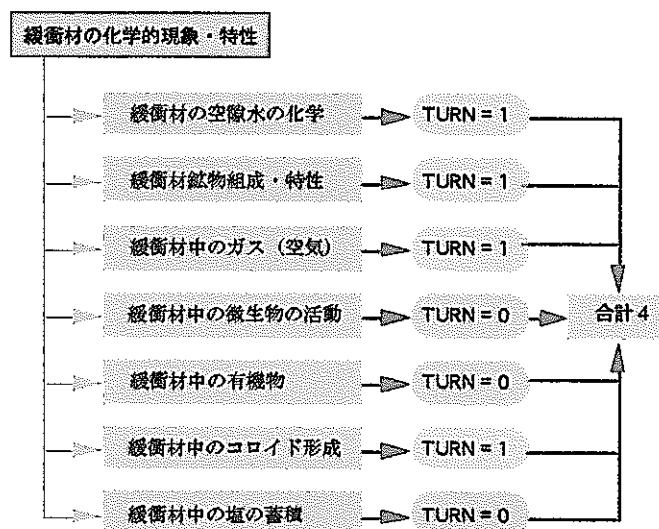


図3. 1-4 サブ・グループ・ターンの合計概念

個別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思われるが、本研究は、情報処理技術の応用としてシステムティックな性能評価を目標とした情報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、前記の緩衝材中の化学的現象・特性にかかわる評価は、例えば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3. 1-5 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。

B-4 緩衝材の化学的現象・特性	
B-4.1 緩衝材中の空隙水の化学	0
B-4.2 緩衝材の鉱物組成・特性	0
B-4.3 緩衝材中のガス（空気）	2
B-4.4 緩衝材中の微生物の活動	3
B-4.5 緩衝材中の有機物	0
B-4.6 緩衝材からのコロイドの形成	0
B-4.7 塩の蓄積	0

閉じる

図3. 1-5 サブ・グループ・ターン完了の表示



### シナリオ参照操作仕様

各ターンを実行する操作をシナリオ評価と考えた。シナリオ評価は、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3. 1-6に示す通りに、シナリオ評価においては、FEPリストの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3. 1-6に示す一連の評価作業の完了をもって1ターンとした。

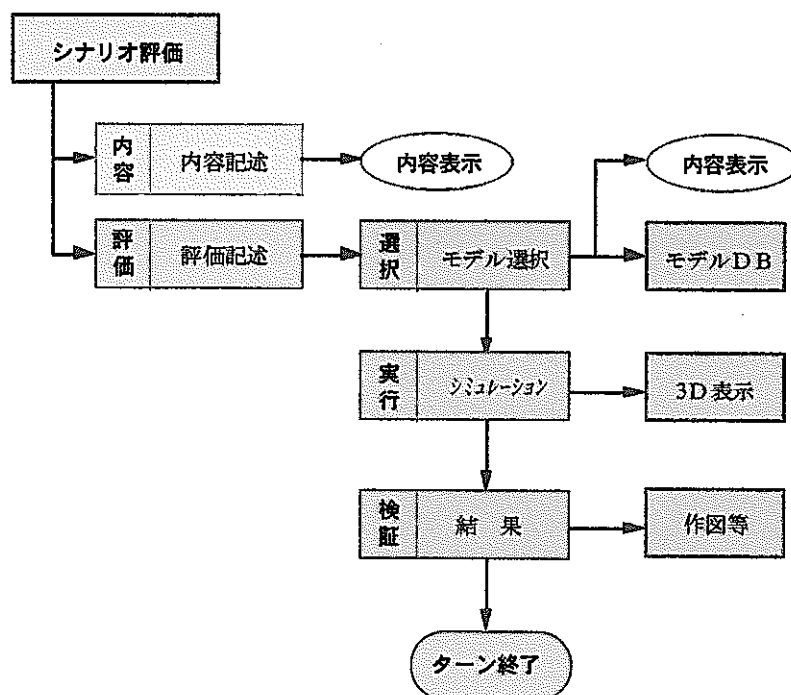


図3. 1-6 シナリオ評価機能概念

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立てる必要があると考えられる。図3. 1-6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。

### シミュレーション操作仕様

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を起動し、計算結果をシステムに共通な3次元動画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、今後詳細な検討が必要である。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、および

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3. 1-7に起動時の操作画面を示す。

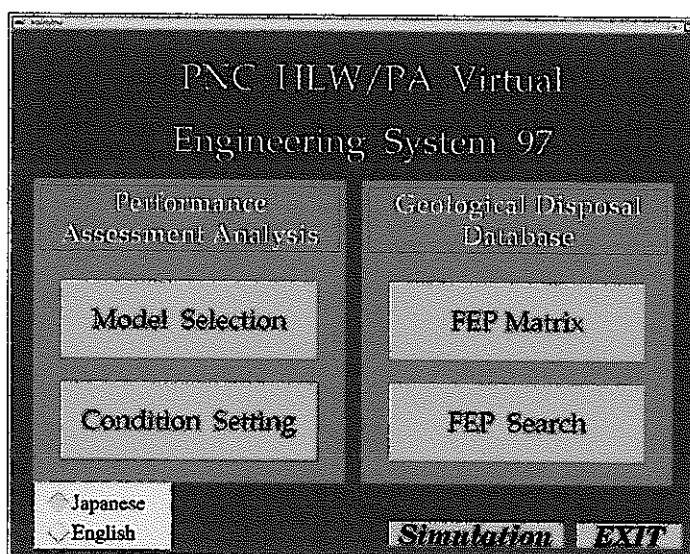


図3. 1-7 初期画面の例

図3. 1-7に示す初期画面は、システムの起動により全ての画面を初期設定し、主操作画面としてFEPデータベースによる評価または性能評価解析による選択のメニューボタンを表示する。また、後の操作に用いる表示画面の日本語または英語を選択できるものとした。これはヴァーチャル・エンジニアリング・システムのアプローチに際し日本語外の対応も配慮したためである。FEPデータベースでは階層マトリクスまたは処分キーワードからの選択、性能評価解析選択ではモデル選択、処分システム条件設定のステージ等の条件を設定できる画面を表示するものとした。図3. 1-8に処分システム条件設定画面を示す。

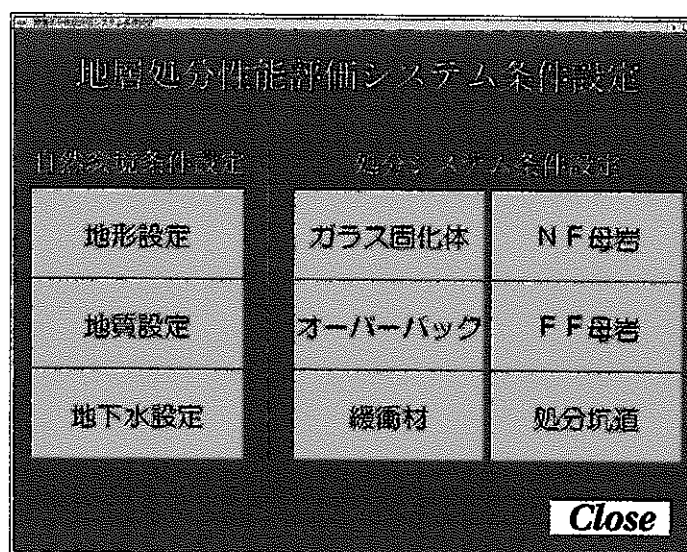


図3. 1-8 処分システム条件設定画面の例

図3. 1-8の処分システム条件設定画面は自然環境条件と処分システム条件に分け、処分システム条件ではステージを、自然環境条件では地形設定や地質設定等の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うために予備的に準備した。図3. 1-9にその例を示す。

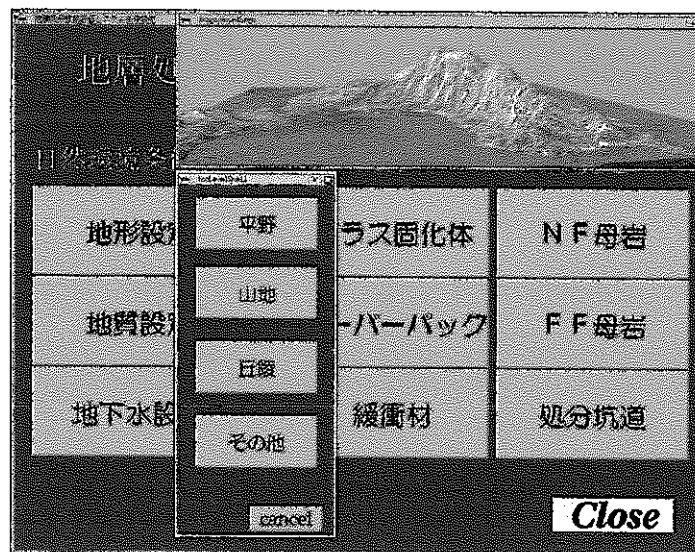


図3. 1-9 天然地形メニューとオブジェクト表示の例

図3. 1-10は処分システム条件設定を選択操作した画面の例である。図では、人工バリア・ステージのオーバーパックを選択したものである。図に示すように、処分システム条件を選択すると当該のオブジェクトを表示しFEPデータベースより関連シナリオリストを収集する。その後、評価するモデルの選択（例えば核種移行など）を行いシミュレーションを実行する。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3. 1-10 処分システム（ステージ）条件設定とオブジェクト表示の例

図3. 1-11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えるようにした。

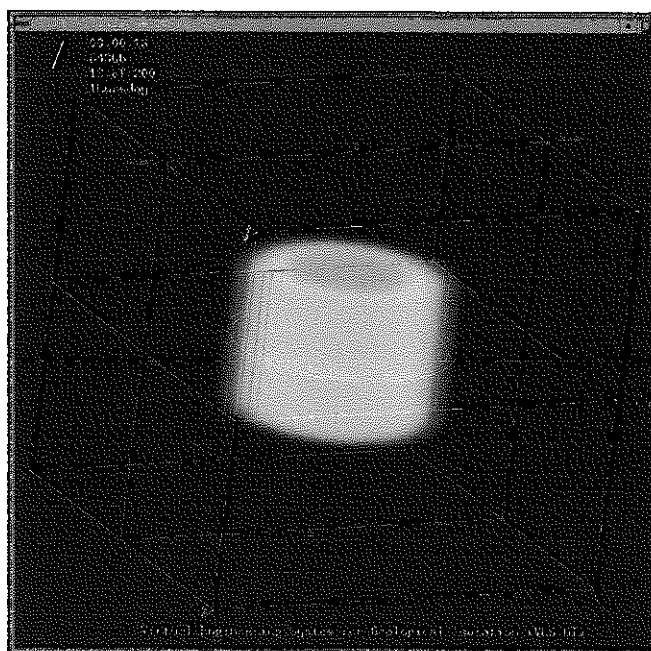


図3. 1-11 3次元シミュレーション動画表現例

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とした。

## (2) シミュレーション制御仕様

シミュレーションの制御を行うソフトウェアの機能は大別して、処分システムとモデルおよびシナリオの設定、シミュレーションの実行、結果の整理（可視化）とした。

### 基本操作仕様

図3. 1-12に主操作画面を示す。これは、システムを初期設定した時に表示される画面である。

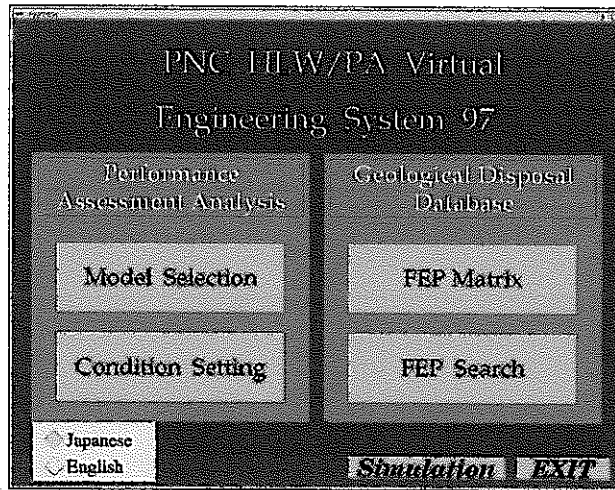


図3. 1-12 主操作画面

図3. 1-12に示すとおり、主操作画面は簡単なメニュー（ボタン）から構成した。メニューは下記の分類に沿ってボタン・ウィジェッドを実装した。

**性能評価解析GUI表示ボタン**

**Model Selection** : シナリオ評価モデル選択GUIを表示させるボタンウィジェッド。

**Condition Setting** : 処分システム条件設定GUIを表示させるボタンウィジェッド。

図3. 1-13にシナリオ評価モデル選択GUI、処分システム条件設定GUI表示例を示す。

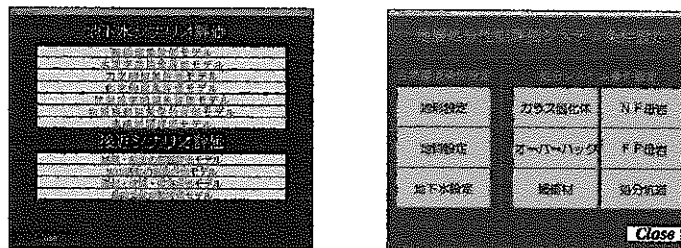


図3. 1-13 シナリオ評価モデル選択GUIと処分システム条件設定GUI

**シナリオ情報選択GUI表示ボタン**

**FEP Matrix** : FEP階層マトリクス・GUIを表示させるボタンウィジェッド。

**FEP Search** : 処分キーワード設定GUIを表示させるボタンウィジェッド。

図3. 1-14にFEP階層マトリクス・GUI、処分キーワード設定GUI表示例を示す。

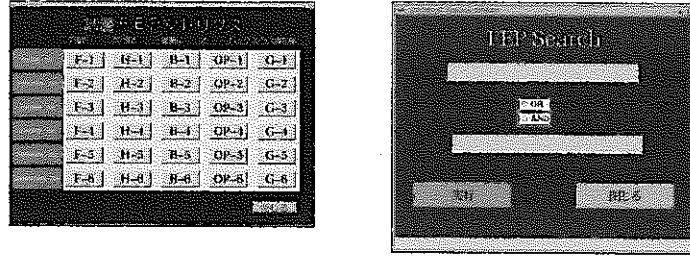


図3. 1-14 FEP階層マトリクス・GUIと処分キーワード設定GUI

**表示言語設定ボタン**

Japanese : この後のGUI表示を日本語とするボタンウィジェット。

English : この後のGUI表示を英語とするボタンウィジェット。

**Simulation、EXITボタン**

Simulation : 設定した各条件でシミュレーションを実行するボタンウィジェット。

EXIT : 処理を終了させるボタンウィジェット。

**処分システム操作仕様**

処分システム操作は図3. 1-15に示す画面で行い、画面は自然環境（シーン）条件設定と人工バリア（ステージ）設定のメニューで構成した。

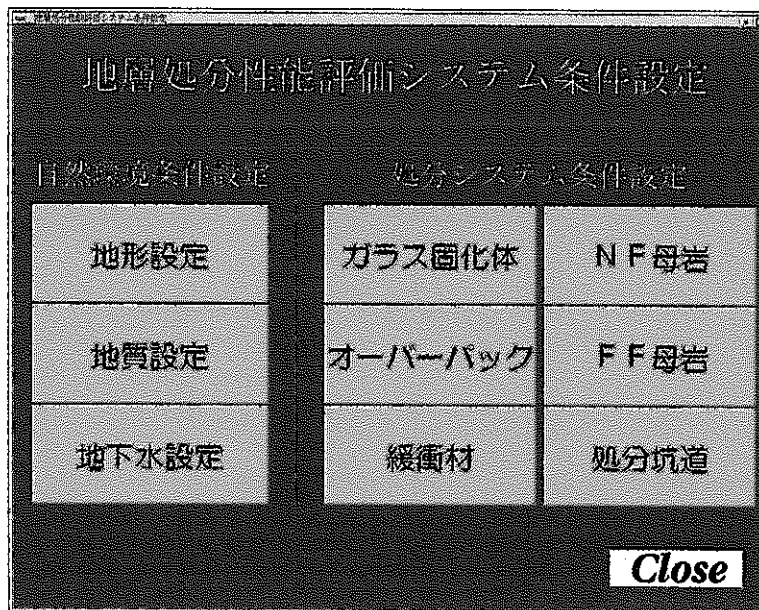


図3. 1-15 処分システム条件設定GUI

### 3. 2 関連研究課題1：評価シナリオ・クラス概念設計

シミュレーション・エンジンを設計するにあたり、対象となる評価シナリオをクラス階層ごとに設計を行った。

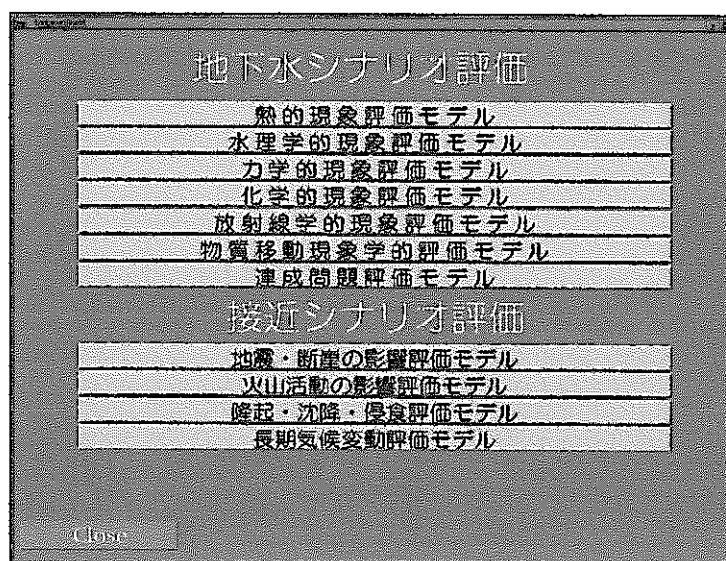


図3. 2-1 シナリオとモデルの階層化管理画面例

また、評価シナリオを構成する事象やプロセスをクラス階層化することにより、個別シナリオとモデルとの連結を可能とするよう設計した。本システムで扱う評価シナリオは、前述のとおり別途開発している「地層処分FEP情報データベース・システム」からの出力を扱う（インポート機能）ものとした。ただし、シナリオの変更が生じた場合には、その関連情報を「地層処分FEP情報データベース・システム」に提供し（エクスポート機能）、同システムによるシナリオ解析の論拠に資する機能についても予備的に検討を行い設計の参考とした。

### 3. 3 関連研究課題2：評価モデルのクラス接続技術研究

クラス別に設計された評価シナリオについて、そのクラス階層（サブクラス）の評価記述を具体化するメンバーとしてのモデルとを接続するための検討を行った。検討にあたっては、メンバーとしてのモデル、考えられるインスタンス変数（例：Kd, pH/Eh等）の扱いについて、プライベート、パブリックの属性概念の規定などについて検討を行い、評価シナリオのクラス設計の参考とした。またクラスごとのシナリオに対応するモデルとして、従来開発されてきた評価コードの適用性の検討を行った。

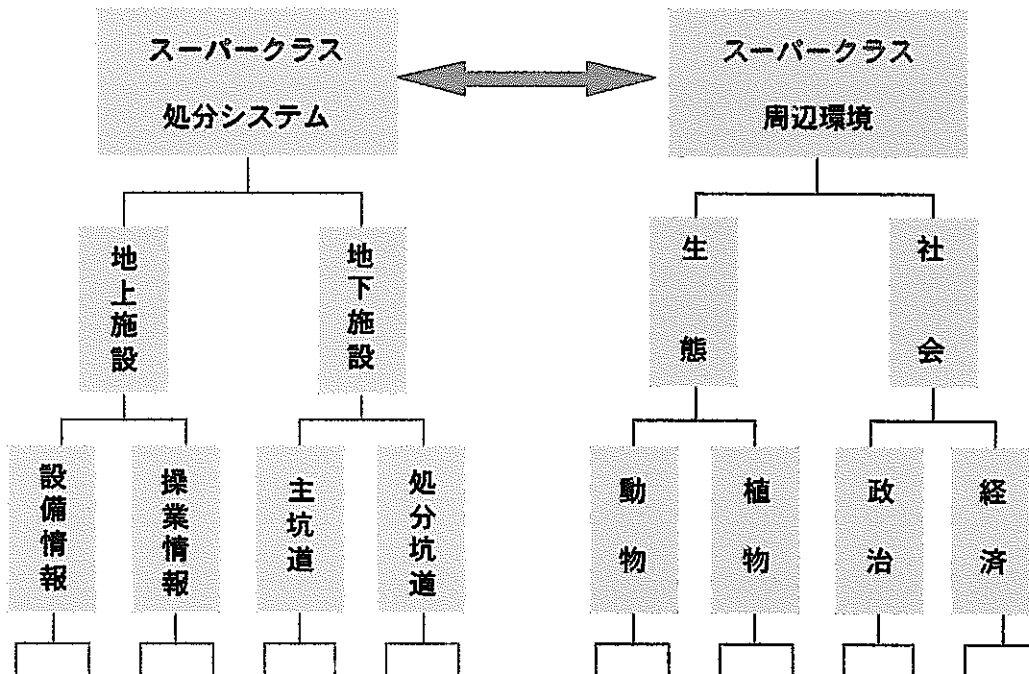


図 3. 3-1 クラス概念



### 3. 4 関連研究課題3：モデル・共通3D動画化プロセス間並列データ通信機能研究

シミュレーション・エンジンを設計するにあたり、必要なモデル・共通3D動画化プロセス間並列データ通信機能の試作評価を行った。この機能は、前述したとおり、TCPソケットインターフェイスを通じて、FORTRAN プロセスから新しい計算サイクルの変量が得られる毎にシミュレーション・エンジンにそれを送り込む機能（並列ネットワーク・メッセージ通信）である。本年度は、同時に2つの計算プロセス（解析コード）が並列に実行しながらメッセージをエンジンに送信してくる条件を想定したプロセス間並列データ通信機能の検討を行った。

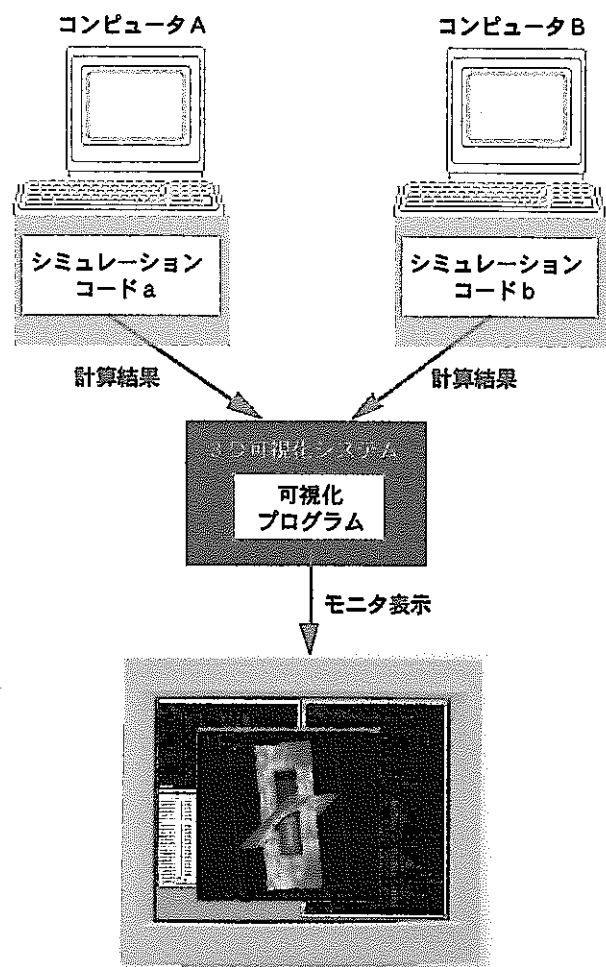


図3. 4-1 プロセス間通信による可視化シミュレーション概念

### 3. 5 関連研究課題4：人工バリア核種放出シナリオ・シミュレーション試作

シミュレーション・エンジンを設計する上で必要な技術について、開発成果を具体的に評価することを目的に、人工バリアからの核種放出シナリオを対象としたシミュレーション・モデルRELEASEを試験的にシステムと接続した。採用にあたっては、ニアフィールド核種移行評価コードRELEASEはガラス固化体から溶出した核種が緩衝材を経て周辺岩盤に移行する挙動を解析するため事業団によって開発された解析コードであり、これまで多くの解析に使用された実績があることと、RELEASEの計算出力が他解析コードでも採用されている円筒座標系1次元モデルであるので可視化技術の検討を行う必要があると判断したためである。試計算に使用した条件を表3. 5-1に示す。

表3. 5-1 計算条件

ガラス固化体の半径 (m)	0.52
ガラス固化体の高さ (m)	1.95
緩衝材の厚さ (m)	0.98
緩衝材の内側表面積 (m <sup>2</sup> )	4.9
緩衝材の真密度 (kg/m <sup>3</sup> )	2.7E+03
ガラス固化体の重量 (kg)	412
ガラス固化体の密度 (kg/m <sup>3</sup> )	2.7E+03
緩衝材の空隙率	0.333
緩衝材中での実効拡散係数 (m/yr)	9.461E-03

	半減期 (yr)	ガラス固化体中 インベントリ (g)	溶解度 (g/m <sup>3</sup> )	分配係数 (m <sup>3</sup> /kg)
Am-243	7.38E+03	9.568E+02	1.1E-02	10
Pu-239	2.41E+04	4.623E+01	8.9E-06	1
U-235	7.04E+08	1.937E+01	1.0E-04	0.1

以上の計算条件で、Am-243、Pu-239、U-235の3核種について計算した。つぎにRELEASEは、1次元円筒座標系モデルのためこれを3次元表示するために以下に示すデータ変換を行った。

- (1) データの補完 : RELEASEコードの出力フォーマットの特性のため
- (2) 座標系の変換 : 1次元円筒座標系データを3次元直交座標系へ変換
- (3) 常用対数への変換 : オプション、出力データの容易な把握のため

### データの補完

RELEASEでは、計算セルは以下に示すように境界に接するセル幅は他のセルの1/2になっている。

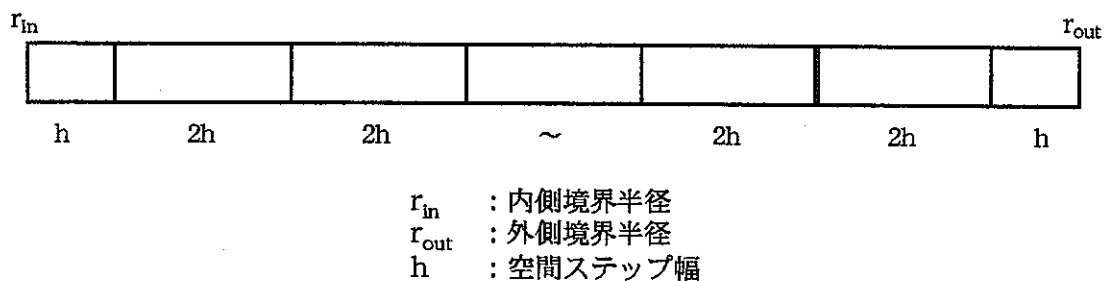
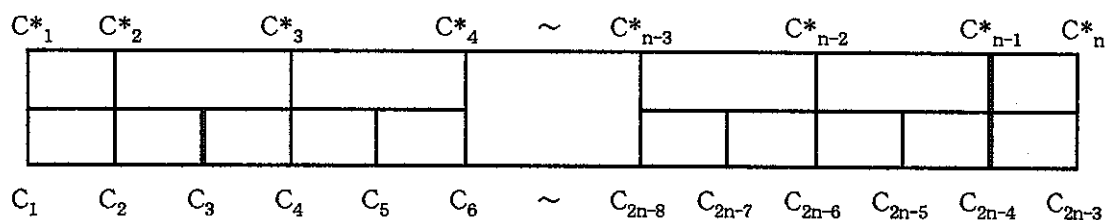


図3. 5-1 計算セル

よって空間ステップ幅をhに統一するためにデータの補完を行った。補完の方法は線形補完を用いた。以下に元の濃度データ群と補完を含めた濃度データ群の関係図及び関係式を示す。



$C^*_a (a=1,n)$  : 元の濃度データ群  
 $C_b (B=1,2n-3)$  : 補完を含めた濃度データ群

$k=1$ の場合  $C_1 = C^*_1$   
 $k=2, n-2$ の場合  $C_{2k-2} = C^*_k$   
 $C_{2k-1} = (C^*_k + C^*_{k+1}) / 2$   
 $k=n-1$ の場合  $C_{2k-2} = C^*_k$   
 $k=n$ の場合  $C_{2k-3} = C^*_k$

図3. 5-2 補間の考え方

### 座標系の変換

1次元円筒座標系による計算では、Z方向及び円周方向に濃度勾配はない。よって、変換は3次元直交座標系の原点Oと点P(x,y)までの距離Rを求め、このRと一致するまたは最も近い1次元円筒座標系の距離rにおける濃度データを点Pに与えて変換した。

以下に今回用いた変換手法を示す。

- (1)  $R^2 = x^2 + y^2$ より3次元直交座標系における中心からの距離を求めた。
- (2) (1)で求めたRに一致する、または最も近い1次元円筒座標系の距離rを求めるためにRを以下の式に代入し評価した。

$$r = \text{idnint}((R-rin)/h)+1$$

- ir : 1次元円筒座標系の出力節点番号に相当
- idnint : FORTRAN77で整数に四捨五入する関数
- rin : 内側境界半径
- h : 前述の空間ステップ幅

(3) 上記irと一致する出力節点番号の濃度データを抽出する。

#### 常用対数への変換 (オプション)

まず、RELEASEコードの出力ファイル中の溶解度及び本プログラム用入力ファイル中の表示オーダーを以下の式に代入して表示最下限値を決定した。

$$\text{表示最下限値} = \text{LOG10}(\text{溶解度}) - \text{表示オーダー}$$

次に、出力濃度を以下のように変換した。

- 出力濃度≠0の場合 : 変換濃度 = LOG10(出力濃度)
- 変換濃度<表示最下限値になった場合 : 変換濃度 = 表示最下限値
- 出力濃度=0の場合 : 変換濃度 = 表示最下限値

以上により、常用対数への変換を行った場合、表示最下限値を下回る濃度は全て表示最下限値に変換される点に注意が必要である。また、3Dで表示される数値は濃度のオーダー (次数) になる。

#### 可視化シミュレーション結果

図3. 5-3に計算に用いたモデルを示す。

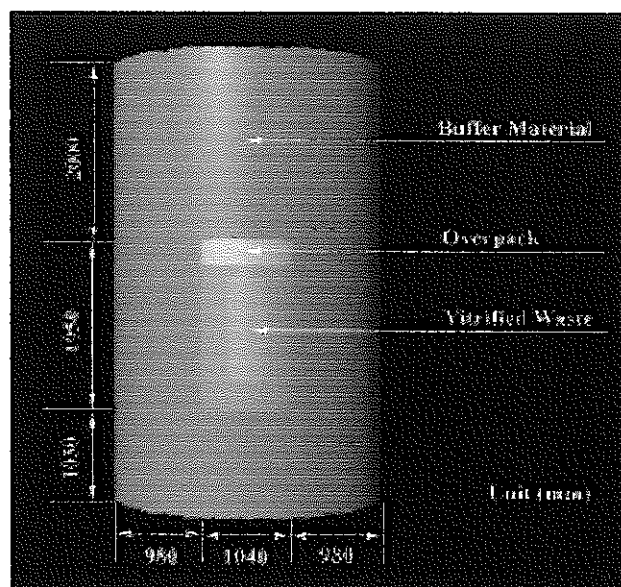


図3. 5-3 計算モデル

図3. 5-4から図3. 5-7に試作したシステムの表示画面を示す。

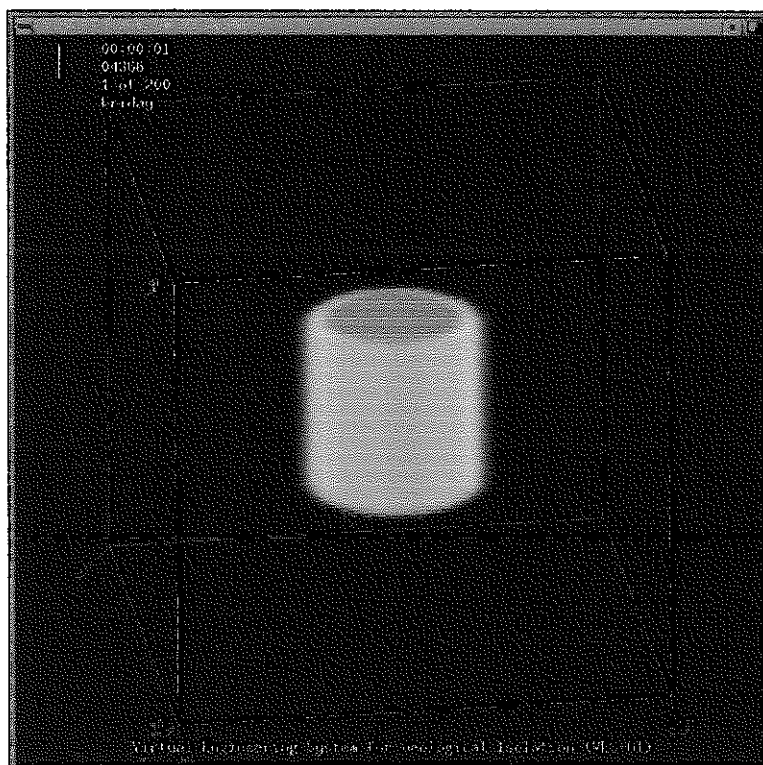


図3. 5-4 Visual.RELEASE開始画面

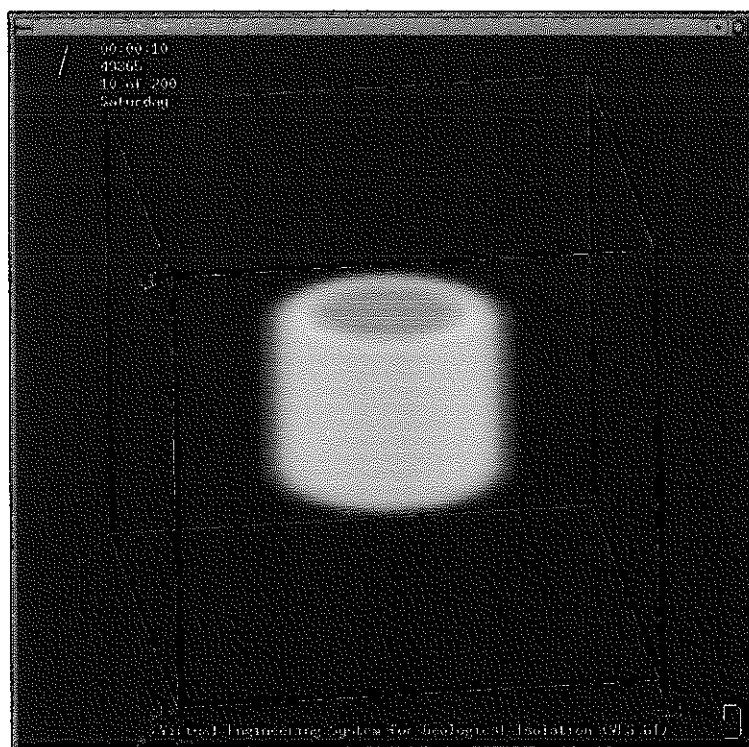


図3. 5-5 Visual.RELEASE動作画面

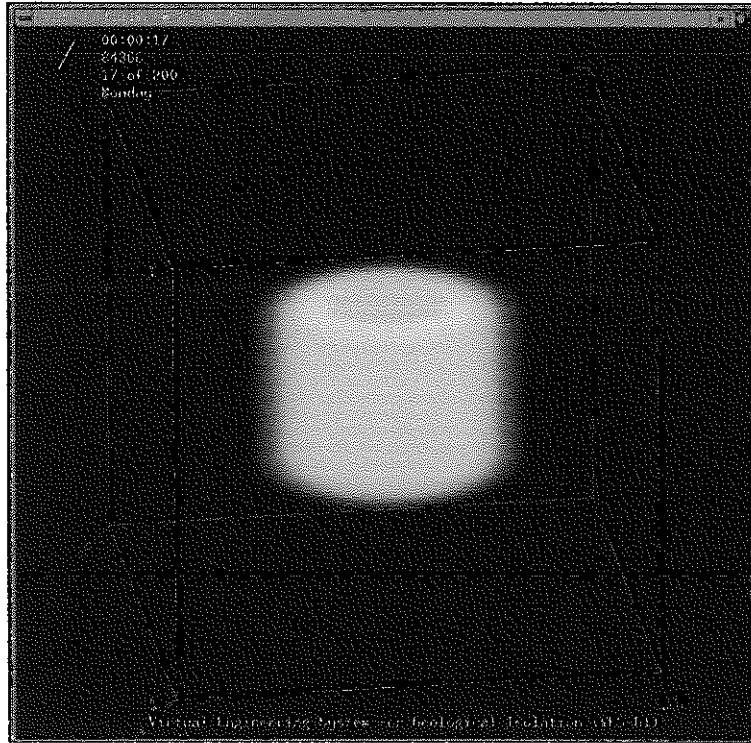


図 3. 5 - 6 Visual.RELEASE動作画面

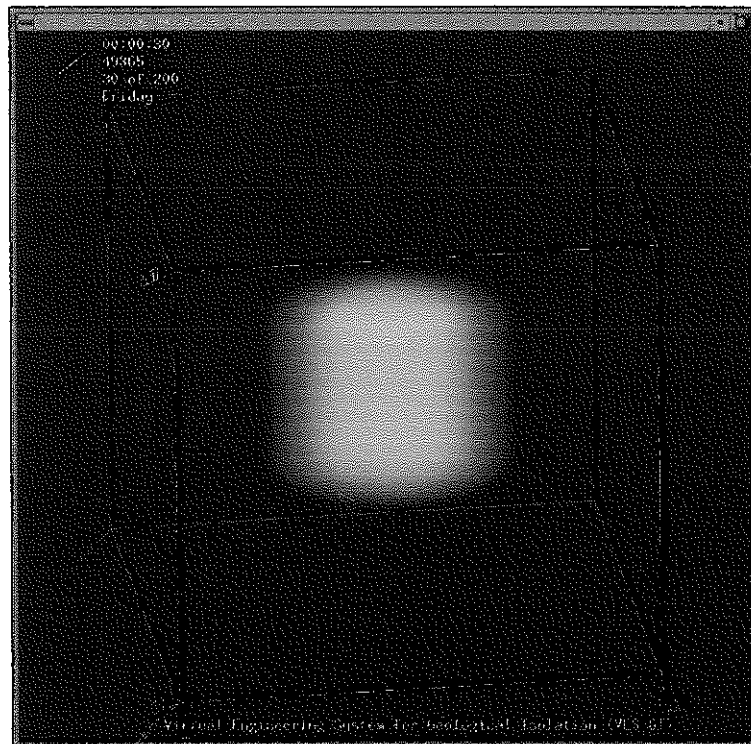


図 3. 5 - 7 Visual.RELEASE動作画面

### 3. 6 関連研究課題5：ニアフィールド核種放出シナリオ・シミュレーション試作

ニアフィールド核種放出シナリオを評価する熱・地下水シミュレーション・モデル（TOUGH2：EOS3（大気/水）モデル）を試験的にシステムと接続した。TOUGH2は多次元（1-, 2-, 3-）2層流体・熱連成問題における世界的に代表的解析コードであり、多くの廃棄物性能評価分野に置いて使用されている実績があることと、複数種類の計算結果出力値を同時に可視化する技術の検討を行う必要があると判断したため採用した。

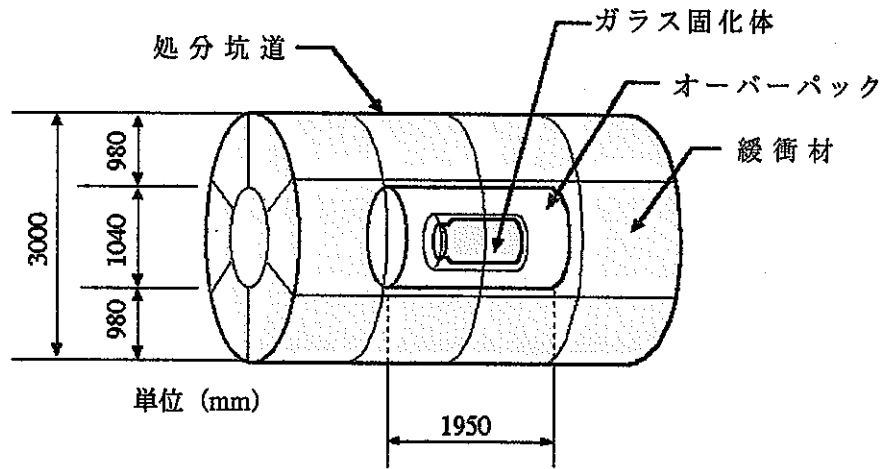
本年度の研究範囲として、処分孔道方式（図3. 6-1 a、横置き）及び処分ピット（図3. 6-1 b、縦置き）の2つの放射性廃棄物処分概念についてシミュレーションすることとした。シミュレーションにおいては、廃棄物パッケージ（ガラス固化された廃棄物をオーバーパックしたもの）、緩衝材（ベントナイト）及び周囲の結晶質岩である母岩などをモデルとして考慮した。また、処分ピットオプションには、アクセス坑道の部分も含んだ場合を考慮した。

出力の視覚化は、単一廃棄物パッケージの場合（図3. 6-1）と4つの廃棄物パッケージの場合（図3. 6-2）とした。対称境界を設けたTOUGH2の計算条件は、それぞれの処分オプションに関して行った。TOUGH2コードによる計算の結果を組み合わせて視覚化を行った。シミュレーション及び視覚化の条件、表3. 6-1に示す。

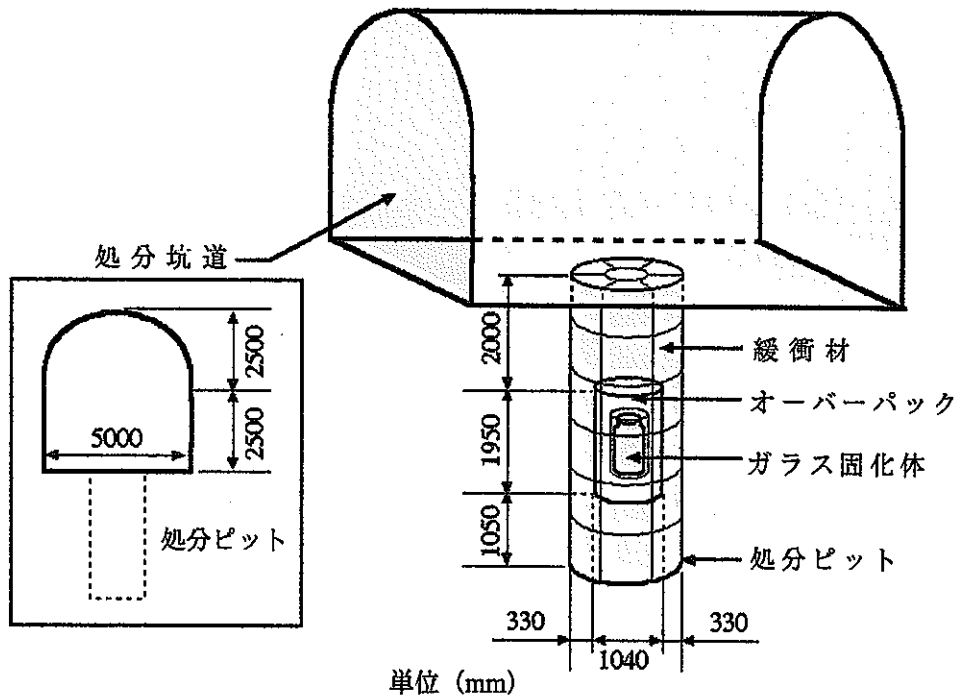
表3. 6-1 TOUGH2シミュレーションおよびVis5D視覚化のリスト

廃棄物オプション	感度オプション	廃棄物パッケージ数	TOUGH2シミュレーション	Vis5Dによる出力
坑道	ベースケース	1	Yes	直接出力
		4	No	1パッケージからの組合せ
坑道	代替的地質状況	1	(Yes)	なし
坑道	不連続な割れ目	1	(Yes)	なし
坑道	発熱量の減少	1	Yes	直接出力
ピット	ベースケース	1/4	Yes	なし
		1	No	1/4パッケージからの組合せ
		4	No	1/4パッケージからの組合せ
ピット	代替的发熱率	1/4	Yes	なし
		1	No	1/4パッケージからの組合せ

(Yes)：予備的なパラメータセットを用いてシミュレーションを行った。シミュレーション結果では、ベースケースの結果とは大きな差異はなかった。



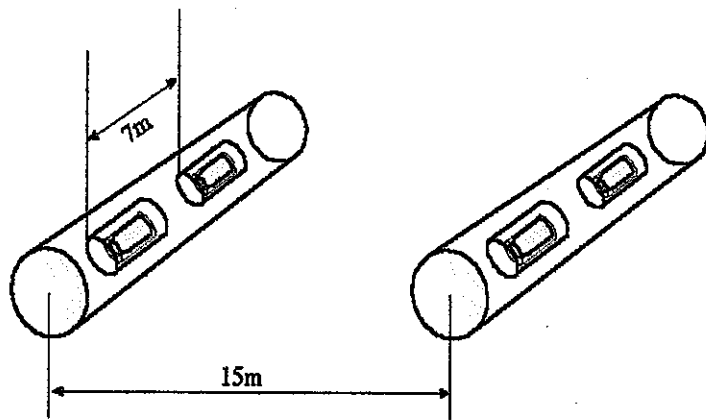
(a) 処分坑道



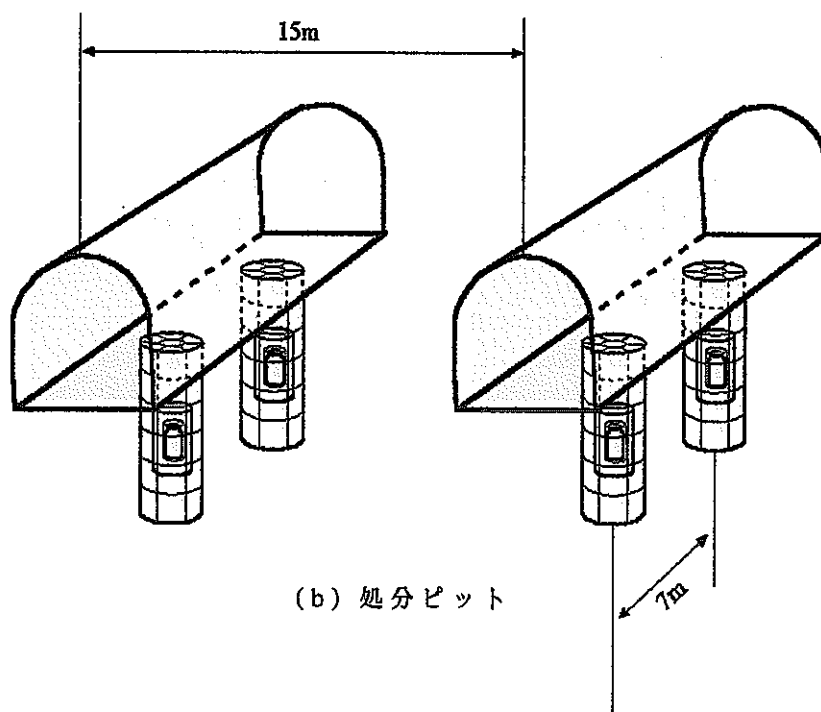
(b) 処分ピット

図3. 6-1 人工多重バリア処分オプション (単一廃棄物パッケージの構成)  
(1992年PNCによる)





(a) 処分坑道



(b) 処分ピット

図3. 6-2 . 近接する廃棄物パッケージ4体の構成を示す処分オプション

## 計算条件

単一廃棄物パッケージ（図3. 6-1 a）を用いてTOUGH2コードによる処分坑道のシミュレーションを行った。単一の廃棄物パッケージの坑道は円筒形の形状をしているために、放射状もしくは円筒状のTOUGH2グリッドが適切であると考えた。

しかし視覚化処理においては、TOUGH2コードによる処分オプションのシミュレーション方法が限定されてしまう。それは、視覚化処理ではデータが規則的な間隔をおいた直交（ $x$ ,  $y$ ）グリッド（垂直方向のもしくは $z$ 方向においては、間隔変更可能）上にあることが必要とされているが、放射状もしくは円筒状のグリッド作成をサポートしていないからである。

例えば、廃棄物とオーバーパック（1.04m）及び緩衝材の寸法（0.98）を正確に表わすためには、円筒、もしくは不規則な間隔におかれた直交グリッドをTOUGH2コードに適用することになる。そのため、視覚化においては、TOUGH2コードによるシミュレーションの結果を「簡略化」するような規則的な一般のグリッドへの補間が必要である。当初、TOUGH2と視覚化処理の両方と適合するような非常に細かい規則的なグリッドを検討した。しかし、細密なグリッドによる計算では、TOUGH2コードによる多相熱シミュレーションが非常に多量の計算を必要とするため、簡単なシミュレーションですらも非常に多くの実行時間が必要であった。

もう一つの解決方法として挙げられたのが、シミュレーションには目の粗い不規則なTOUGH2グリッドを適用し、その結果を非常に細かい規則的な視覚化処理グリッドに補間することである。この方法によって、TOUGH2の実行時間に関する制限が取り除かれる。その一方で、円筒形状を視覚化するために必要な視覚化処理グリッドブロックの数に比例して、コンピュータの必要とされるメモリ（RAM）が増加することにより限界が生じた。

以上のような理由から、処分坑道シミュレーションのために、計算実行時間とメモリ上の制限の双方において妥協できるグリッド間隔を選択した。この間隔を用いることにより、廃棄物パッケージ、ベントナイトで充填された坑道、母岩等の主要特性を近似化することにした。そして、ベースケースや感度解析に適用される単一廃棄物パッケージ坑道の3次元的なグリッド（図3. 6-3を参照）は、 $15 \times 8 \times 25$ の要素（全部で3000の要素）を有し、 $15\text{m} \times 7\text{m}$ の横方向の範囲、及び55mの縦方向の範囲を網羅した。

15mの $x$ 軸方向の間隔は、近接する坑道間の距離を表わしており、7mの $y$ 方向の間隔は坑道に沿った廃棄物パッケージ間の距離を表わしている（図3. 6-2 aを参照）。横方向の境界については、熱流動及び地下水流動はなしとした。縦方向の（ $z$ 軸）グリッド間隔は、一定した温度と圧力の上下の境界線から境界影響を受けずに、廃棄物パッケージの熱を消散するのに十分な容量が出来るように設定した。

コンピュータ処理の時間を多少なりとも少なくし、解析対象体積を増やすため、境界の近くで $z$ 軸方向の間隔を広げた。解析段階で広げられたグリッド間隔は、解析結果の視覚化においては1mのブロックに分割した。また、4つの単一廃棄物パッケージのシミュレーションを統合して、4体の廃棄物パッケージシミュレーション結果を表現することが出来るように横方向の離散化や境界条件を設定した。

このシミュレーションにおいて、廃棄物パッケージ及びオーバーパックは、2つのグリッド要素（ $1.0\text{m}^2$ の断面積、2.0mの長さ、 $2.0\text{m}^3$ の体積）によって表わされる。それぞれのグリッド要素において、熱が均等に発生するとした。この立方体による表示は、実際の形状（1.04mの直径、

1.95mの長さ、 $1.7\text{m}^3$ の体積)とはわずかに異なる。ベントナイト緩衝材は、円筒形状を立方体によってそのおおまかな形を表現することとした。緩衝材のグリッドの全要素は、 $3.0\text{m}$ 平方の断面及び $7.0\text{m}$ の長さ、 $61.0\text{m}^3$ の総体積の範囲を網羅する。実際の坑道緩衝材の総体積は、直径 $3.0\text{m}$ 、長さ $7.0\text{m}$ で体積が $47.8\text{m}^3$ となっている。

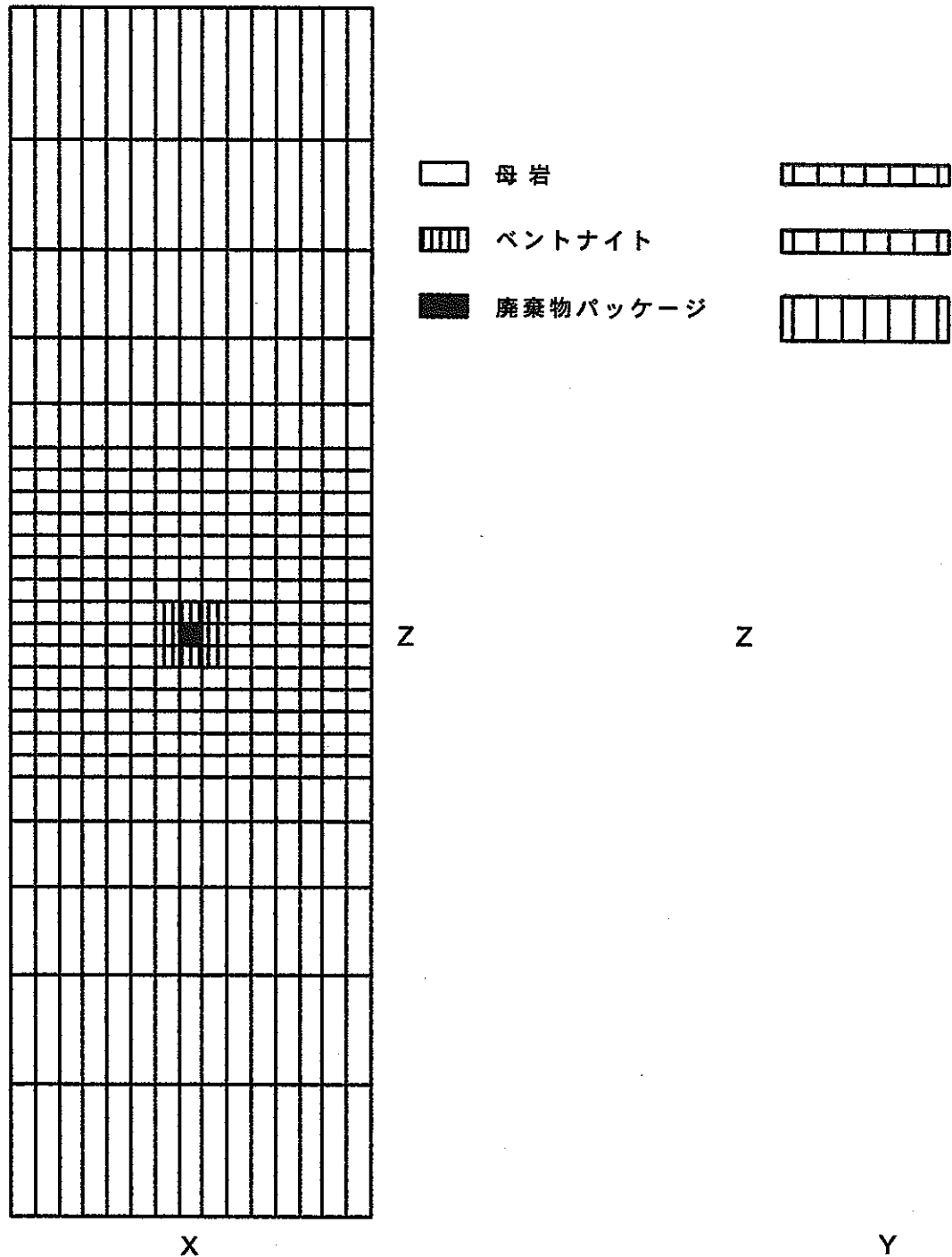


図3. 6-3 単一廃棄物パッケージ処分坑道のシミュレーション用TOUGH2グリッド

## 処分ピット

処分坑道のグリッドの場合と同じように、動画化プログラムとの互換性のためには、処分ピットにおいても規則的な間隔の直交グリッドが必要であった。これは、処分ピットが含まれている坑道の形状が複雑であるため、坑道付近では0.5m間隔のTOUGH2グリッド（図3. 6-4を参照）を用いた。形状が複雑であるために、処分坑道のグリッドと比較して、z軸方向全体で1mもの違いが生じている。

グリッド間隔を小さくすると、処分坑道のシミュレーションに比べて、約4倍多い数のグリッド要素が生成されることから、計算実行時間を削減するために単一廃棄物パッケージの1/4を用いて、TOUGH2コードによる処分ピットのシミュレーションを行った。

1/4廃棄物パッケージを処分ピットによってシミュレーションするために、直交グリッドを採用することにより、廃棄物パッケージ、緩衝材、碎石で充填した坑道及び母岩を近似化する必要があった。

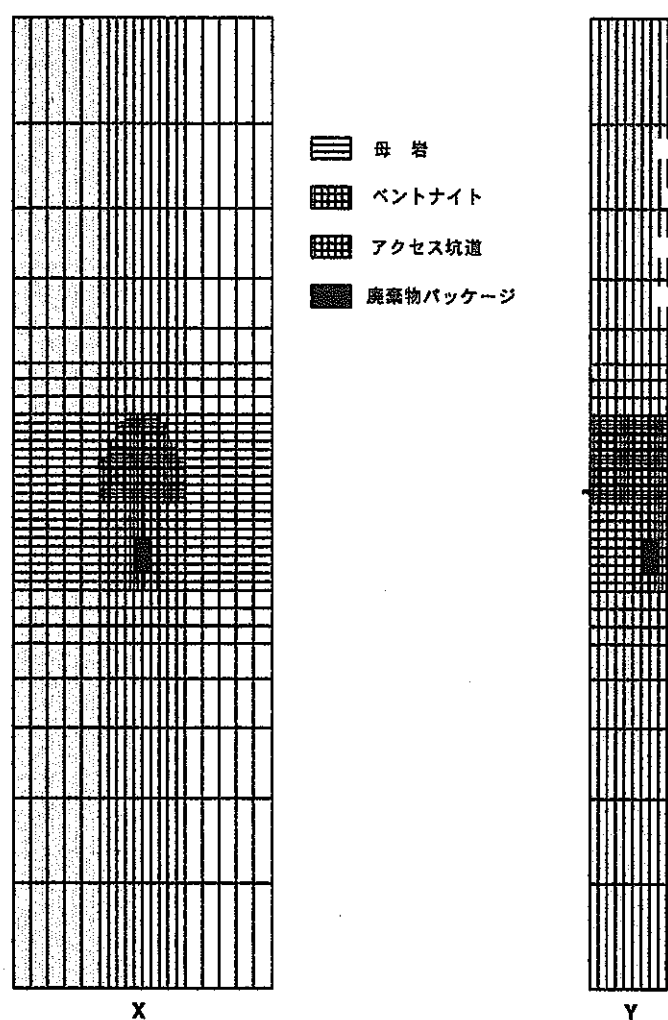


図3. 6-4 処分ピットのシミュレーション（単一廃棄物パッケージと同等物を示し対応する1/4の廃棄物パッケージの線り度を影で表示）

図3. 6-4には、単一廃棄物パッケージのシミュレーションに相当するグリッドが示されており、1/4廃棄物パッケージのグリッドが薄い灰色の陰で示されている。ベースケースや感度解析に適用される1/4廃棄物パッケージの処分ピットの3次元グリッドは、 $10 \times 7 \times 36$ の要素（全部で2520の要素）を有し、 $7.5\text{m} \times 3.5\text{m}$ の横方向の範囲及び56mの縦方向の範囲を網羅している。1/4廃棄物パッケージを4倍にした単一廃棄物パッケージの寸法は、 $15\text{m} \times 7\text{m} \times 56\text{m}$ である。これらの寸法は、処分坑道のグリッド寸法とほとんど同じである。

15mのx軸の間隔とは、豎置処分孔を含んだ隣接している坑道間の距離を表わしている。一方、7mのy軸方向の間隔とは、坑道に沿った処分ピット間の距離を表わしている（図3. 6-2bを参照）。横方向の境界は、熱流動及び流体流動無しとして設定した。処分坑道のグリッドと同様に、縦方向の（z軸）グリッドの間隔は、上下の一定した温度・圧力境界からの影響を受けずに廃棄物パッケージによる熱が消散されるように十分な間隔をおくように設定した。1/4廃棄物パッケージのシミュレーションの結果を4つ組み合わせることにより、一体の廃棄物パッケージに相当する結果が示されるように、横方向の離散化及び境界条件を設定した。さらに、単一廃棄物パッケージのシミュレーション結果を4体組み合わせ、廃棄物パッケージ4体分のシミュレーション結果を作成することにした。

1/4廃棄物パッケージ及びオーバーパックは、4つのグリッド・ブロック（ $0.5\text{m}^2$ の面積及び2.0mの長さ）によって表わされる。それらの一つブロックにおいて同量の熱が発生するものとする。 $1.7\text{m}$ の実際の体積に対して、それに相当する1つの廃棄物パッケージの体積は、 $2.0\text{m}^3$ である。単一廃棄物パッケージのベントナイト緩衝材のグリッド要素は、 $2.0\text{m}$ 平方の断面積及び5.0mの長さ、 $18.0\text{m}^3$ の総体積の範囲を網羅している。実際の豎置処分孔の緩衝材は、直径1.7m、長さ5.0m、総体積が $9.7\text{m}^3$ となっている。

豎置処分坑を含んでいる横置坑道は、規則的なグリッドに制約があるために、階段状のグリッドによって表現することになる。それに相当する単一廃棄物パッケージアクセス坑道グリッドの要素は、5.0mの高さ、7.0mの長さ、 $143.5\text{m}^3$ の総体積の範囲を網羅している。実際の豎置処分孔を含んでいる横置坑道は、5.0mの高さ、7.0mの長さ、 $156.2\text{m}^3$ の総体積の範囲である。

### シミュレーションパラメータ

TOUGH2コードにおいては、シミュレーションを行う岩石の水理特性及び温度特性に関する情報が必要である。仮想上の日本のサイトにおける岩石特性については、H3レポート（PNC1992年）において報告されているデータや、その他の関連した文献（「FreezeとCherry」1979年）などを用いて設定した。処分坑道及び処分ピットの両方式のシミュレーションで適用したパラメータ値を表3. 6-2に示す。

表3. 6-2 TOUGH2シミュレーションパラメータ

素材	特性	単位	範囲	シミュレート 評価	参考
廃棄物パッケージ	透水性	m <sup>2</sup>	N/A	10 <sup>-20</sup>	
	間隙率	—	N/A	0.001	
	容積密度	Kg/m <sup>3</sup>	2800 - 7800	5000	H3 (P.3-50)
	気孔圧	Pa <sup>-1</sup>	N/A	3×10 <sup>-9</sup>	
	熱電導率	W/m <sup>3</sup> °C	1.2 - 53.0	30.0	H3 (P.3-50)
	温度特性	J/Kg°C	460 - 960	700	H3 (P.3-50)
ベントナイト	透水性	m <sup>2</sup>	10 <sup>-17</sup> - 10 <sup>-20</sup>	10 <sup>-20</sup>	H3 (P.4-45) Freeze, Cherry
	間隙率	—	0.3 - 0.7	0.3	H3 (P.4-45) Freeze, Cherry
	容積密度	Kg/m <sup>3</sup>	1900 - 2100	2000	H3 (P.3-50)
	気孔圧	Pa <sup>-1</sup>	10 <sup>-7</sup> - 10 <sup>-9</sup>	3×10 <sup>-9</sup>	Freeze, Cherry
	熱電導率	W/m <sup>3</sup> °C	0.9 - 1.7	1.5	H3 (P.3-50)
	温度特性	J/Kg°C	600 - 1000	900	H3 (P.3-50)
母岩	透水性	m <sup>2</sup>	10 <sup>-14</sup> - 10 <sup>-19</sup>	10 <sup>-15</sup>	H3 (P.4-45) Freeze, Cherry
	間隙率	—	0 - 0.05	0.01	H3 (P.4-45) Freeze, Cherry
	容積密度	Kg/m <sup>3</sup>	N/A	2700	H3 (P.3-50)
	気孔圧	Pa <sup>-1</sup>	10 <sup>-8</sup> - 10 <sup>-10</sup>	3×10 <sup>-9</sup>	Freeze, Cherry
	熱電導率	W/m <sup>3</sup> °C	N/A	2.9	H3 (P.3-50)
	温度特性	J/Kg°C	N/A	840	H3 (P.3-50)

N/A：参考文献では、範囲について特に指定されていない。

廃棄物パッケージの特性は、ガラス固化された廃棄物とオーバーパックの平均値を表わしている。TOUGH2によるシミュレーションにおいて、これらの2つの素材が組み合わされて、単一グリッド要素となる。ベントナイトの緩衝特性は、H3レポートにおいて示されている乾燥値と飽和値の中間の状態を表わしている。ベントナイトにおける気体の飽和のTOUGH2の初期値は、30%と仮定した。従って、ベントナイト特性はH3飽和値の方に傾斜している。母岩特性は、ベースケースのシミュレーションのためのもので、それは、多孔性媒体に相当する複合体（母材と割れ目帯）を表わしている。不連続性亀裂の透水性は、10-13m<sup>2</sup>ほどの高い数値にもなり得る（PNC、1992年4～45ページ）。処分ピットオプションにおいて、アクセス坑道は10-15m<sup>2</sup>の透水性と0.3の間隙率を有していると仮定した。

均一の毛細管圧力と相対的な透水性機能を適用した。0.2の残留ガス飽和値（Sgr）、0.2の残留液体飽和値（Slr）、2.0の間隙の分配指数λ及び0.5MPaの大気流入圧力を有するTOUGH2のBrooks-Coreyの公式をすべての素材に適用した。

流体圧の初期値は、流体静力学的であると仮定し処分場の深さは500mと仮定、1000kg/m<sup>2</sup>の密度の水と9.81m/s<sup>2</sup>の重力による加速度によって、流体静力学的な流体圧を算出した。母岩とベン

トナイトの初期の温度は、30℃と推定され、深さが500mの処分場の一般的な地温勾配と一致するものである。廃棄物パッケージにおける初期の温度は45℃と仮定しガス飽和の初期値は、母岩においてはゼロ、ベントナイトにおいては0.3、廃棄物においては0.75と仮定した。

廃棄物が含まれているTOUGH2の一つグリッド要素に適用される熱の発生には、COGEMA製廃棄物パッケージ（図3. 6-5を参照）の発熱量データを適用した。COGEMAの発熱量については、H3レポートに説明されている。TOUGH2のシミュレーションにおいては、廃棄物は30年間冷却されると仮定されている。従って、図3. 6-5は、PNCのH3レポートで示された図を30年間分ずらして表にしたものである。感度解析オプションで使用する（JNFL）廃棄物パッケージの発熱量についても、図3. 6-5に示した。

今回のTOUGH2コードでは、廃棄体埋設直後から500年間のシミュレーションを行うこととしている。しかしながら、出力結果をみると、出力値の変化が頻繁に観察されるのは、最初の100年間である。

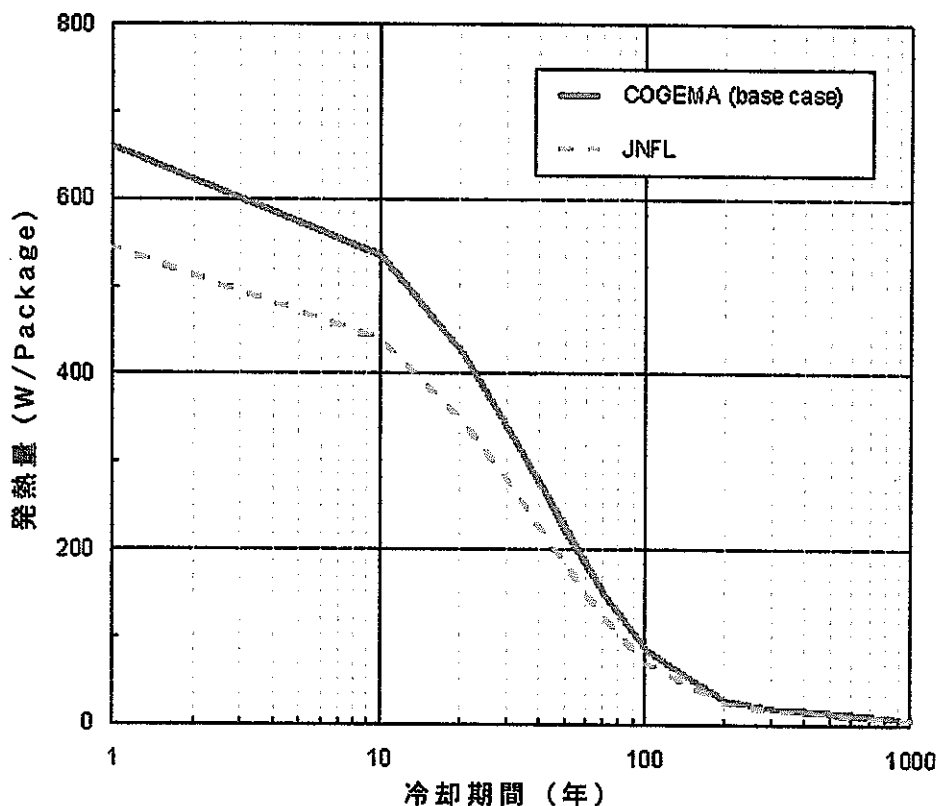


図3. 6-5 30年の冷却期間経過後のCOGEMA、JNFL両社の廃棄物パッケージ発熱量のシミュレーション

図3. 6-6にシミュレーション結果を示す。

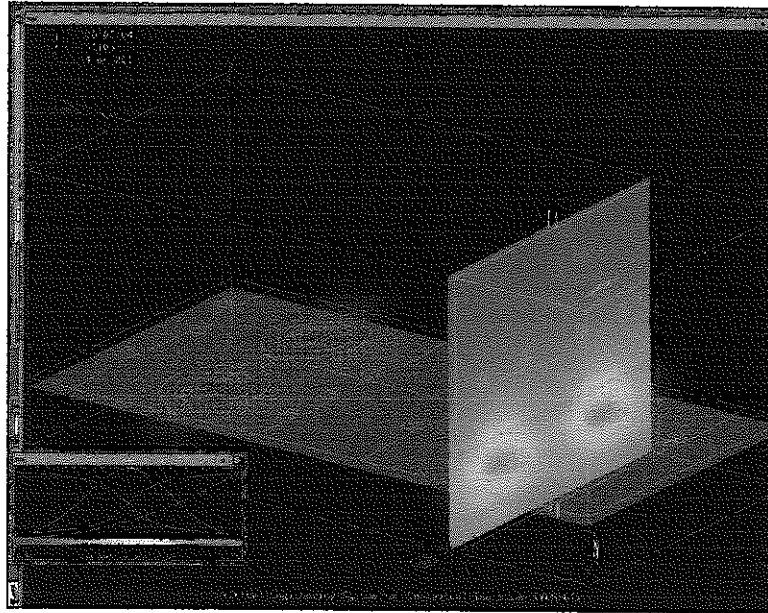


図3. 6-6 TOUGH2シミュレーション結果 (4体)



### 3. 7 関連研究課題6：地層中核種移行シナリオ・シミュレーション試作

シミュレーション・エンジンを設計する上で必要な技術について評価することを目的に、地層中の地下水移行シミュレーションモデルを試験的にシミュレーション・エンジンと接続することとした。利用したモデルは、3D-FEMWATERとした。3D-FEMWATERは広域水理を対象とした飽和・不飽和の多孔質媒体中の地下水流を解析する代表的解析コードであることと、地下水流を示すベクトル表現等に対して可視化技術の検討を行う必要があると判断したためである。3D-FEMWATERは、地層中の地下水移行を評価する3次元コードである。また、同コードは3D-FEMWASTEと接続して広域物質移行を評価することができる。この機能については、次年度以降の課題とした。

本年度は、開発しているヴァーチャル・エンジニアリング・システムの視覚化モジュールと同コードのインターフェイス周辺を試作し、3D-FEMWATERコードのビジュアル・シミュレーション機能の評価した。尚、3D-FEMWATERコードは3次元モデルを用いているため、視覚化モジュールとのインターフェイスはデータの受け渡しのみであった。以上のようにして3D-FEMWATERコードの計算結果を3次元で動画化し、ビジュアルなシミュレーションを行う基本的機能を確認した。以下に、シミュレーション画面の例を示す。

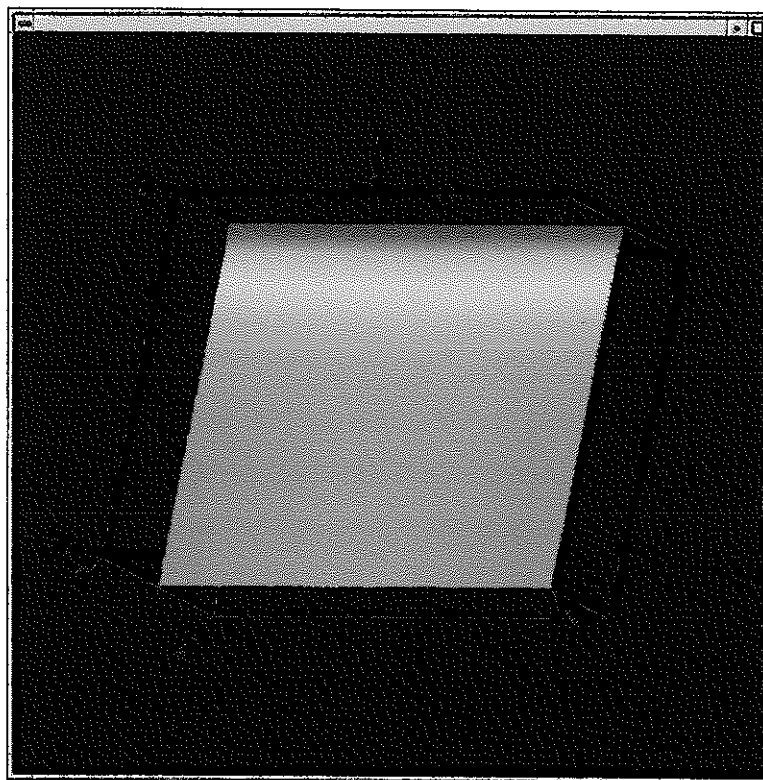


図3. 7-1 3D-Visual.FEMWATER開始画面

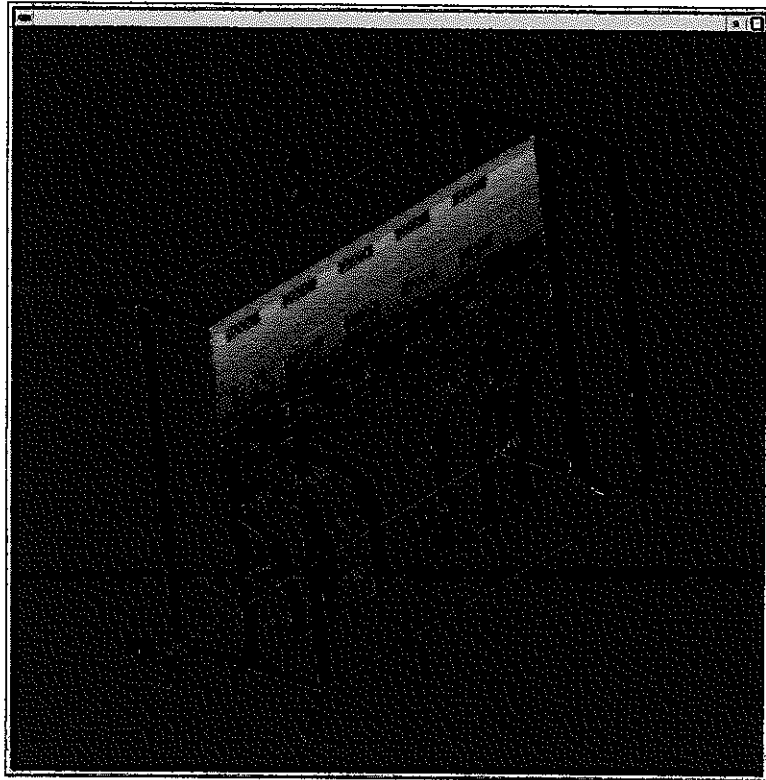


图 3. 7-2 3D-Visual.FEMWATER 动作画面

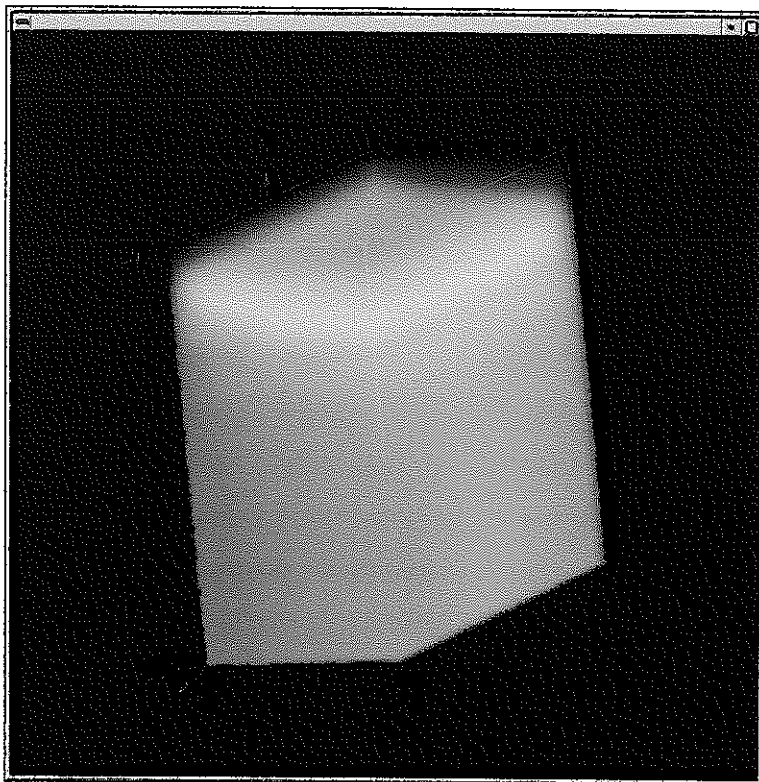


图 3. 7-3 3D-Visual.FEMWATER 动作画面

## (1) 試験用サンプルデータの作成

地層中の地下水移行モデルの試験用データ作成は、NUREG/CR-5387「LOW-LEVEL WASTE SHALLOW LAND DISPOSAL SOURCE TERM MODEL: DATA INPUT GUIDES」を参照してデータの作成を行った。この報告書は2次元のFEMWATER/BLTコードを対象としたものである。BLTはBreach, Leach, Transportの略である。作業項目は以下のとおりとした。

### 3DFEMWATER部分

- ・2次元版との入力データ項目のデータ項目の順序、入力形式等の違いを検討して対応。
- ・2次元のデータに対して奥行きを持たせて3次元化して入力データを作成。
- ・3DFEMWATERコードを実行し、流速等の出力ファイルを保存。

### 3DLEWASTE部分

- ・3DLEWASTEは、BLTのTransport部分のみに対応している。BLの部分は主として漏洩の生じる時間の計算を行っている。上記報告書から漏洩量の概略値を評価して、3DLEWASTEのソースとした。
- ・入力データを作成して、計算を試みたところ未定義変数の問題が生じたため、ソースを一部変更した。
- ・3DFEMWATERの出力を読み込み物質移動の過渡計算を行った。

#### a. 計算データ

NUREG/CR-5387で用いている2次元データの3次元化を行い計算を実施した。このデータは、non-site specificであり一般的な状況に対応しているモデルデータである。計算体系を図3. 7-4に示す。横56m、縦34mである。横方向は対称面の左半分である。右上部に埋設部がある。3次元に際して奥行き方向に15層長さ1mもメッシュを入れた。表面より2次元データと同一の127cm/yearの流入があったとして流れの定常解を求めた。対称面での流れの出入りはないものとした。それ以外の側面は自由流出、底面の左右のセルでは一定量の圧力値を指定している。これらの条件は着目している埋設部から充分離れているので計算に大きな影響を与えていないと判断される。土壤の物性は4種類であり、等方的な物性を用いている。

埋設部分の拡大図を図3. 7-5に示す。ハッチ部分が埋設位置であり、すべての埋設位置から漏洩があるとした。漏洩量は、図3. 7-6の平均値を使用した。上記の流速データを読み込んで、濃度変化の過渡計算を行った。

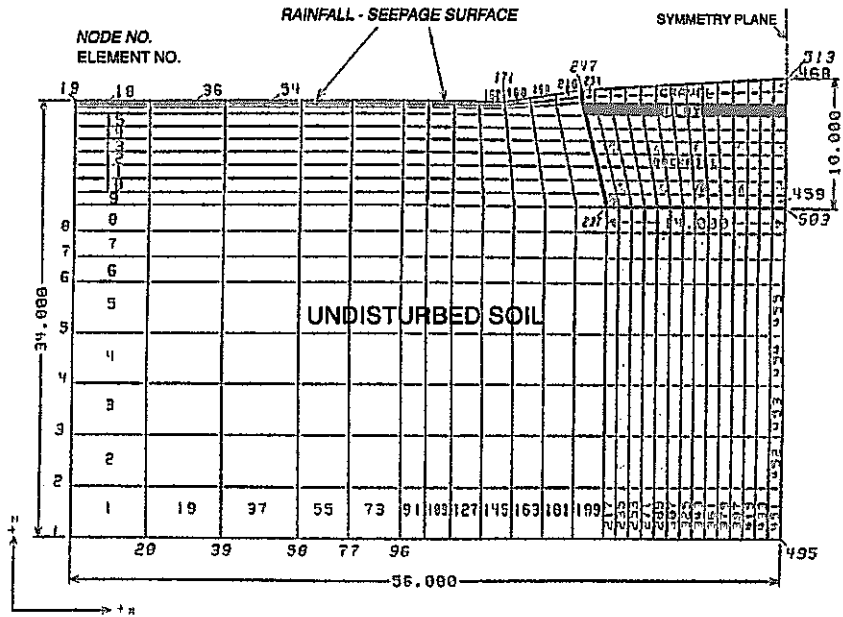


図 3. 7 - 4 計算体系

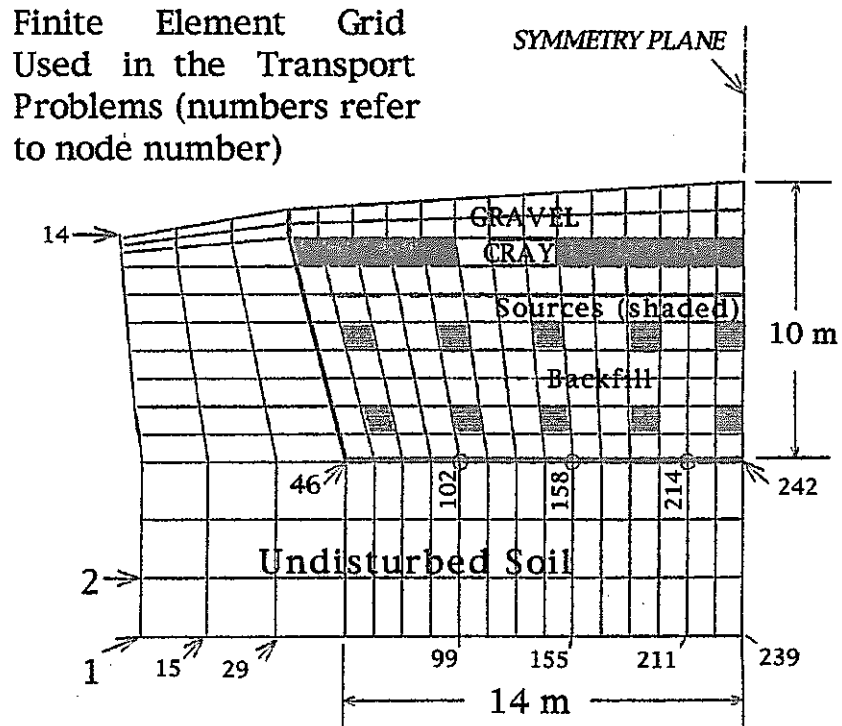


図 3. 7 - 5 埋設部分の拡大図

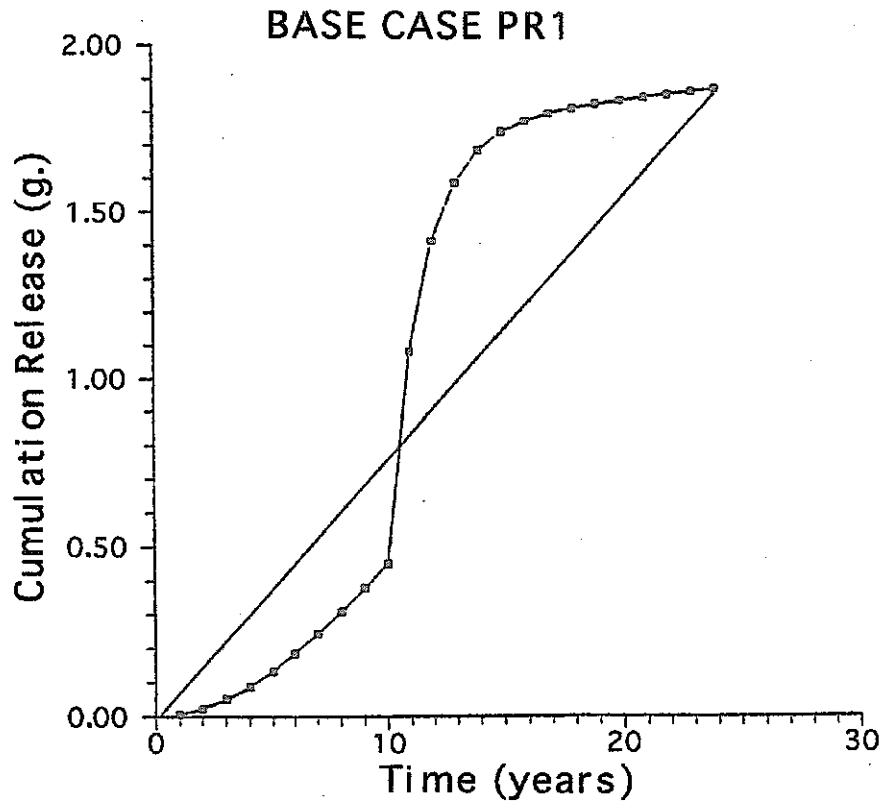


図 3. 7-6 漏洩累積

b. 3DLEWASTEの修正部分

SUBROUTINE DATAINで次の部分を修正した。

KVI>0の場合、LRLが未定義になるための修正

```
C*****DATA SET 5: material properties
```

```
C
```

```
READ (LUINP,1000) DATNAM
```

```
WRITE (LUOUT,2060)
```

```
DO 110 I=1,NMAT
```

```
READ (LUINP,*) (PROP(J,I),J=1,NMPPM)
```

```
WRITE(LUOUT,2070)I,(PROP(J,I),J=1,NMPPM)
```

```
WRITE(6,2070)I,(PROP(J,I),J=1,NMPPM)
```

```
110 CONTINUE
```

```
C
```

```
C$$$ Begin of reading geometric data
```

```
IF (KVI.GT.0) THEN
```

```
C reading node, element, and subregion data via unit 11 if KVI.GT.0
```

```
RFWIND(UNIT=I,UFI,W)
```

```
READ(LUFLW) TITLEM,NPROBLEM,NNP,NEL,NBNP,NBES,NTIM,NREGN
```

```
READ(LUFLW) (X(N),N=1,NNP),(Y(N),N=1,NNP),(Z(N),N=1,NNP),
```

```
> ((IE(M,I),M=1,NEL),I=1,9),(DCOSB(I,M),I=1,3),
```

```

>         M=1,NBES),((ISB(I,M),I=1,6),M=1,NBES),(NBES),(NPBB(N),
>         N=1,NBNP),(NNPI,R(N),N=1,NRF(GN),((GNI,R(N,I),N=1,
>         I.TMXNP),I=1,NRF(GN)
c-----
C/mri.kubota insert
      IF(IGEOM.LE.3) THEN
        CALL PAGEN
      I         (IE,NNPLR,LUOUT,
      M         GNLR,
      O         NLRL,LRN,LRL,LNOJCN,LMAXDF,NTNPLR)
      END IF
      IF(IGEOM.LE.1) THEN
C       identify boundary elements and compute directional cosines of
C       boundary sides if geometrics are read via cards
        CALL SURF
      I         (LUOUT,X,Y,Z,IE,Irl,
      O         DCOSB,ISB,NPBB)
      END IF
c--insert end -----
      ELSE
C
C***** DATA SET 6: node coordinate via card if KVI.LE.0
C
      NPI=0
      READ(LUINP,1000) DATNAM
c-----
c/kubota

```

表面メッシュ節点でDTI を1. E30とするとTRANSIENT計算で計算がすすんでも現象が変化しない事に対する変更

```

      SUBROUTINE BTGN
      I         (X,Y,Z,VX,VY,VZ,IB,NI,RI,I,RI,IE,CP,MAXNP,MAXFI,DELT,
      I         NNP,I,LUOUT,NTI,MXKBD,
      M         CS,DTI)
      ;
250 CONTINUE
c
      IF(IJUDGE.EQ.1)THEN
        KCOUNT=KCOUNT+1
        IJUDE=2
        GOTO 80
      ENDIF
C
      IF( (IB(N).NE.0) ) THEN
c.mri.kubota-----
c       DTI(N)=1.0D30
c       CS(N)=CS(N)
c-----

```

GOTO 950  
ENDIF

図3. 7-7から図3. 7-14に表示例を示す。

demo no 1 h9.12.

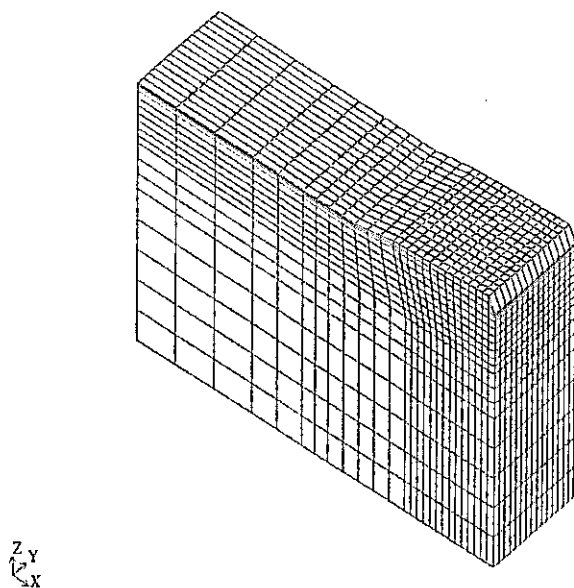


図3. 7-7 表示例

demo no 1 h9.12.

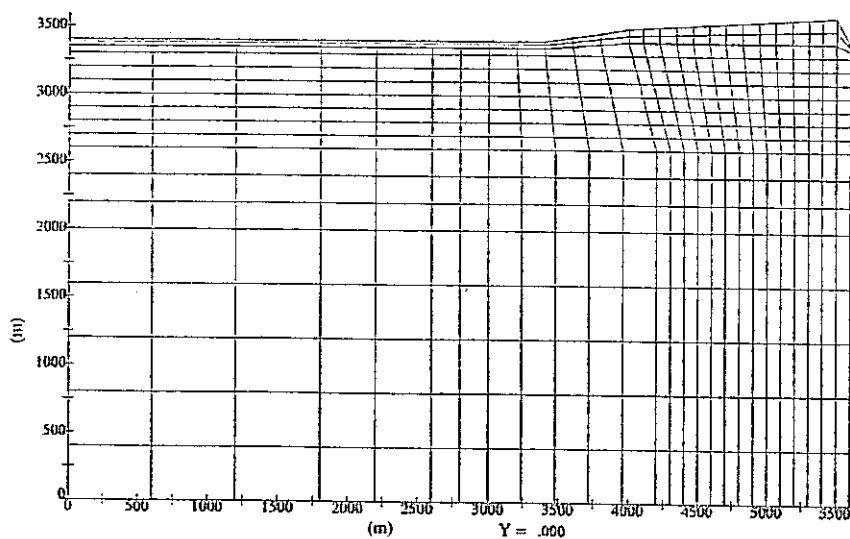


図3. 7-8 表示例

demo No.1 h9.12.22  
H:pressure head

$\cdot 10^{-3}$

V:Darcy velocity

TIME = .0000  
 $\rightarrow 4.6000 E - 4$

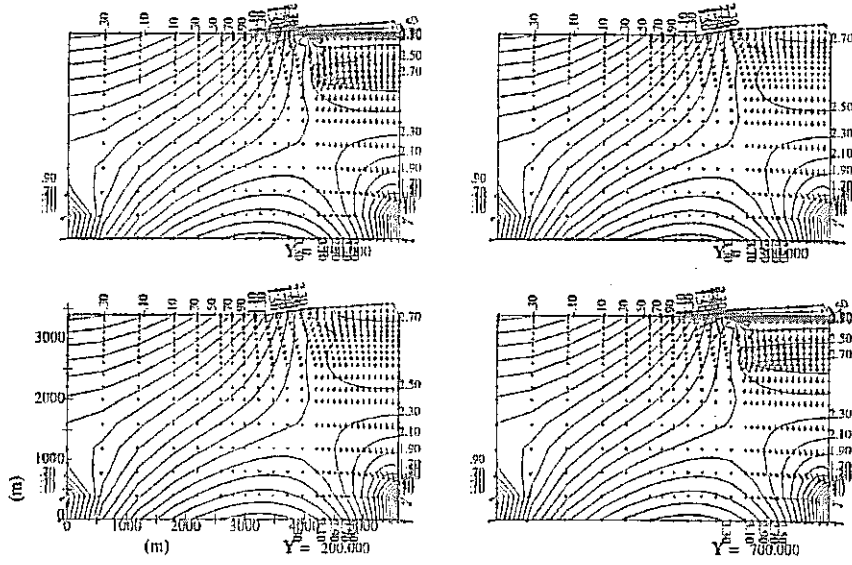


图 3. 7-9 表示例

demo No. 1 h9.12.22  
H:pressure head

$\cdot 10^{-3}$

V:Darcy velocity

TIME = .0000  
 $\rightarrow 2.0000 E - 5$

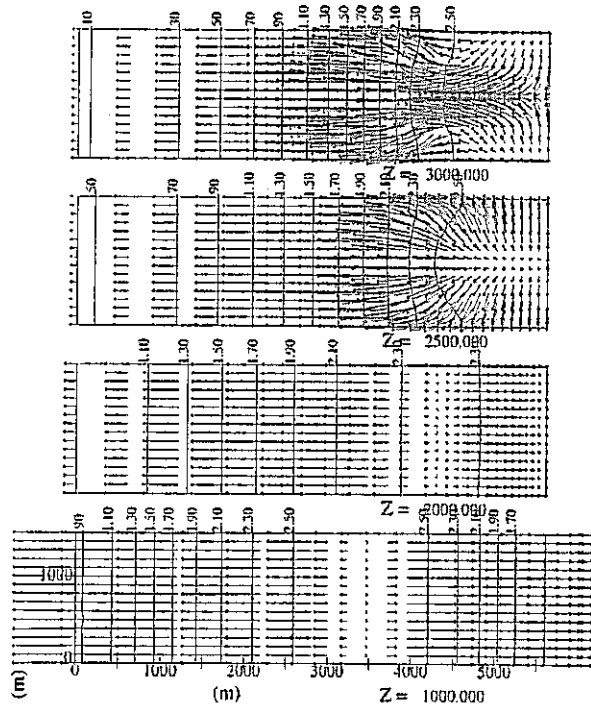


图 3. 7-10 表示例



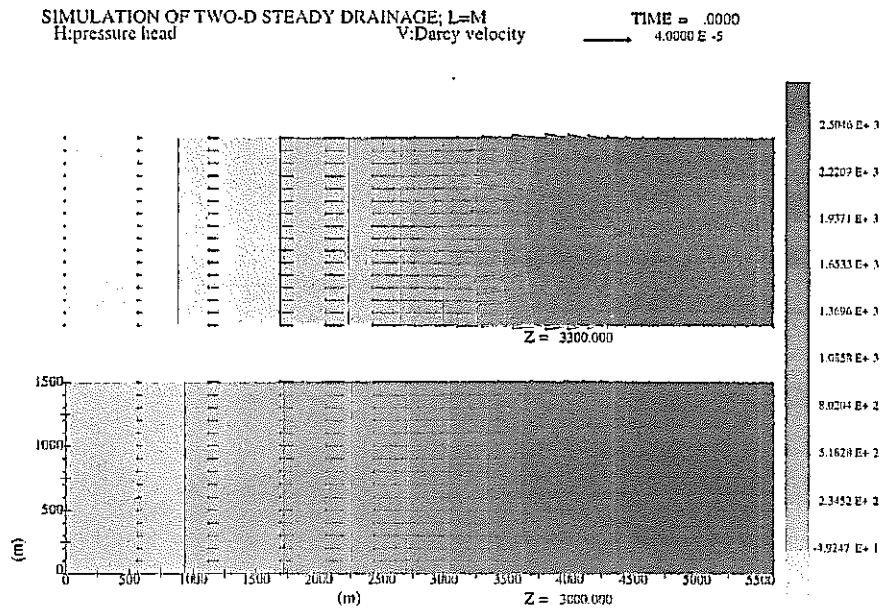


图 3. 7 - 1 1 表示例

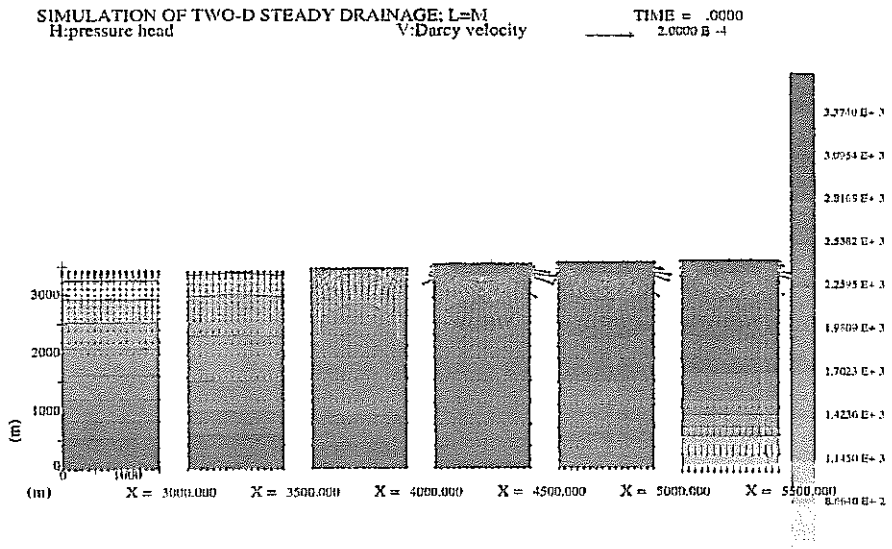


图 3. 7 - 1 2 表示例

demo no 1 h9.12.  
C:Concentration

$\cdot 10^{-4}$

F:Material Flux

TIME = 1.5700 E+ 6  
 $\longrightarrow$  10.000E -9

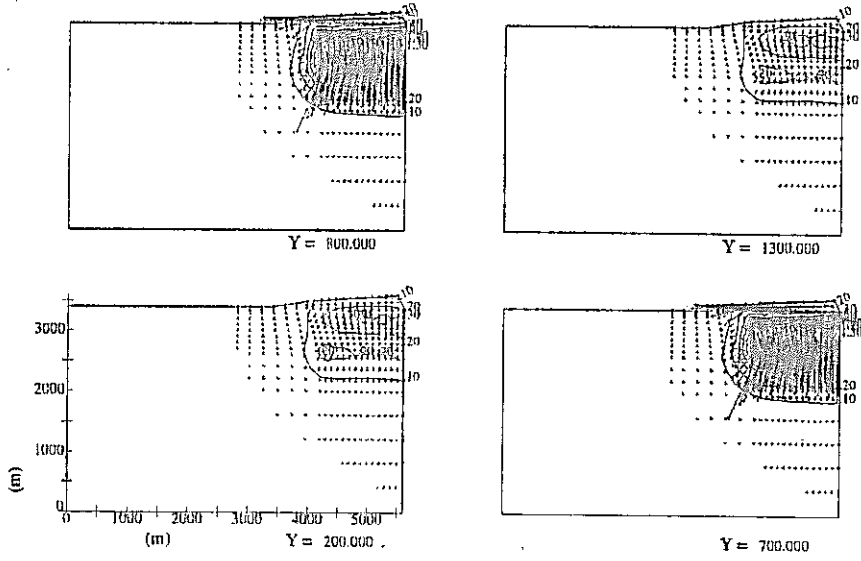


图 3. 7 - 1 3 表示例

demo no 1 h9.12.  
C:Concentration

$\cdot 10^{-4}$

F:Material Flux

TIME = 9.5850 E+ 6  
 $\longrightarrow$  2.0000 E -8

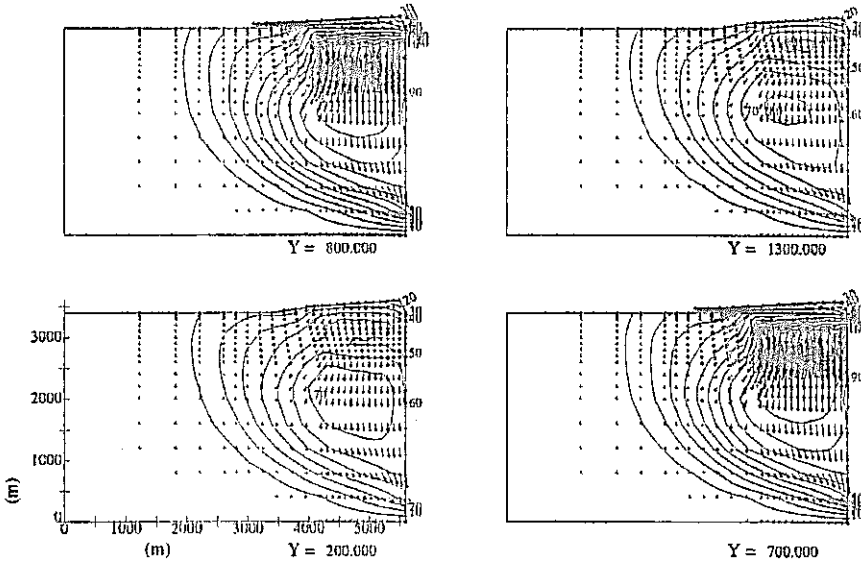


图 3. 7 - 1 4 表示例

## (2) 試験用データの作成

前(1)のサンプルデータをベースにして試験用データ作成の計算を行った。計算体系として30Km四方のものと処分場近傍の2種類を用いた。

計算の設定の変更点は次のとおりである。

- ・物性としては処分場はbackfil、その他の場所はsoilを使用した。
- ・年間降雨量は、130mm/yearとした。

計算の要素数はつぎのとおりである。

- ・36000要素 (40 x 40 x 30)
- ・21600要素 (24 x 30 x 30)

計算は、サンプル計算と同様に流れの定常場を求め、次に物質移動の過渡計算を行った。計算体系図、濃度のグラフ、流速ベクトル、濃度の等高線を次図から示す。

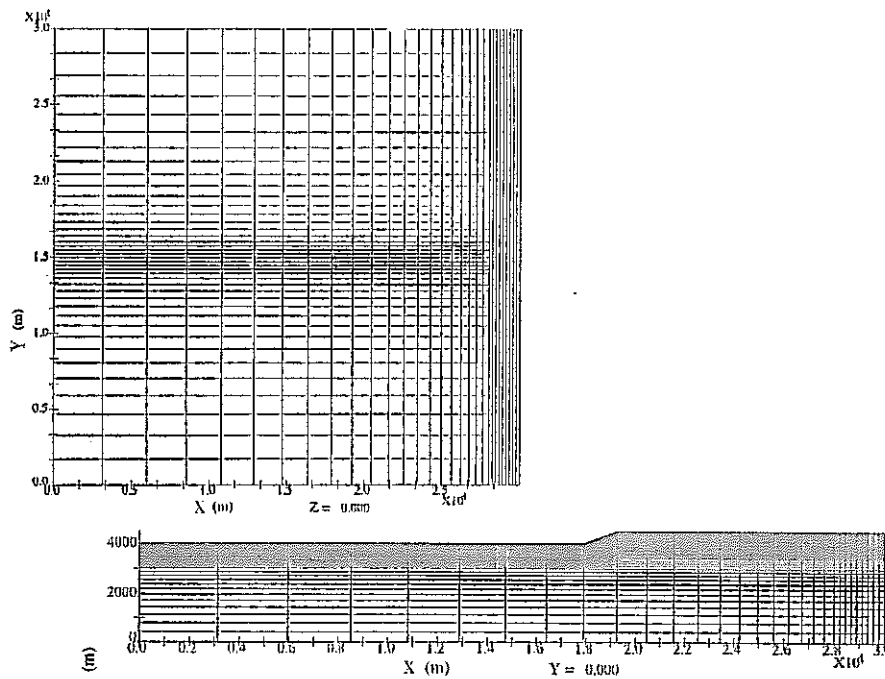
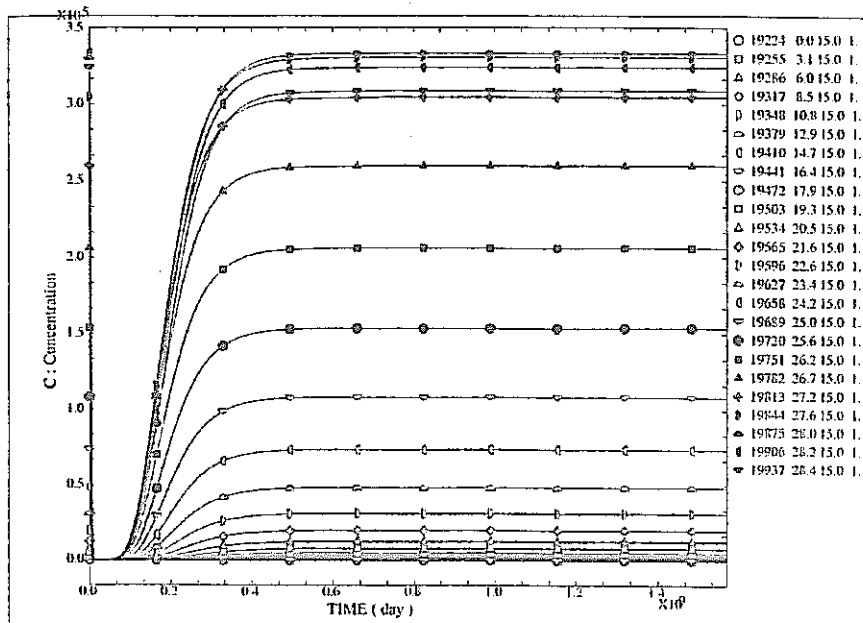
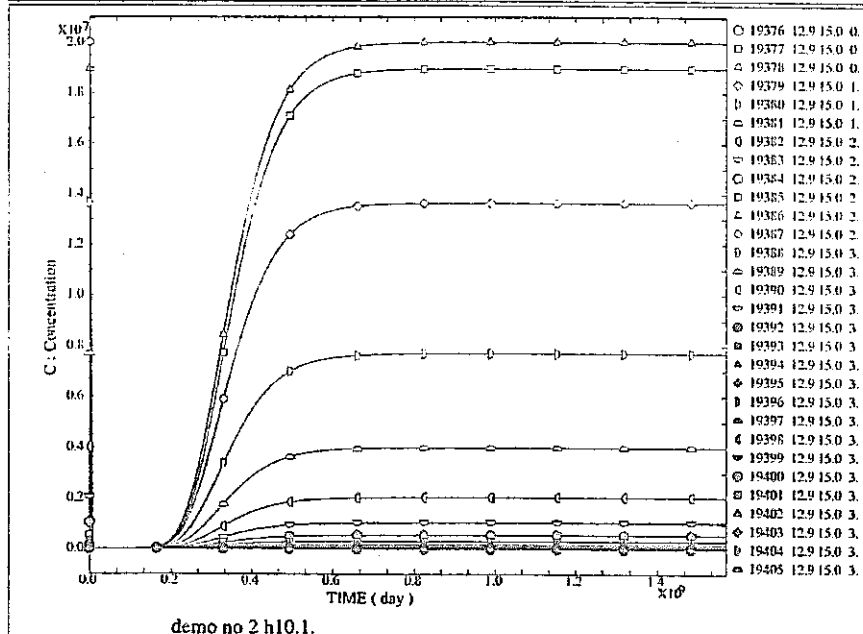


図3. 7-15 表示例



demo no 2 h10.1.



demo no 2 h10.1.

图 3. 7-16 表示例

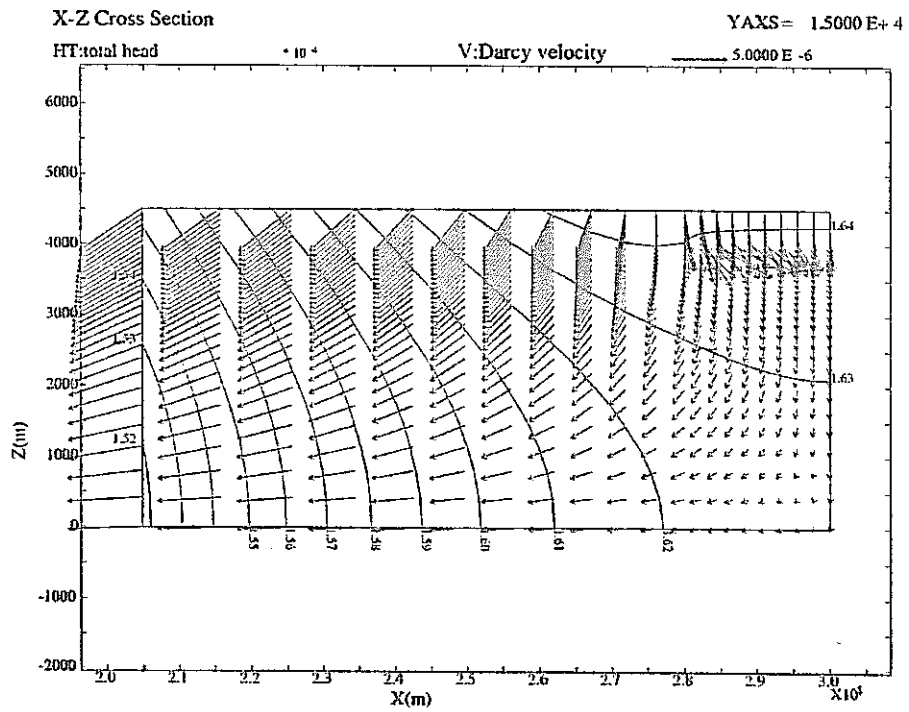
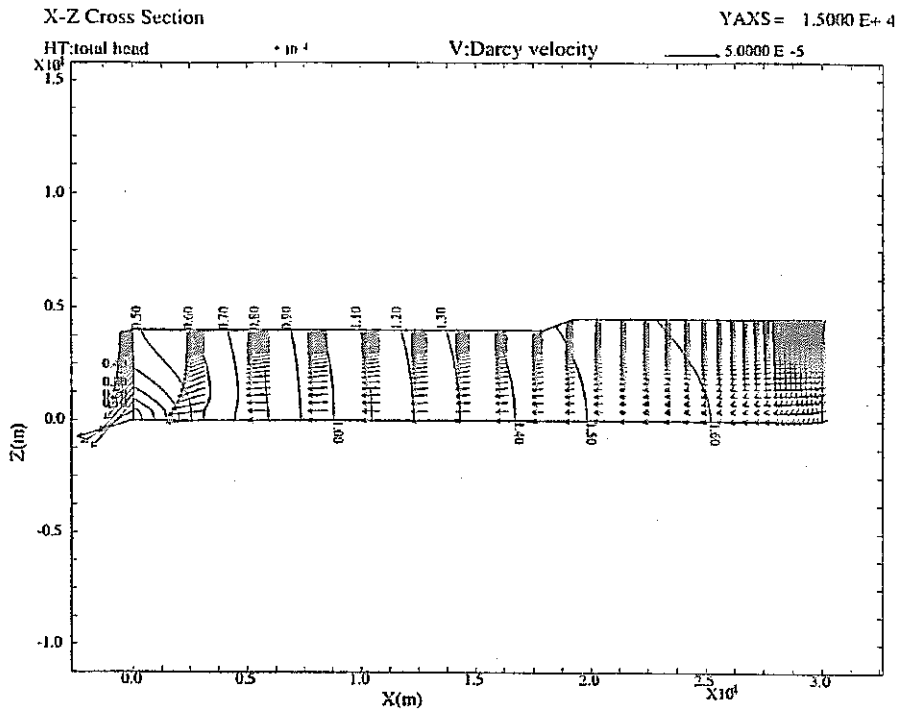


图 3. 7-17 表示例

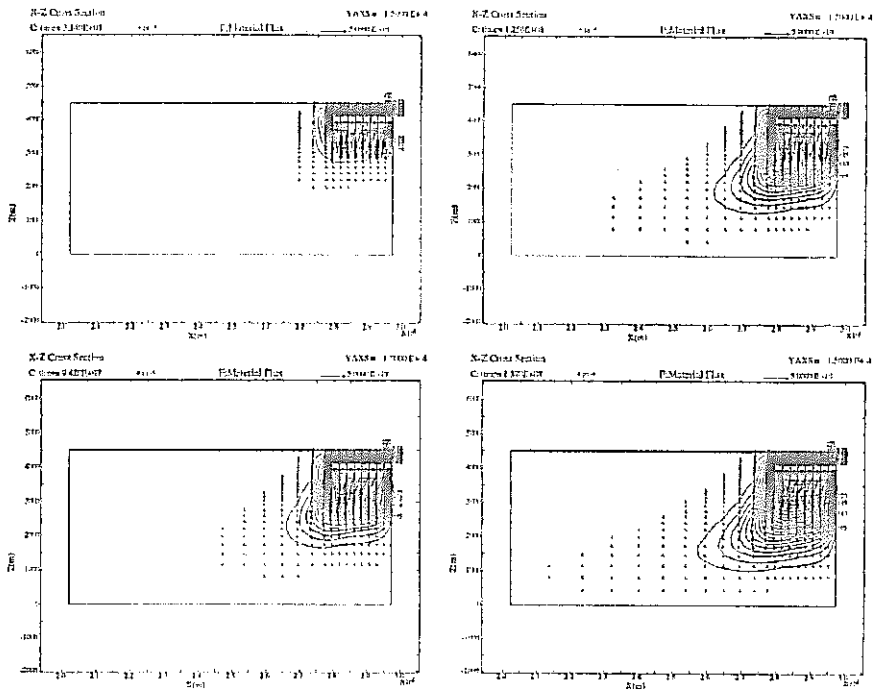


图 3. 7-18 表示例

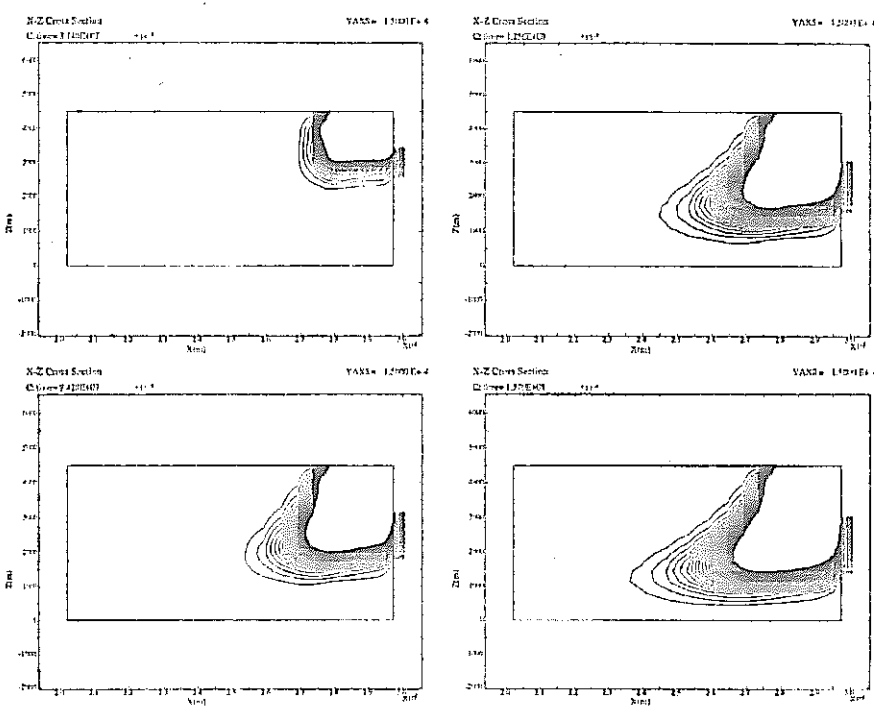


图 3. 7-19 表示例

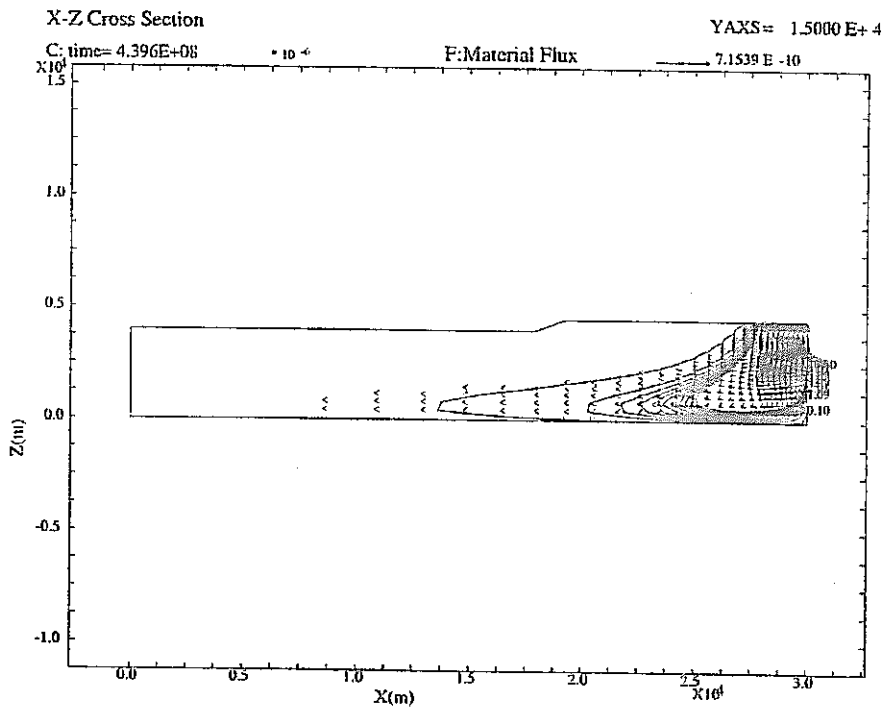
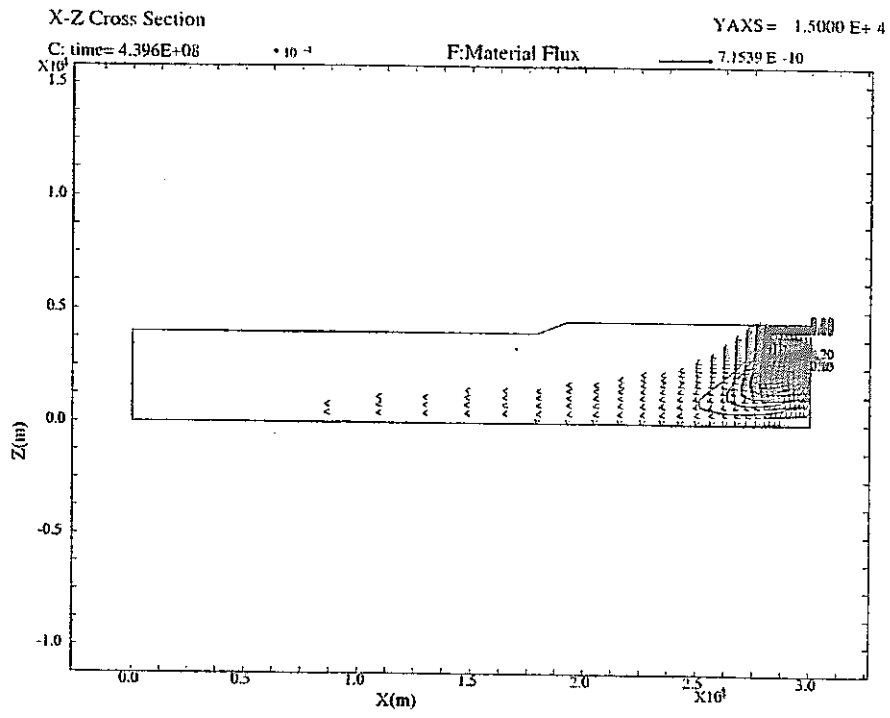


图 3. 7-20 表示例

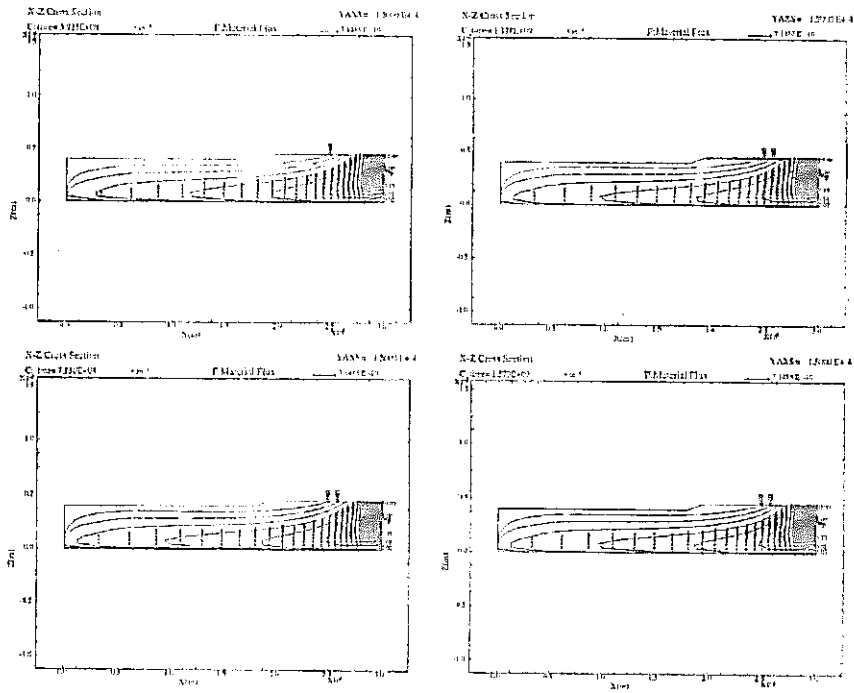


図3. 7-21 表示例

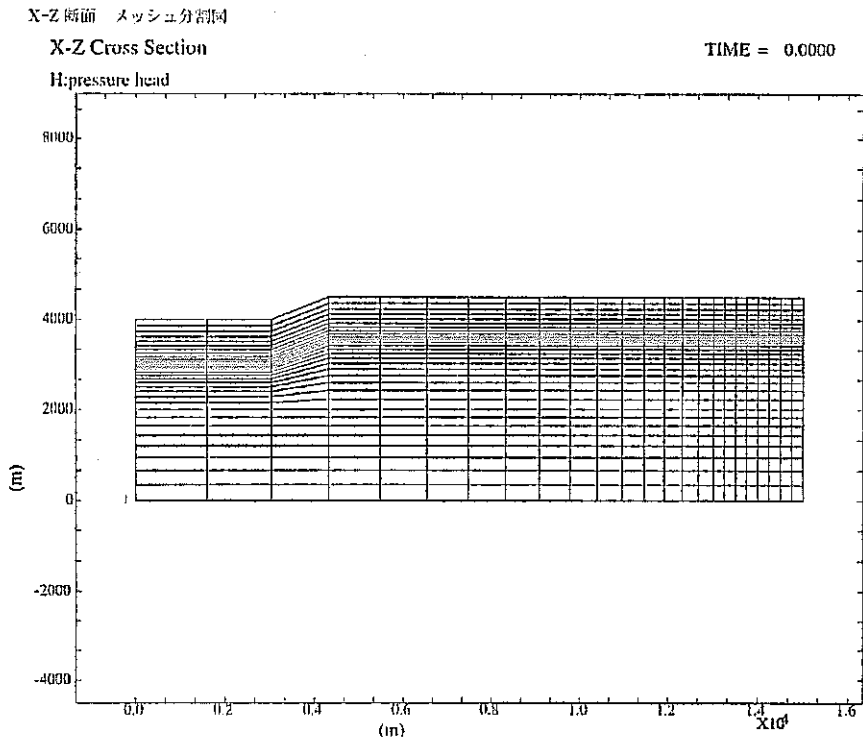


図3. 7-22 表示例



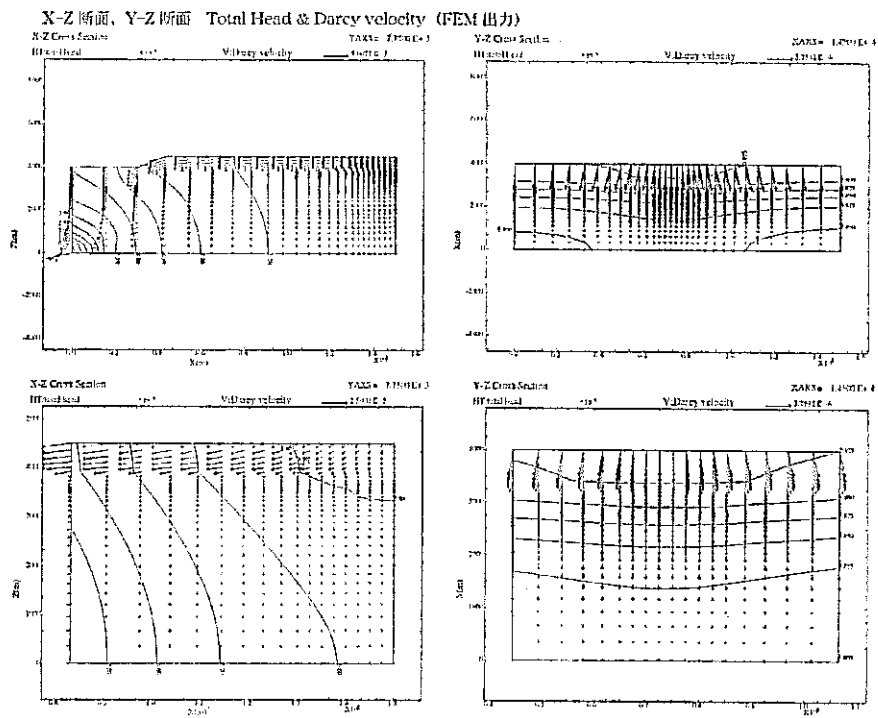


图 3. 7 - 2 3 表示例

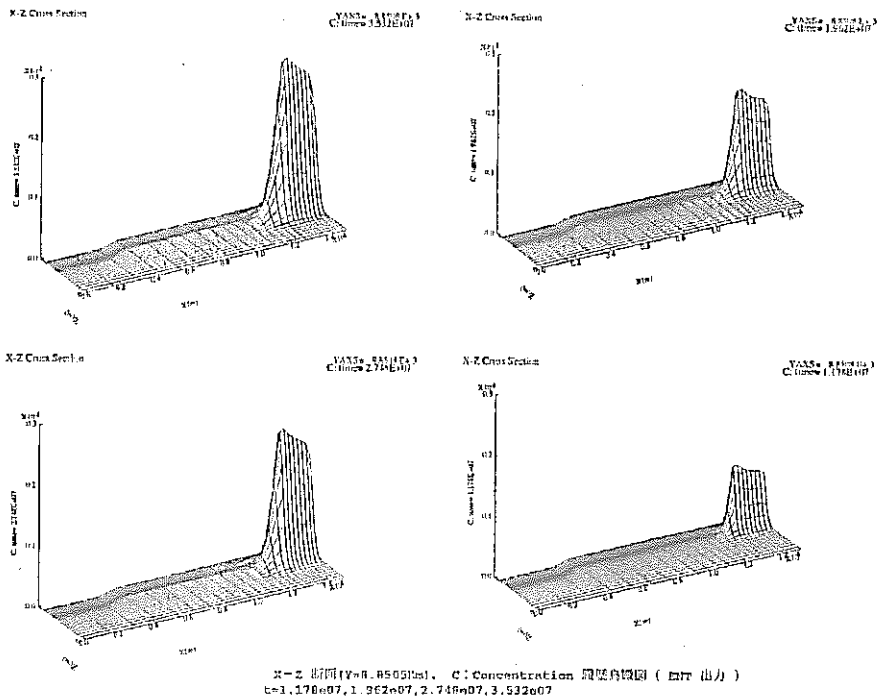
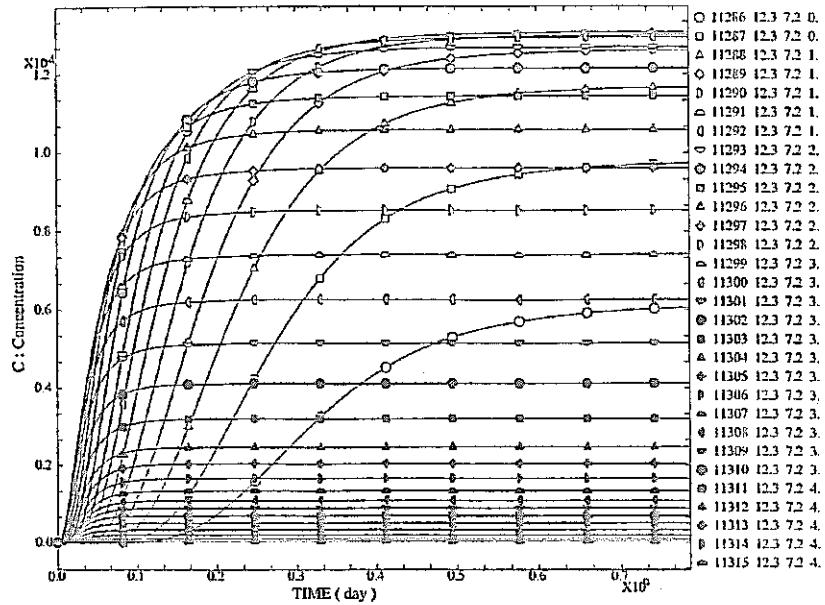
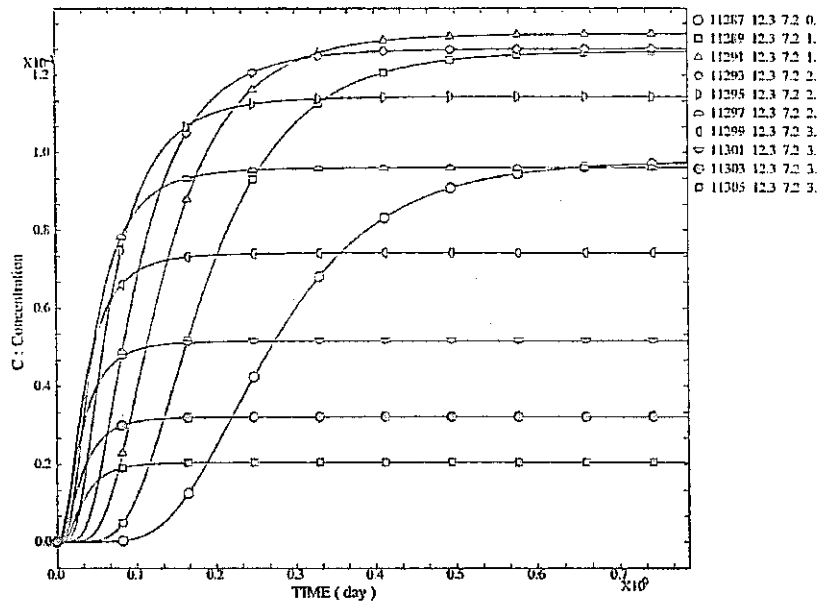


图 3. 7 - 2 4 表示例

節点 11286~11315 Concentration 履歴カーブ (t=0~7.85e08)  
 x=12.306 Km y=7.25 Km, z=0.35117~4.5 Km



demo no 2 h10.1.



demo no 2 h10.1.

図 3. 7-25 表示例.

## 4 データベース管理機能の基本設計

### 4.1 データベース管理機能基本設計仕様

データベース管理機能概念設計は、技術評価等に資するため個々のデータベースについて試作し、データベース間で異なる形式の情報交換や対話システムの要求に応じた情報の組み立てが行えることを検討した。本システムのデータベースは下記のとおりに分割して設計を行った。

- ・シミュレーション・シナリオ・データベース
- ・シミュレーション・モデル・データベース
- ・試験データ等、数値データベース
- ・品質管理データベース

データベース管理機能は、上記の各データベースに蓄積格納する本システムが扱う様々な情報を、対話システムの要求に対して、各データベースそれぞれから必要なデータを取り出し要求に応じた情報として提供するものである。

本システムのデータベース管理機能概念を図4. 1-1に示す。

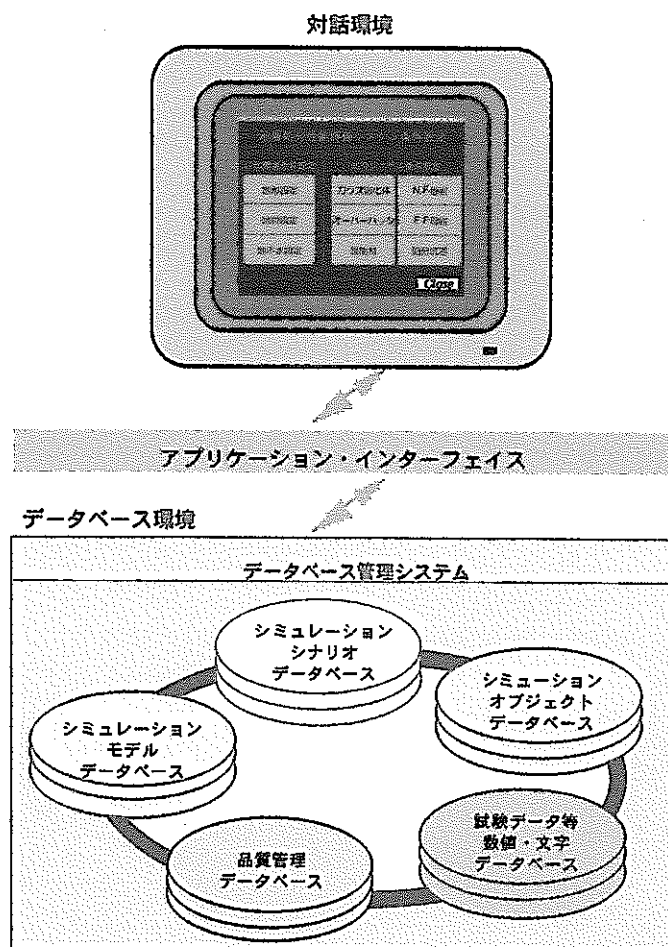


図4. 1-1 データベース管理機能概念

図4. 1-1は、本システムの情報を分野別に異なるデータベースを統合したマルチ・データベース概念を示したものであり、全体としてはデータベース管理システムおよびアプリケーション・インターフェイスを介して対話システムの操作要求によって機能するものである。各データベースは、対話システムから与えられた要求に対し、それぞれのデータ分野範囲で情報を取り出す。対話システムの要求は、1つの特定された情報（例えば、核種移行に係わるシナリオ検索）に限らず、「評価シナリオに沿って評価シミュレーションを実行する」といったデータベースから見ると複合した情報の集まりである。そのためには、データベース間で何らかのインタラクティブな環境を作り上げる必要がある。ここでいう「インタラクティブ (Interactive)」とは、相互作用的、つまり、お互いに影響を及ぼし合うという意味である。コンピュータ（データベース）の情報処理能力を利用して、人間の発想や既存概念による環境（評価モデル）への働きかけ、そして環境からのフィードバックに新たな内容を加えようとするものである。コンピュータに蓄積した情報をもとに、より自然で暗黙的に人間の意図が環境に伝わり、環境は人間の認知的負担を軽くするような適切な応答を返し、その結果、人間が情報世界と密に交流できる基盤を創作することが大事である。そのためのコンピュータ環境の在り方は次の2つの視点から捉える。ひとつは、地層処分といった広範囲な知識を必要とする研究において、コンピュータを独立した専門知識を蓄積している存在であるということ、ひとつは、生活圏の行き届かない（地下1,000mや海底10,000mの世界）環境を持ち備えることができるといったものである。このコンピュータ環境と人間が対話をして意図を伝達し、ある時はコンピュータが人間の代理となって仕事を行ったり、ある時は互いに協力して仕事をしていくといった考えを、ここでは「エージェント指向」と呼ぶこととした。

本システムで人間と対話するエージェントは、他のエージェントと協力して仕事を行うことも考えられる。つまりエージェントとは、ある種の自律性を持ち、人間を含む他者とコミュニケーションを行って自分の行動を決定することのできるシステムである。このようなエージェント指向インターフェイスは、ヒューマン・インターフェイスに自律性や協調性を持たせて、ヴァーチャル・エンジニアリング・システムにおける人間のパートナー的存在を構築しようという試みである。

図4. 1-2から、インターフェイス・エージェントは主にアプリケーション・プログラムと人間との仲立ちをするエージェントであり、シナリオ・エージェントやオブジェクト・エージェントといったエージェントは、ソフトウェア・エージェントと呼び情報検索などの人間の仕事を代行するエージェントである。これらエージェントは、人間からの要求に対してそれぞれの役割に関する情報を提供し、役割不足や補足の情報に関しては、他のエージェントに委ね最終的に1つの回答をもって結果を返す。発展的（将来的）なエージェントの在り方は次のとおりと思われる。

- ・能動性（あるいは自発性）：エージェントが主導権を持ってインタラクションができること。
- ・自律性（あるいは適応性）：環境の変化に適応してエージェントの判断で行動できること。
- ・社会性（あるいは協調性）：個々のエージェントがバランスしていること。
- ・意図の世界：ユーザーの曖昧な要求からその意図を推論できること。
- ・擬人性：人間的な外観や挙動を表示すること。
- ・代理性（あるいは信頼性）：人間から権限を委譲されてタスクを遂行すること。
- ・安全性：権限を委譲した人間が不都合になるような行動をとらないこと。

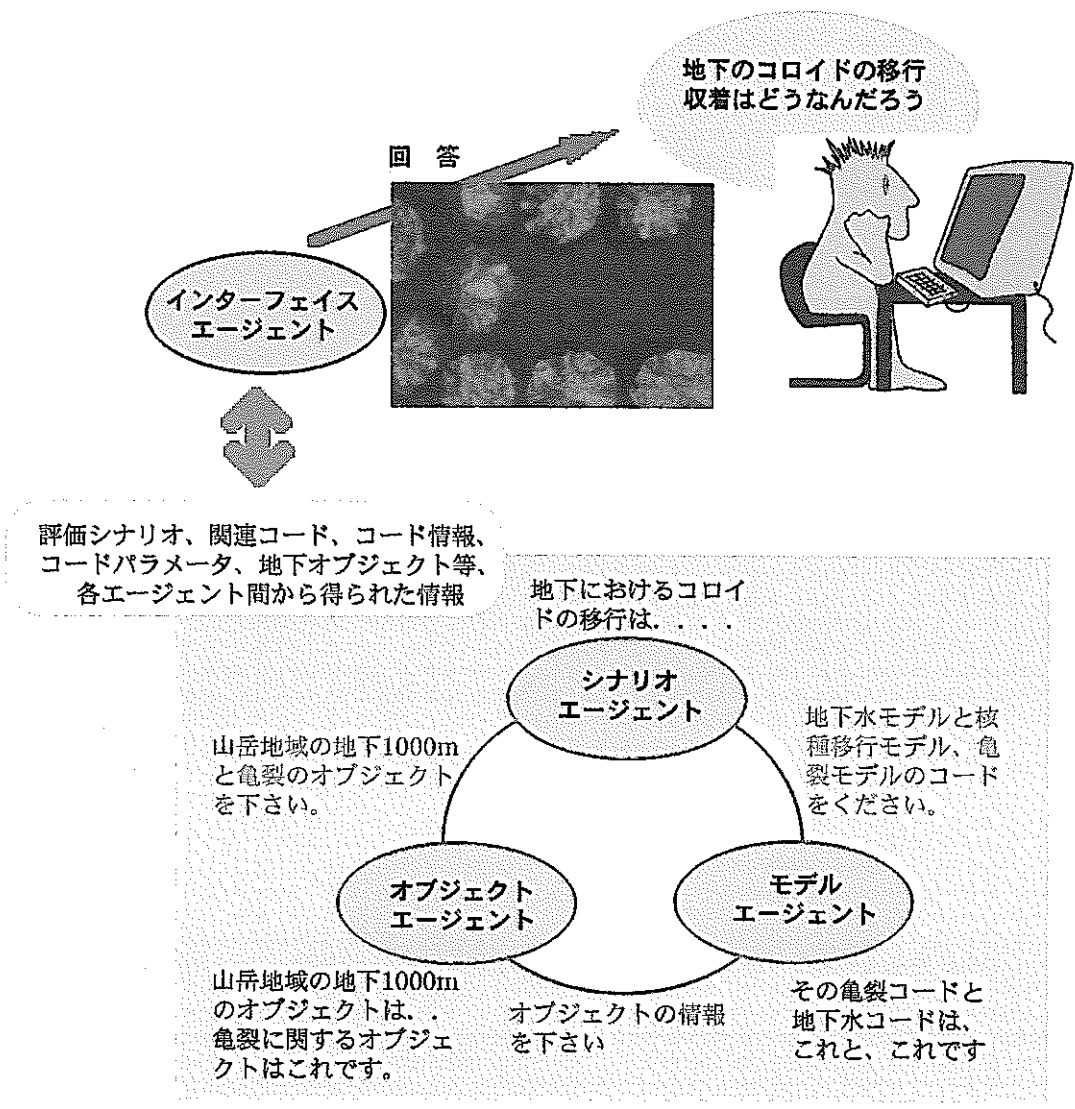


図4. 1-2 本システムにおけるエージェントの考え方

本年度のデータベース管理機能の設計では、このエージェントを本システム操作対話機能の一部と捉え、人間からの要求に対する情報を取り持つデータベース間のインターフェイス概念機能の検討を行った。検討においては、下記に示す本システムで取り扱う情報を、本システムでいうエージェントからの要求に応じたデータを提供できるためのデータベース・モデルの検討、データベース管理システム機能、データベース管理システム環境の概念設計およびオブジェクト指向技術の応用性について行った。

- ・地層処分性能評価のためのシミュレーション・シナリオ情報
- ・シミュレーション・シナリオに接続するモデル（評価コード）情報
- ・処分システムを構成する物理的な機器、構造物形状データ
- ・試験データ、文献データ等の数値、文字データ
- ・プログラム品質管理情報、数値データ品質管理情報および文書情報品質管理情報

#### 4. 2 データベース・モデル

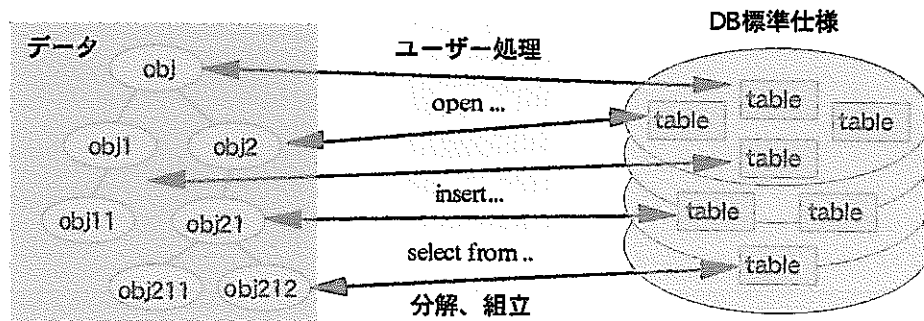
一般的にデータベース・モデルには階層型データベース、ネットワーク型データベースおよびリレーショナルデータベースがあり、最近の傾向としてデータベース・システムという表（テーブル）を基盤としたリレーショナル・データベースを指す場合が多い。データベース・モデルの相違は、論理的にはデータベースを構成する基本的な単位の情報（データ）を様々なデータベースシステムを相互に区別する判断基準として、この情報がどのように組織されているかである。

ネットワーク型データベースでは、情報は項目ごとに例えばカードとしてバラバラに存在させ、関係するカードを糸で結ぶという形態で情報を組織している。情報の検索時には目的とするカードを設定した後、そのカードを引き寄せることによって、そのカードの配下の糸で結ばれたカードを収集するといった方法を用いる。

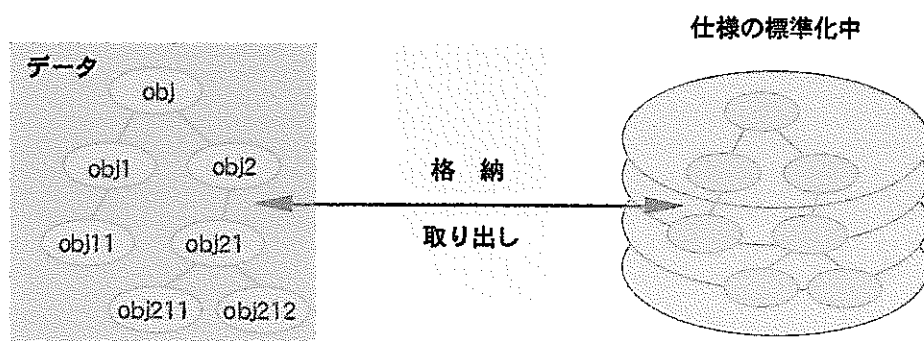
リレーショナル・データベースの情報の組織は、データを行と列をもった2次元の表（名前を持ったテーブル）で表し、表と表の間に相関を持たせることが出来るものである。データベースへのアクセスは標準化されたSQLで行い、データの構造や変数の型などを意識する必要がなく、複数のランダム処理に対しても整合性がとれる。

オブジェクト指向データベースは、相互に関連し合うオブジェクト（データとロジックが一對となった情報）をその関連を保持した状態で蓄積、検索することを可能としたデータベースである。このオブジェクト間の関係をクラスという構造にまとめ、表現することにより、機能の重複を防ぎ、部品の再利用を促すことによってソフトウェア開発や保守の生産性の向上を図る。

オブジェクト指向データベースは近年様々な場所でよく耳にするが、テーブル概念やSQLといった標準化がされているリレーショナル・データベースに比べて、まだまだオブジェクト指向データベースは標準化の具体的確立がなされていないと思われる。それは、オブジェクト指向に関する用語や概念は、複数の分野でほぼ同時発生的に形成されてきたものであり、オブジェクト指向に携わる技術者も、これらの異なる分野に立脚点をもつ場合が多い。このことが、オブジェクト指向に関する用語や概念の混乱に拍車をかけ、標準化を困難にしているといえるからである。これは、前述した「エージェント」にもいえると思われる。しかしながら、オブジェクト指向データベースはその考え方からまもなくデータベースの主流となるであろう。それは、近年のアプリケーションの開発ではオブジェクト指向技術が主流となりつつあり、オブジェクト技術には、プログラムコードの再利用性や拡張性の向上および開発期間の短縮といった利点があり、データベースの構築を含めアプリケーション開発全体にオブジェクト指向技術を適用することは効果的、効率的であると考えられる。アプリケーション上で生成されたオブジェクトに永続性を持たせるためにオブジェクト指向データベースシステムを利用する場合、ガラス固化体やオーバーパックおよび緩衝材といった人工バリア、それに関わる天然地形データ等の複雑な構造のオブジェクトをそのまま格納し読み出すことが可能であり、プログラムとデータベースの型に合わせるための変換がほとんど必要でないという利点がある。また、オブジェクト指向データベースはクラス階層を持っており、これによってデータの整理が容易になり、上位クラスから下位クラスへのデータの継承は物理的な節約になるという利点もある。



リレーショナル・データベース



オブジェクト指向データベース

図4. 2-1 オブジェクト指向データベースとリレーショナル・データベース

ヴァーチャル・エンジニアリング・システムのような景観データや形状データといった画像データや、評価コードや実験データといった各種形式の文書・数値データを入り組んだ方法で使用する場合に、一般ビジネスデータに比較しデータ構造が複雑であり、データサイズが多様である。その複雑なデータをいかにデータベースに格納するかが大きな問題であった。本検討では、現在データベースの標準であるリレーショナル・データベースと、まもなく主流となるであろうオブジェクト指向データベースを本システムで扱うデータの適用性について行った。

#### リレーショナル・データベースの特徴

リレーショナル・データベースは言語としてSQL (Structured english Query Language) を備え、このSQLはANSI規格が定める事実上のデータベースの標準言語としての地位を獲得している。リレーショナル・データベースに格納する情報の構造はシンプルな関係を持った表 (テーブル) でまとめられ容易に構築が可能である。主な特徴は次のとおりである。

- ・内部情報構造は簡単にいえば表の集まりである。
- ・データを2次元の表で表し、表と表の間に相関を持たせることができる。
- ・ファイルを用いて管理する場合と異なり、データの構造や変数の型などを意識しなくてよい。
- ・複数人間がランダムにアクセスしてもデータの整合性がとれる。

- ・豊富な検索機能
- ・SQLでデータベースアクセス
- ・表の幅（列：フィールド）が変更されるような使用は苦手で、列の項目が確定されているものに対しては便利。
- ・SQLは規格化、標準化されており、どのデータベースソフトでも基本的には共通であり、個々のデータベースソフトによる拡張性もある。

#### 4. 3 データベース管理システム機能仕様

データベース管理システム機能は、本システムで取り扱うシナリオや形状モデル、試験データやコード評価データといった数値、文字およびイメージデータ等をデータベース化し、対話システムからの要求に対し的確な情報を回答するものである。データベース管理システム機能はデータベースシステム（DBMS）そのものが持つ機能と対話システム等からの要求に対するデータベースへのアクションやデータベース間を取り持つ機能に大別することができる。いずれにしろ、各データはデータの用途に合わせて何らかのデータベースモデル形式としてデータベース化する必要がある。前項の検討から本システムで取り扱う情報のデータベース化はおよそ次のとおりとした。

##### 文字、数値データ

試験データは試験方法の変更や測定スケジュールで追加更新され、評価コードはアップデートにより履歴更新されるデータであり、随時更新される性格のものである。また、他の異種な情報とのカップリングを同データベース内で求めるものではなく、同形式データとのリレーショナルな方法にて収集し取り扱われること、定型フォーマットによる処理が有用であると考えられた。そのことから、試験データや評価コードデータなどの数値データや文字、文書データは表形式で扱うリレーショナル・データベースが適当と思われた。

##### シミュレーション・シナリオ、シミュレーション・モデルデータ

シナリオを構成する個々のデータ（個別リスト）は地層処分システムのガラス固化体やオーバーパックといった場単位や場に関連する事象・特性により作成されている。システムから要求する情報は、場にかかわる事象や特性シナリオであり、これはオブジェクト指向データベースというクラスという単位でまとめることが適当と思われる。シミュレーション・モデルデータは地層処分システムにかかわる形状データや景観データが主なデータであり、形状データや景観データには形状イメージデータと形状仕様データをカップリングさせることが有用と思われる。また形状データは人工バリアをオブジェクト指向データベースの地層処分システムのサブクラスと考えた場合、個々のオブジェクトになりうる。それらから、シミュレーション・シナリオデータとシミュレーション・モデルデータはオブジェクト指向データベースが適当と思われた。

本システムのデータベース管理システム機能においては前項でも記述したとおり、リレーショナル・データベースやオブジェクト指向データベースといったデータベースシステムそのものに依存するものではなく、システム機能としてこれらのデータベースシステムの特徴を有効に利用するものと考えた。データベースはデータの形式および取り扱いに適当と思われる格納（エンジンとして）のための手段であり、それらからユーザー（対話システム等）の要求する情報に対し、グラフィクス技術やインターフェイス技術を駆使したデータベース管理システム機能を用いてアプリケーションを作成し分かりやすく表現するものである。データベース管理システム機能の検討は、試験データや評価コードデータ、シミュレーション・シナリオやモデルデータを格納するそれぞれのデータ



ベース（マルチデータベース）を、ユーザーの要求する情報に対して、ネットワークやGUIおよびエージェント指向概念等の技術を利用して一元的に行えることに着目し行った。

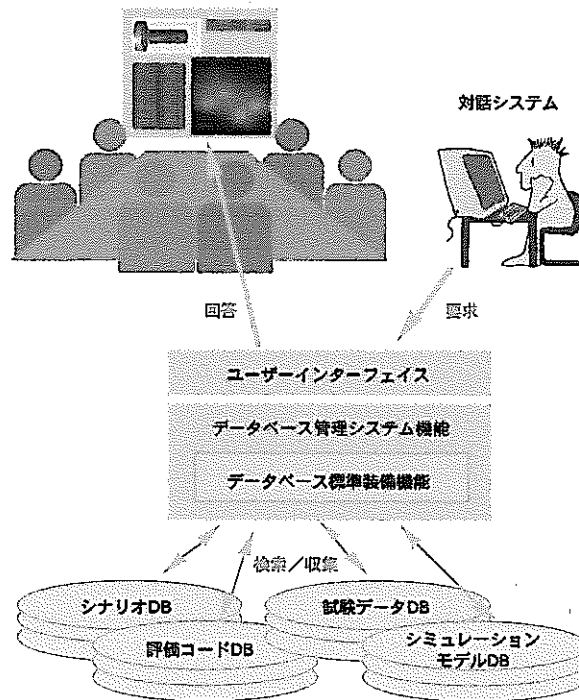


図 4. 3 - 1 データベース標準装備機能とシステム機能

#### データベース標準装備機能

データベースシステム標準装備機能は、データ格納エンジンとなるデータベース管理システム（DBMS）が標準装備している機能である。リレーショナル・データベース管理システム（RDBMS）はベンダーにより各種存在するが、テーブル作成や削除およびアクセス等のデータベース管理機能はほぼ標準化されている。オブジェクト指向データベース管理システム（OODBMS）は近年市場に出てきたプロダクトであり、データベース管理機能として標準化が進められているところである。データベースへのアクセス操作や保守機能等についてプロダクトにより扱い方に多少の相違があるが、およその基本機能は次のとおりである。

- ・データベース作成、削除
- ・データベースアクセス（対話型SQL、埋込型SQL）
- ・レポートライター（DB内容の編集出力）
- ・分散構築環境（サーバー・クライアント）
- ・HTML対応
- ・マルチメディアデータ対応

DBMSの基本機能はデータベース構築、保守が中心であり、個々のシステム構築のためには別途CやC++、4GLといった言語を用いて、埋込型SQLやレポートライター等のデータベース基本機能ツールで開発を行わなければならない。

## データベース管理システム機能

データベース管理システム機能は、データベース標準装備機能やグラフィックツールおよび開発言語を用いて開発を行う。ユーザー（対話システム等）から要求される内容に対して、個々のデータベースに格納されているデータから要求に合致した情報を取得し、取りまとめユーザーインターフェイスに渡し、要求側に回答することがデータベース管理システム機能の主たる機能とした。

### 4. 4 オブジェクト指向技術の応用

オブジェクト指向では、今の比喩でいう物事の実行主体のことをオブジェクトと呼んでいる。複数のオブジェクトが、相互作用しながら問題解決を行うような形にソフトウェアの構造を作り上げていくのがオブジェクト指向の作法といえる。各オブジェクトは、その領域において1つ1つ明確に区別された主体として、シナリオ・オブジェクトや評価コード・オブジェクトといった具合に識別する。そして各オブジェクトには、それぞれの領域の分担を割り当てる。言い換えると、この責任分担がそのオブジェクトの持つべき知識や能力を表している。オブジェクトは自分自身の責任担当の動作を実施するのに必要な知識や情報（データ）と、処理に必要なルール（アルゴリズムや手順）を内部に持ち、オブジェクトは自身が保持しているこれらの知識やルールを使ってできる範囲のことを実行する。また、オブジェクトは自分の責任範囲で手に負えないことは、他のオブジェクトに依頼して実行してもらうが、この依頼はメッセージを送信することによって行う。メッセージは、依頼したい仕事の種類とその仕事の実施に必要な情報（メッセージ・パラメータ）からなる。すなわち、オブジェクトは仕事の一部を責任もって遂行できる実行主体ということが出来る一種の擬人化といえる。オブジェクトは、仕事の性格に応じて3つに分類して考えることができる。ユーザー（対話システム）とのやり取りに責任を持つインターフェイス・オブジェクト、処理の割り振りを行ったり、複数のオブジェクトからの情報をもとに処理を実行したりするコントロール・オブジェクト、問題領域に存在する物や知識を資源として保持管理するためのドメイン・オブジェクトである。これらオブジェクトは、マルチプラットフォームで相互との関係を見ながらタスク処理し、あるひとつの目的に対して最終的に統合され、結果を生み出す場合にエージェントと呼ぶこともできる。ヴァーチャル・エンジニアリング・システムでは、これらのオブジェクトのカテゴリーは重複して実現されることも多いので、オブジェクト指向でモデル化する際には、その重複を解きほぐすことはシステム分析において重要な考慮点である。

#### リレーショナル・データベースとオブジェクト指向データベースの統合化手法

前述したとおり、オブジェクト指向はデータベースに依存するものではなく、アプリケーション開発における考え方であり、オブジェクト指向に基づきシステムを作り上げていくことによって、効率的な処理やユーザーライクな対話環境を構築することができる。リレーショナル・データベースとオブジェクト指向データベースは、それぞれ特徴を持つが特定のデータベースに依存した形でシステムを構築することは、柔軟なシステムになり得ないと考えられる。そこで、それぞれのデータベースの特徴を基に、オブジェクト指向技術概念を基にデータベースの有効利用としてリレーショナル・データベースとオブジェクト指向データベースの統合について検討を行った。

フリーソフトウェア（パブリックドメインとも呼ばれる）により、一般的にオブジェクトSQLと呼ばれるSQLライクな言語（SQLの上位互換ではなく標準でもない）を取得することができる。

これを従来の検索機能がなかったり、単純な検索機能しか持たないオブジェクト指向データベースに付け加えることで、オブジェクト指向データベースの定義と検索を行うことが出来る。このオブジェクトSQLは、オブジェクト指向データベースから単にリレーショナル・データベースへのゲートウェイ的な機能を提供するものである。

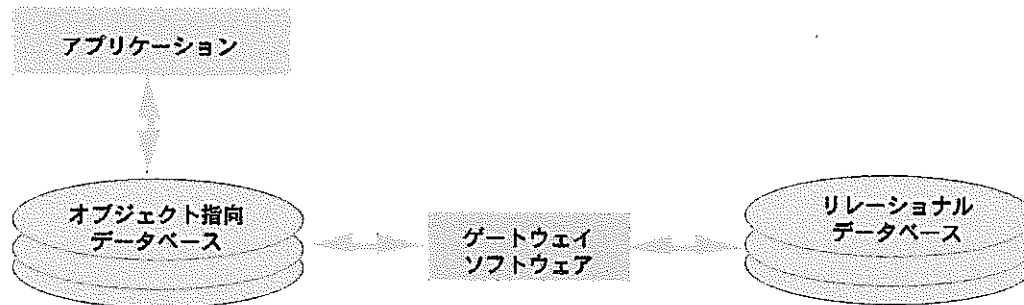


図4. 4-1 オブジェクト指向データベースとリレーショナル・データベース間ゲートウェイによる統合

図4. 4-1 においては、アプリケーションはオブジェクト指向データベースを考慮し作成される。ゲートウェイを使う方法では、処理中にオブジェクト指向データベースの要求がリレーショナル・データベースへ送られ、リレーショナル・データベースからの結果が要求元へ返される。ゲートウェイは、リレーショナル・データベースにとっては普通のデータベースユーザーのように見える。ゲートウェイの実現方法では、オブジェクト指向データベースへの要求に様々な制限がある。読み出ししか出来なかったり、複数の要求を一つのトランザクションとして扱うのではなく一つの要求しか実行できなかったり、リレーショナル・データベースが処理できる様々なクエリのうち簡単なものしか実行できなかったりという制限である。

ゲートウェイによる方法ではアプリケーションプログラムは、オブジェクト指向データベースから取り出したデータもリレーショナル・データベースから取り出したデータも使うことが出来るが、要求や返ってきたデータの変換やリレーショナル・データベースとの通信オーバーヘッドのため、その速度といった性能に限りがある。

図4. 4-2 は、オブジェクト指向プログラミングによるリレーショナル・データベース統合を示したものである。

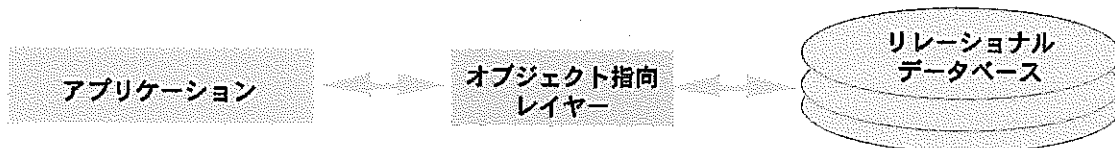


図4. 4-2 オブジェクト指向レイヤーによるリレーショナル・データベースの統合

図4. 4-2 に示すオブジェクト指向レイヤーによる方法では、アプリケーションはオブジェクト指向データベース言語を使用し作成する。オブジェクト指向レイヤーは、データベース言語

のオブジェクト指向的な部分をその下にあるリレーショナル・データベースが扱えるように等価でリレーショナルな表現に変換する。例えば、オブジェクト指向レイヤーは、オブジェクトとリレーションのタプルを対応づけ、オブジェクトのOIDを生成し、それらをリレーショナル・データベースのインターフェイスを使ってリレーショナル・データベースのタプル（レコード）の属性にする。再びリレーショナル・データベース・インターフェイスを使って、オブジェクト指向レイヤーは、オブジェクト中のOIDをリレーショナル・データベース中のオブジェクトと対応する。

リレーショナル・データベースは、データ管理レイヤーと格納管理レイヤーの2つのレイヤーから構成されている。データ管理レイヤーはSQL文を実行し、格納管理レイヤーはデータをデータベースへ関連づける機能を持つ。オブジェクト指向レイヤーは、SQL文を使ってリレーショナル・データベースへアクセスするためにデータ管理レイヤーと、低レベルの手続き呼び出しを使ってリレーショナル・データベースへアクセスするために格納管理レイヤーと連絡をとる。OpenODBは、オブジェクト指向レイヤーとリレーショナル・データベース間の通信にデータ管理インターフェイスを使用する。

この方法は、オブジェクト指向レイヤーに適合させるような修正をリレーショナル・データベースへ加えないことを仮定していて、洗練されたデータベース機能が必要となった時に性能面や操作性の面で大きな問題が起こることがある。例えば（動的スキーマ変更で）クラス階層の中で多数のクラスをロックする必要がある場合オブジェクト指向レイヤーは、

- ・リレーショナル・データベースは、クラス階層を単位としてロックかける機能がないので、いくつかのロックを一度にかけにくい。性能上の問題とデッドロックの危険を招く。
- ・データベース全体に対してロックをかける。他のユーザーは全くアクセス出来なくなる。

のどちらかを選択しなければならない。さらに、オブジェクト指向レイヤーがメモリ中のオブジェクトの更新をサポートし、アプリケーションのトランザクションが終了する時に自動的にデータベースへフラッシュする場合、リレーショナル・データベースインターフェイスを使って個々のオブジェクトを一度にデータベースへ戻さなければならない。

オブジェクト指向レイヤーによる方法の合理的な点は、既存の様々なリレーショナル・データベースに対して移植できることである。このフレキシビリティは多少の性能を犠牲にすることがあるが、オブジェクト指向レイヤーによる方法は様々なデータベースをアプリケーションに対して単一のデータベースに見せることができ、このデータベースシステムは「マルチデータベースシステム」構築の手段としてヴァーチャル・エンジニアリング・システムのデータベースとして検討することとした。オブジェクト指向レイヤーによる方法は、オブジェクト指向データベースやリレーショナル・データベースからデータを取り出すアプリケーションプログラムを可能にするマルチデータベースシステムの基礎となると思われる。

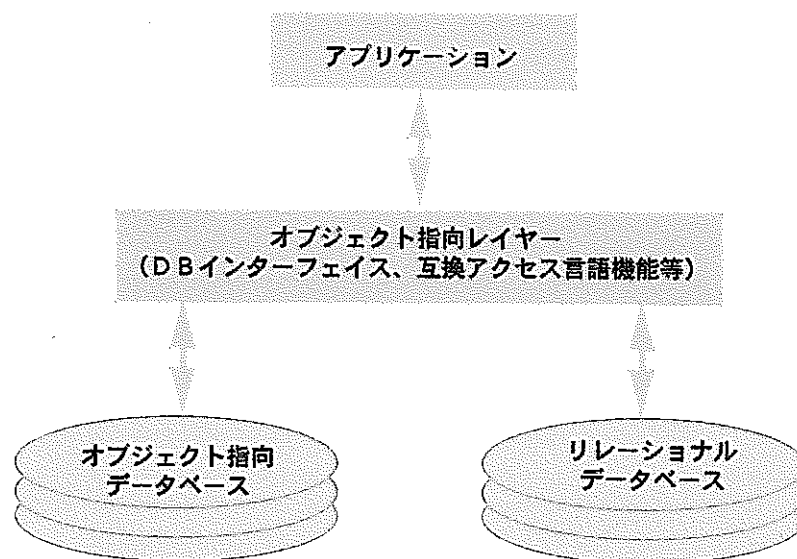


図4. 4-3 データベース統合アプローチ

統合化するためのオブジェクト・リレーショナルの方法は、オブジェクト指向レイヤーとリレーショナル・データベースを一つのレイヤーにまとめるために、リレーショナル・データベースの格納管理とデータ管理に必要な次の検討を行った。

- ・リレーショナルとオブジェクト指向のデータモデルを一つの統合化されたモデルにまとめる。
- ・ANSI SQLデータベース言語の上位互換言語の検討。
- ・SQLのデータの定義、検索（ジョインや集合操作など）および更新機能。
- ・データベースシステムのデータベース言語で提供される機能のサポート。
- ・動的スキーマ変更、クエリの自動最適化、クエリの自動処理、アクセス方法（B+ツリー、拡張ハッシュ、ソートなど）、同時実行制御、ソフトウェア障害やハードウェア障害からのリカバリ、トランザクション管理、アクセス権の付与と取消などのサポート。
- ・リレーショナル・データベースと同じ操作はリレーショナル・データベースと同じ性能で、オブジェクト指向データベースと同じ操作はオブジェクト指向データベースと同じ性能で実行する必要がある。

オブジェクト・リレーショナル・データモデルによって複雑なデータを簡単で自然に表現することができることから、リレーショナル・データベースにオブジェクト指向アプローチを取り入れ、データとその属性値をファイルを処理するプログラムをカプセル化してオブジェクトとし、データを管理するヴァーチャル・エンジニアリング・システムのデータベースを検討した。多くのオブジェクトモデルでは属性値の集合をインスタンスとし、その属性値を操作するものがメソッドとしている。以上より、本システムで検討したオブジェクトモデルは図4. 4-4に示すよう

にインスタンスは属性値の集合とデータで構成した。

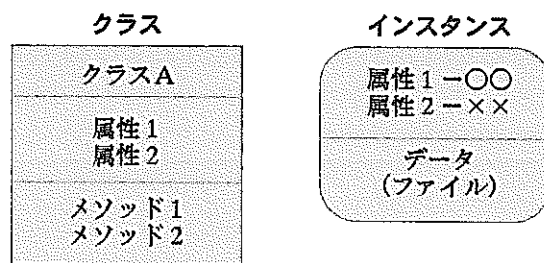


図4. 4-4 オブジェクトモデル

上図の方法によって、従来使用していたデータとプログラムをそのままの形でデータベース内に格納できる。図4. 4-5にクラス階層概念を示す。

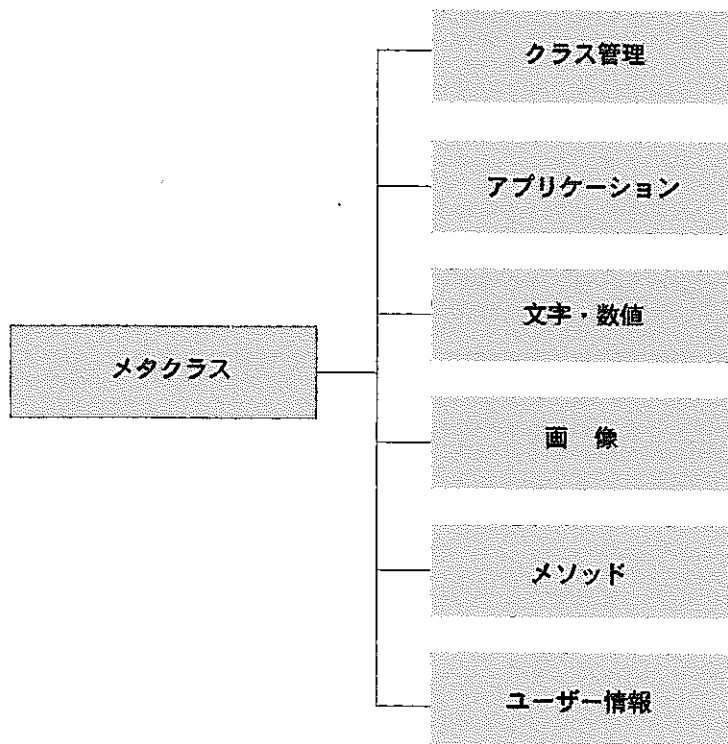


図4. 4-5 オブジェクト空間図

図4. 4-5において、各クラスのクラス、属性およびメソッドは次のとおりである。

**メタクラス** : 全てのクラスの基となるメタクラスで、ルートクラスにはデータの正しさを設定するインテグリティ属性、機密保護の設定をするセキュリティ属性およびインスタンスの識別を行うインスタンス識別属性が付属する。

**クラス管理** : 各クラスをインスタンスに持つ特別なクラス (クラスのクラス) で、クラス管

理クラスにはメソッドとしてオブジェクト生成、削除、実行等の各コマンドが付属する。

アプリケーション：アプリケーションごとに1つのインスタンスを持つようなクラス。

文字、数値：オブジェクトを生成する際の文字、数値データのメタクラス。

画像：オブジェクトを生成する際の画像データのメタクラス。

メソッド：他のクラスのメソッドをインスタンスとして持つようなメタクラス。

ユーザー情報：このクラスでデータベース操作管理を行う。

#### 4.5 シミュレーション・シナリオ・データベース試作

シミュレーション・シナリオ・データベースは、地層処分性能評価のためのシミュレーション・シナリオを格納するデータベースとした。対象とする評価シナリオは、別途開発して地層処分FEP情報データベース・システムからの出力を利用することとした。

本システムにいうシミュレーション・シナリオは、評価シナリオから組み立てたクラス階層としている。クラス階層にする理由は、文章の形式で記述された評価シナリオに沿って評価シミュレーションを実行する上で、それを実行可能な情報構造に変換する必要がある、そのための適切な手法として、評価シナリオをクラス階層化することが考えられるからである。この手法により、メンバーとしてシナリオにモデルの取り込み(Encapsulate)を可能にすることができると考えられ、評価にあたっては、対応するモデルとコードを自然に関係づけ、起動できるようなアルゴリズムを開発できると予想されるからである。このため、スクリーニング済みのシナリオ(FEP、インフルエンス・ダイアグラム)から、等価な内容に沿って評価シミュレーションを記述するクラス階層データベースを試作研究した。

地層処分FEP情報データベース・システムは、個別FEP情報を基盤とした因果関係情報やインフルエンス・ダイアグラム情報等のFEP情報で構成されている。個別FEP情報は、FEPの同定による現象をテキスト・ファイル化したものであり、シナリオ作成はこのリストを現象で分類し、FEPの組み合わせによって体系づけしている。

##### B-4.1 緩衝材中の空隙水の化学

###### 1. FEPに関する記述とその評価

###### 1) 内容【当該FEPの説明】

ベントナイト空隙水化学は地下水の飽和に伴うベントナイトと地下水との化学反応によって変化していく。この際、空隙水化学は地下水の水質、処分場に侵入した空気や微生物、ガラス固化体からの熱や地熱、ベントナイトの物理化学的性質、ベントナイトの膨潤、溶質の拡散の影響を受ける。さらにオーバーパックの腐食に伴う腐食生成物や水、ガスの影響も考えられる。オーバーパック破損後は、ガラス固化体から溶けだした核種の崩壊による放射線分解の影響も想定される。また岩盤/ベントナイト境界部で起こるベントナイト・コロイドの散逸現象への空隙水化学の影響も考えられる。一方、空隙水化学はオーバーパックの腐食速度を決定し、ガラス固化体から核種の溶解およびベントナイト中の物質移行に影響を与えることになる。またベントナイトのイオン交換特性はベントナイト中の核種移行に影響を与える。

###### 2) 原因【当該FEPに影響を及ぼす因果関係リスト、その影響の記述と評価上の扱い】

##### B-2.1 緩衝材の冠水→B-4.1 緩衝材中の空隙水の化学

###### (当該FEPへの影響)

ベントナイトの冠水に伴い、空隙水化学はベントナイトと地下水との化学反応によって変化していく(油井ら, 1992)。

…… (以下、省略)

図4.5-1 個別FEPリスト例

個別FEP情報は、地層処分研究に携わる研究者の解析や実験による研究報告として個別FEP情報形式（フォーマット）でまとめられている。

1. FEPに関する記述とその評価
  - (1) FEPの記述 ⇒FEPそのものの内容の記述
  - (2) 因果関係リスト（原因） ⇒当該FEPへの影響の記述
  - (3) 因果関係リスト（結果） ⇒当該FEPが及ぼす影響の記述
  - (4) 評価上の取扱い ⇒当該FEPに対し、H3報告書での評価内容とそれ以降の評価とに大別して記述
2. モデル化に関する記述 ⇒概念モデル、基礎理論、数学的定式化（境界条件を含む）パラメータの種類
3. 文献リスト ⇒1.、2.で根拠となる文献リスト
4. 諸外国での取扱い ⇒上記取扱いと異なる点に重点を置いた記述

図4. 5-2 個別FEP情報記述形式

地層処分FEP情報データベースでは、蓄積格納された個別FEP情報を用いて専門分野間のコミュニケーションやシナリオ研究の進展による変更等に利用するために、様々な視点から観覧できるように上図の個別FEP情報を構造化してデータベースとしている。本システムのシミュレーション・シナリオ・データベースでは、シナリオを評価するためのシミュレーション実行として利用可能とするために必要なデータベースを検討した。

前述のとおりシミュレーション・シナリオ・データベースは、本システムを構成するマルチ・データベースの一部であり、シミュレーション・シナリオ・データベースの動作概念を次のとおりとし、検討を行った。

- ・対話機能によりシナリオ条件を設定できること。
- ・設定（シナリオ選定）されたシナリオをクラス階層化構築できること。
- ・シナリオ評価のためにオブジェクト・データベースから情報を受け取れること。
- ・シナリオ評価に必要な評価コード情報を受け取れること。
- ・上記で取得したデータ（情報）を要求元で渡せること。

以上の条件を満足するためのデータベース構成要素として、個々の連携を含む個別FEPリストと設定シナリオに基づく解析コードおよび試験データ等のデータベース間情報受け渡しが必要と考えられる。シナリオ中心の処理においては、シミュレーション・シナリオ・データベースから得るシナリオデータをオブジェクト空間から文字・数値に置き、シナリオを構成するにあたり必要な画像および評価コード情報を合合できるものとした。



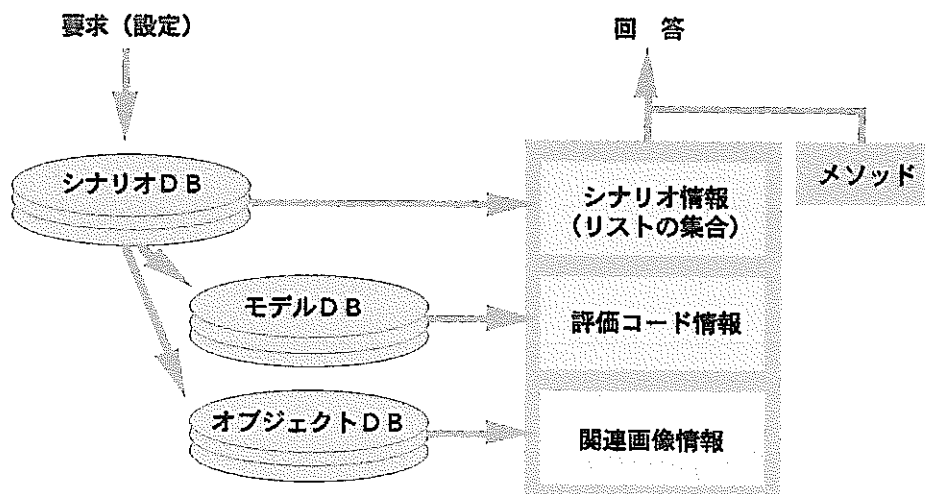


図 4. 5-3 シナリオデータベースの動作概念

シナリオの設定においては、要求先からの必要な情報を各データベースから取得しインスタンスとして格納する。設定終了後、メソッドを加えカプセル化し、オブジェクトとして生成し、アプリケーションからの要求を待つ。なお、図 4. 5-3 でメソッドはデータベースから得られた情報の表示装置への出力やシミュレーション・エンジンへのメッセージパスである。

```

Attributes:
  feyno          character(16,1) Not Null
  des            ntext(80,256,256,5)
  uname         integer Not Null
  date          time(11)
  checkok       integer
3* display table CE;
*** Table: CE ***

Attributes:
  cau           character(16,1) Not Null
  eff          character(16,1) Not Null
  rel_ce       character(16,1) Not Null
  inf_ce       ntext(80,256,256,5)
  ce_sig       integer
  uname       integer Not Null
  date        time(11)
  checkok     integer
4* display table EVA;
*** Table: EVA ***
  
```

図 4. 5-4 試作したシミュレーション・シナリオデータベース・テーブル

#### 4. 6 シミュレーション・モデル・データベース試作

シミュレーション・モデル・データベースは、シミュレーション・シナリオに接続 (Encapsulate) するモデル (評価コード) のデータベースとして設計を行った。シナリオとモデルを接続するためには、モデルとその関連情報の整理、シナリオとモデルとの接続方法に関する検討が必要となる。前者については、海外にて実施されたコード比較、モデル別コード情報整理の経験などを参考に、本システムにて利用するために適当な形式の情報構造を検討した。

なお、本項にて触れるクラス設計とは、オブジェクト指向データベースのデータ構造、もしくは、オブジェクト指向プログラミング言語の参照となるプライベートなクラス・ライブラリーを設計することを意味する。シナリオとモデルとの接続方法については、モデルを反映する評価コードが FORTRAN 言語で開発された既存のものであることを考慮し、いくつか考えられる手法を試作検討を行った。例えば、クラス化されたシミュレーション・シナリオの評価記述に対応するモデルが FORTRAN で書かれたコードである場合においては、当該クラスのメンバー関数は、同 FORTRAN コードをアクティブにするための UNIX Shell を起動する関数として Encapsulate する手法が考えられる。ただし、この場合においても FORTRAN コードが、シミュレーション・エンジンにメッセージを送り込むアルゴリズムの組み込みが必要である (MPI、ソケット等)。この手法の利点は、モデルを反映するコードそのものを書き換える必要がないため、再検証等の手間がかからないことが挙げられる。

また、将来の完全なオブジェクト指向によるモデル記述の可能性を検討するための予備的研究として、スクリーニングが施され設定された評価シナリオ (例えば、地下水による物質移動評価シナリオ) について、インフルエンス・ダイアグラム等を参考に、オブジェクト・シミュレーションのためのモデル・クラス設計を検討した。これはモデルに対するクラス化のアプローチとして、従来のモデル開発とは異なる手法として評価を行った。

```
*** Table: code ***
Attributes:
  c_name                character(30,1)
  c_summary             text(80,32,32,1)
  c_machine             text(80,32,32,1)
  c_method              text(80,32,32,1)
  c_restriction         text(80,32,32,1)
  c_run_time            text(80,32,32,1)
  c_unusual_feature     text(80,32,32,1)
  c_aux_program         text(80,32,32,1)
  c_status              text(80,32,32,1)
  c_reference           text(80,64,64,1)
  c_machine_requirement text(80,32,32,1)
  c_language            text(80,32,32,1)
  c_os                  text(80,32,32,1)
  c_other_limitation   text(80,32,32,1)
  c_auther              text(80,32,32,1)
  c_material            text(80,32,32,1)
  c_keywords            text(80,32,32,1)
  c_field               character(30,1)
  c_category            text(80,32,32,1)
*
```

図 4. 6-1 シミュレーション・モデル・データベース (テーブル)

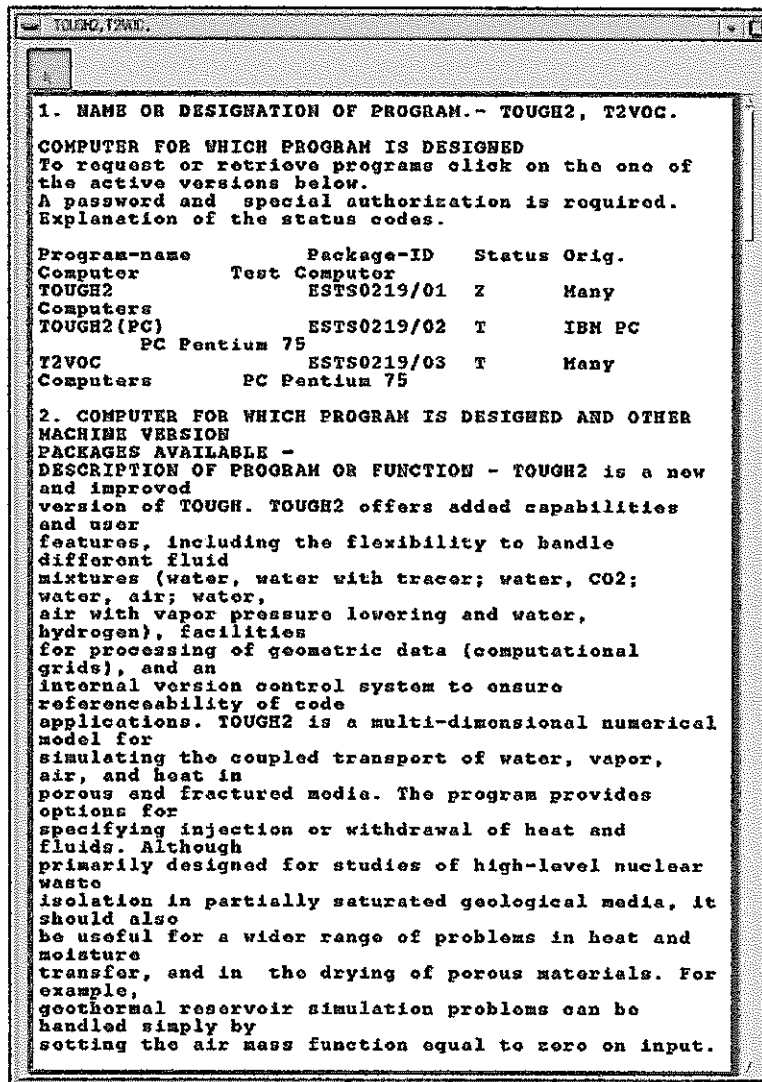


図 4. 6-2 モデル情報の表示

図 4. 6-1 はモデル別コード情報のデータベースを示したものである。モデル別コード情報はシナリオ・データベースから呼び出され、シナリオ情報に接続されシミュレーション・エンジンに渡される。図 4. 6-2 にデータベースからモデル情報を表示した例を示す。

また、モデル情報に付属して、シミュレーション・シナリオの評価記述に対応するモデルが FORTRAN で書かれたコードと想定し、データベース化を検討した。シミュレーション・シナリオ・データベースで記述したように、シナリオは設定時に対応するモデル情報や画像情報をカップリングさせメソッド（データの扱い）とともにオブジェクト化される。ここで、モデル情報を取得する際にオブジェクト空間図に示したアプリケーション・クラスから該当のコード・データを取得しシミュレーション・エンジンに渡すことで処理可能となる。図 4. 6-3 に試作したコードデータ管理画面を示す。

Code Name	Category	Field
001F3	Groundwater Flow	Far Field
1000H1, 1210G	Groundwater Flow, Thermal Process, Gas Transport Near Field, Near Well	
1000H1A, 011	Groundwater Flow	Far Field, Near Field
1000H1B	Groundwater Flow	Far Field, Near Field
001F1B	Groundwater Flow	Far Field

図 4. 6 - 3 コードデータ管理画面

#### 4. 7 シミュレーション・オブジェクト・データベース試作

シミュレーション・オブジェクトは、処分システムを構成する物理的な機器、構造物を3次元形状データとして記述し、グラフィック・ディスプレイ・バファ内に、性能評価シミュレーションにおける物理量表現のための空間を提供する目的で利用するクラス化されたポリゴナル情報とした。この情報は、対話システムを通じ、最も頻繁に操作者が触れる情報となる。この理由は、ユーザーが視覚的に一連の性能評価操作を行う機能を提供することを本システムの大きな目標としているからである。このため、将来的には3次元のオブジェクト空間を利用し、3次元でのデータベースにアクセスする手法等の研究も想定することが可能となる。これらの一連の3次元オブジェクトを利用したコンピュータへのアクセスは、本システム開発の基本的な課題となる。

オブジェクト（画像等）データは形状のデータとその部品情報としてカプセル化や部品データ集合による形状オブジェクト化等考えられる。画像等のデータは以上のオブジェクト生成のためにオブジェクト空間における画像クラスとして管理するものとした。

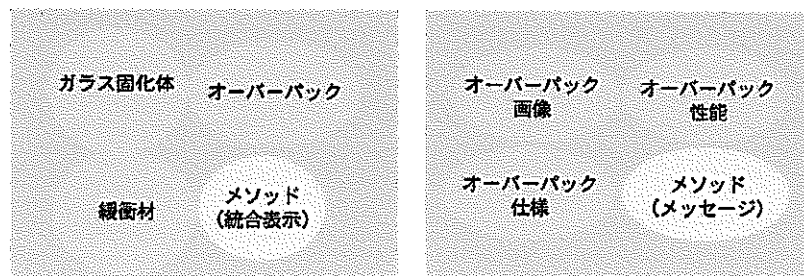


図4. 7-1 シミュレーション・オブジェクトのオブジェクト化

シミュレーション・オブジェクトは本システムの情報の基盤となり、様々な方法や形でオブジェクト化され利用する。シミュレーション・オブジェクト・データベースに格納するオブジェクトデータ（画像等）は、メソッドや動作させるアプリケーションにより再利用可能となるようデータベースに格納するものとした。

図4. 7-2 はオーバーバックの仕様を表示装置に出力した例である。

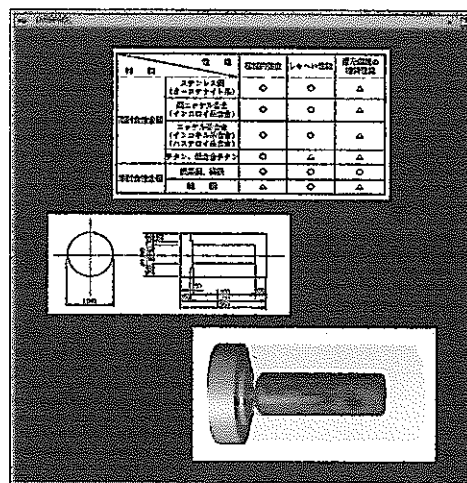


図4. 7-2 オーバーバック仕様オブジェクト表示例

#### 4. 8 試験データ等、数値・文字データベース試作

試験データ、文献データ等は、数値、文字からなる情報であり、一般的に研究開発における試行錯誤的な検索を条件とする要求については、表形式のリレーショナル・データベースが使いやすいと考えられる。例えば、元素とその化合物水溶液中における化学的平衡論の問題は、地下水の地球化学的特性を評価する上で重要であることはよく知られている。この問題について考えた場合、大別して2つの課題がデータベースに関連するものとして考えられる。第1に化学的平衡反応の系内で、その反応に関係するあらゆる化学種すべてを取り上げ、その平衡状態における存在比を確定しデータベースとする作業である。第2に平衡状態に達するまでの所要時間（反応速度）を解明（モデル開発）することである。これらの問題は、地球化学コード等の熱力学データ・ベース、モデル式等を整備する問題として従来より諸外国において研究が行われており、リレーショナル・データベースを応用した研究の代表的なものである。

本システムの、試験データ等、数値・文字データベースは、個別の解析コード向けデータベース（例えば、PHREEQETDB等）とは別に、各国にて研究報告されている熱力学データ、収着データ、腐食データ等の試験データや、フィールド・テストデータを文献と連結して管理し、シミュレーション結果の比較や、試験データ間の比較を対話方式で行う機能の開発について予備的な検討を行うために試作するデータベースとした。従って、基本的な作図、作表機能等もあわせて試作評価した。

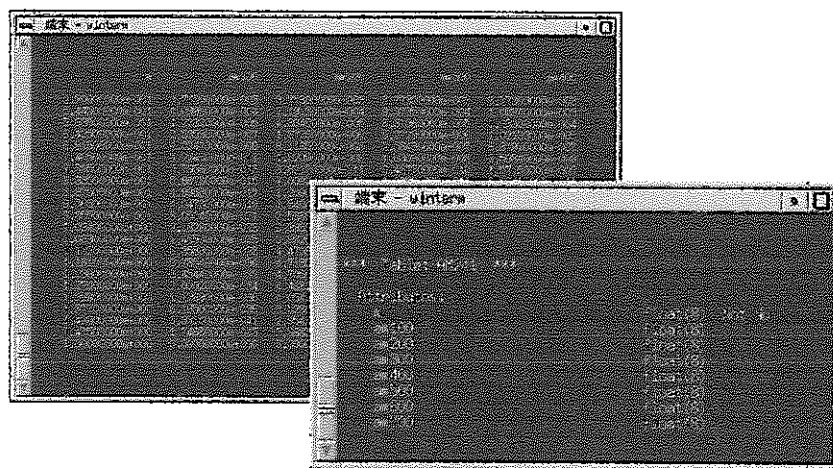


図 4. 5-1 数値データ・データベース

#### 4. 9 品質管理データベース試作

品質管理データベースとして管理する情報の種類は、大別して、プログラム品質管理情報、数値データ品質管理情報、文書情報品質管理情報があげられる。これらの情報は、そのほとんどが文字データより構成されるものであり、フィールドとして与えられた項目に沿って情報が多数存在するため表形式のリレーショナル・データベースが使いやすいものと考えられる。本年度は、プログラム品質管理情報として、ソースプログラムや複雑度等を管理する手法、数値データ品質管理情報として熱力学データ等の信頼性を管理する手法、文書情報品質管理情報として技術文献等のトレーサビリティについて管理する方法等を試作研究した。

## 5 ソフトウェア開発管理機能の概念設計

前年度の研究の視点と成果の概要は下記のとおりである。

### ソフトウェア・エンジニアリング環境の基礎的検討

検討の結果、本システムの対話機能、シミュレーション機能、データベース機能等を実現するためのソフトウェア開発においては、開発するプログラムの品質を保証するための論拠となる手法を合わせて開発することが必要であると思われる。

### ソフトウェア・マネジメント・モデルの基礎的検討

検討の結果、将来の本格的なシミュレーションにとって重要な熱力学データ、収着・拡散データ等の試験データについては、データの抽出や選定における試行錯誤的な検討を支援する機能が、データ管理上重要な位置を占めると思われた。

### ソフトウェア・ライフサイクル管理手法の基礎的検討

検討の結果、モデルとコードの開発作業のライフサイクルを管理することは、本システムのシミュレーション結果の検証作業にとって重要であると思われた。

以上のことから本年度は、ソフトウェア開発管理機能の概念設計を中心として以下の通りの関連する技術研究を実施した。

### ソフトウェア開発管理機能概念設計

本システムを設計開発するための情報について、管理（履歴管理、品質管理）統制を行うサブシステム機能についての概念設計を行った。なお、ソフトウェア開発管理機能は、本システムのために開発するソフトウェアの管理手法と機能、試験データ等の数値（文字）データの品質管理手法と機能、技術文献等の関連性管理の3種類の機能セグメントに大別して設計検討した。

### プログラム品質管理機能試作

プログラム品質管理機能の概念設計を行うため、プログラム品質の定量的評価手法等について研究した。定量化手法については複雑度解析が従来より知られているが、この手法は、C、Fortran言語等の、手続き型言語で記述された実行手続き部分の分岐の数量と、実行経路の組み合わせ等から判断するアルゴリズムを基本とした手法であるため、本システムのプログラムの主要な部分を構成するオブジェクト指向プログラミングにおいて妥当であるか検討を行う必要があると考えられた。

### 数値データ品質管理機能試作

数値データの品質管理については、データが試験装置により計測されたものを想定し、データの管理と簡易解析を行う機能を研究した。例えば、誤差範囲を示すフィールドを設けること、散布図等によりデータを一括して視認できる機能等である。また、前年度に予備的に検討した対話方式によるヒストグラム処理を応用したインフルエンシブ評価機能についてその適用可能性を検討した。

### 文書情報品質管理機能試作

本プロジェクトに関連する文書情報や海外から入手した文書情報等について、その品質を管理する手法を

研究した。また、本プロジェクトに関する情報品質の管理についてはISO9000（内部品質監査）参考として手法を研究した。

## 5. 1 ソフトウェア開発管理機能概念設計

地層処分ヴァーチャル・エンジニアリング・システムの主要な構成要素であるソフトウェアは、従来の個別解析コードのような単体ソフトウェアと異なり、多くの種類のソフトウェアを統合するため、大規模化・複雑化・多様化する。現代のXウィンドウを実装したUNIXシステムでは、ソフトウェアの規模をステップ数で見積もることは、ほとんど意味をなさないと考えられる。例えば、Xウィンドウ・システムだけでも、100万ステップ以上のステートメントで構成されていると言われていた。従って、本年度に改良整備したシステム・ソフトウェアをステップ数で見積もれば、おそらく数100万ステップに及ぶと予想される。

このような背景から、所定の生産性を確保しながら、計画する要求性能（品質）を満足するソフトウェアを開発できるように、明確な開発計画の下で開発プロセスの全工程を系統づけて管理する必要がある。開発管理の対象となる主要な要素は、品質、予算、作業期間である。したがって、精度の高いソフトウェアの開発規模と工数の見積りを行って、開発資源の適正化を図りながら、各管理対象に対する品質管理、予算管理、および工程管理を効果的に推進するための手法とシステム化が必要である。そのため、本年度は目標システムのソフトウェア開発管理機能の要求を整理し概念設計を行った。

ソフトウェア開発管理機能の設計にあたり、その基本的考え方とシステム化要件を明確にする必要がある。そのため、基本となるソフトウェア開発プロジェクトを推進するための開発技術と、開発管理の基本的な考え方について概要を報告する。機能設計要件として、ソフトウェア開発プロセスで作り込まれる品質特性と品質管理要件について概説し、システム化の基本手法となる定量的品質管理モデル（ソフトウェアメトリクス）に基づくソフトウェア品質の計量化についても補足する。また、それらの設計要件を評価するため簡単な複雑度解析機能を試作した。設計検討の結果、大規模なソフトウェアの開発・統合化プロジェクトには、ソフトウェアの定量的な品質管理手法とそれに基づく解析評価機能が必要である。

### ヴァーチャル・エンジニアリング・システムのソフトウェア開発管理機能要件

ソフトウェア生産活動の特徴は以下のとおりと考えられる。

- 1) ソフトウェアの生産は、一般にチームの協同作業により遂行される。
- 2) ソフトウェアの生産物は不可視で、知的活動の所産である。
- 3) ソフトウェアの生産は、複雑でかつ長期の工程となる。
- 4) ソフトウェアの生産にあたっては、開発メンバー相互間のコミュニケーションが必要である。

このような特徴を有するソフトウェアの開発プロセスにおいて、要求仕様どおりの品質の高いソフトウェアを所定の資源、期間、予算の範囲内で開発するためには、前述のソフトウェア管理技術が重要である。また、開発プロセスの管理に先立って、開発規模、開発工数、開発予算を正確に見積もる作業があり、これらの基本となる考え方や手法も管理技術の中に含める必要があると考えられる。開発規模はプログラムの大きさを表わす行（ステートメント）数や命令数であり、開発工数はソフトウェア技術者の作業時間を（人・日）や（人・月）で表わしたものであり、開



発費用は人件費（工数×単価）や計算機使用料（計算機使用時間×単価）などの開発関連予算である。ソフトウェアの開発管理は、ソフトウェア開発の諸活動を総合したプロジェクトを明確な目標、計画、スケジュール、予算の下で推進するという意味合いから、プロジェクト管理と呼ばれることもある。

一般に生産管理とは、製品を作り出すプロセスの管理であり、品質管理、工程管理、予算管理および要員管理といった要素から構成されていると考えられる。ソフトウェアの開発についてもソフトウェア工学に基づく考え方が必要と思われる。したがって、ソフトウェア開発管理機能の設計上の要件は、これらの開発管理活動の合理化を支援する要求から分析することが適当と考えられる。

ソフトウェア開発管理における重要な問題は、ソフトウェアのライフサイクルに基づく品質、作業期間、予算を計量化し、それぞれ品質管理、工程管理、予算管理という活動の中でいかに分析・評価して検討を加えるかということであると考えられる。

品質管理では、ユーザの要求品質を満足するように開発結果である最終品質を保証することが目的であるので、品質目標や保証基準の設定、および品質管理制度や手法の確立が重要となると思われる。特に、ソフトウェアの品質特性のうち、充足されていて当たり前で不十分であるとユーザに不満感を与えるという、「当たり前品質」としての信頼性が重要な要素であると思われる。これに対して、ユーザが製品自体に魅力を感じ満足感をもつ品質特性として「魅力的品質」がある）。

工程管理では、日程計画の立案、各開発プロセスの定義と成果物の規定、マイルストーン（工程管理の要所）設定と作業ネットワークの作成、各工程の進捗度の把握、工程の遅れや仕掛かりの検出、関連部署との調節などの作業が重要となると思われる。実際には、ソフトウェアの生産物は不可視であるので進捗状況を把握することは難しく、常に開発の進捗具合の図表化などを行ない、事前にマイルストーンでのレビューのポイントとその時点における評価項目を設定して、これに対する進捗度をチェックしていくのが効果的であると考えられる。

予算管理では、開発・保守に費やされる工数、人件費、計算機使用料などを見積もり、これらが計画どおりに投入されるように管理することが目的となると思われる。具体的には、予算と支出予算の関係把握、人員計画と要員配置、見積りと実績の差異分析、標準原価などの予算構造の明確化、損益収支分析、生産性指標の設定と改善効率の把握などが主な作業であると思われる。

図5. 1-1に、ソフトウェアの開発プロセスにおける開発管理の考え方を示す。また表5. 1-1に、ソフトウェアの開発管理技法の一例を示す。これらの管理活動を効果的に推進していくためには、管理のステップと標準化の考え方が重要である。管理のステップは、図5. 1-2に示すような計画、実行、確認、改善というサイクルを繰り返す管理の実践・維持の手順である。これは、一般の工業製品の生産管理あるいは品質管理の実施手順としてよく用いられ、管理のサイクルと呼ばれている。標準化は、ソフトウェアの開発作業の均質化を図るものであり、個人の能力に頼る職人的な手工業生産からソフトウェア工学的な工業製品の生産へと脱皮するために不可欠である。

ソフトウェア開発における標準化には、作業自体の標準化と管理活動の標準化の2つがある。前者は、ソフトウェア開発プロセスおよび保守のすべての段階にわたる作業内容の標準を設定するものである。後者は、ソフトウェアの開発管理で実施される品質管理、工程管理、予算管理など、すべての管理業務を効果的・効率的かつ円滑に行うための基準を設定することである。

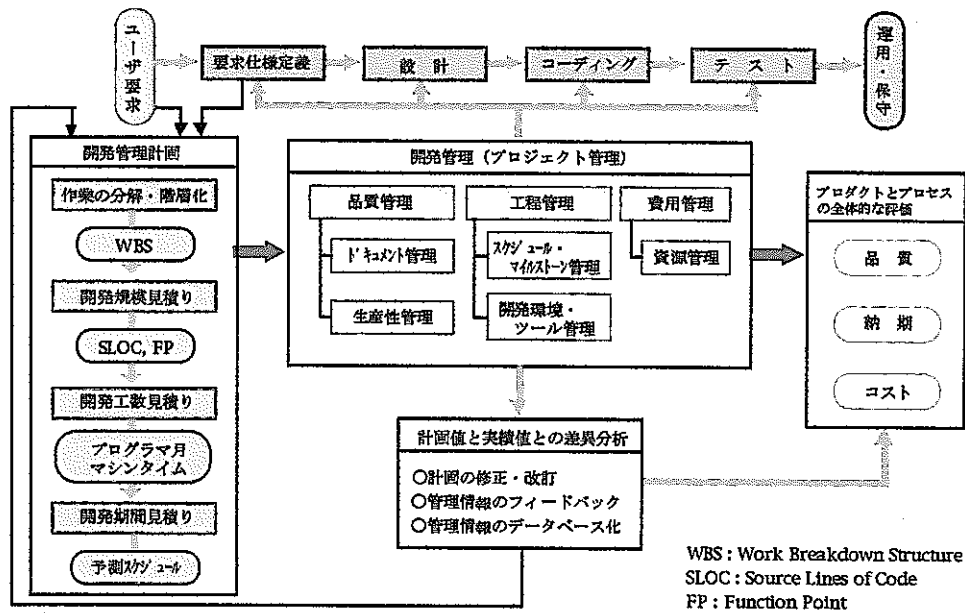


図 5. 1-1 ソフトウェアの開発管理の考え方

表 5. 1-1 ソフトウェアの開発管理技法とその評価内容

	管理技法	評価内容
品質管理	<ul style="list-style-type: none"> <li>ウォークスルー (walk-through)</li> <li>インスペクション (inspection)</li> <li>QC七つ道具</li> <li>テスト支援技術</li> <li>信頼性評価技術</li> </ul>	<ul style="list-style-type: none"> <li>品質指標の評価</li> <li>品質の定量的分析</li> <li>要求品質と出荷品質の評価</li> <li>ソフトウェアエラーの原因調査</li> <li>開発方法論とツールの評価</li> </ul>
工程管理	<ul style="list-style-type: none"> <li>ガント (Gantt) チャート</li> <li>CPM / PERT</li> <li>マイルストーンチャート</li> <li>進捗度予測シミュレーション</li> <li>トレンドチャート</li> </ul>	<ul style="list-style-type: none"> <li>計画と実績の比較</li> <li>進捗状況の把握</li> <li>納期管理</li> <li>工程の遅延に対する対策</li> </ul>
費用管理	<ul style="list-style-type: none"> <li>SLIM*</li> <li>SQAM*</li> <li>ドティ (Doty)モデル</li> <li>ココモ (COCOMO)*モデル</li> </ul>	<ul style="list-style-type: none"> <li>開発規模・工数・費用の見積りと実績の比較</li> <li>予算と費用実績の比較</li> <li>基準生産性の見直し</li> </ul>

\* SLIM : Software Life-Cycle Management  
SQAM : Software Quality Assessment and Measurement  
COCOMO : Constructive Cost Model

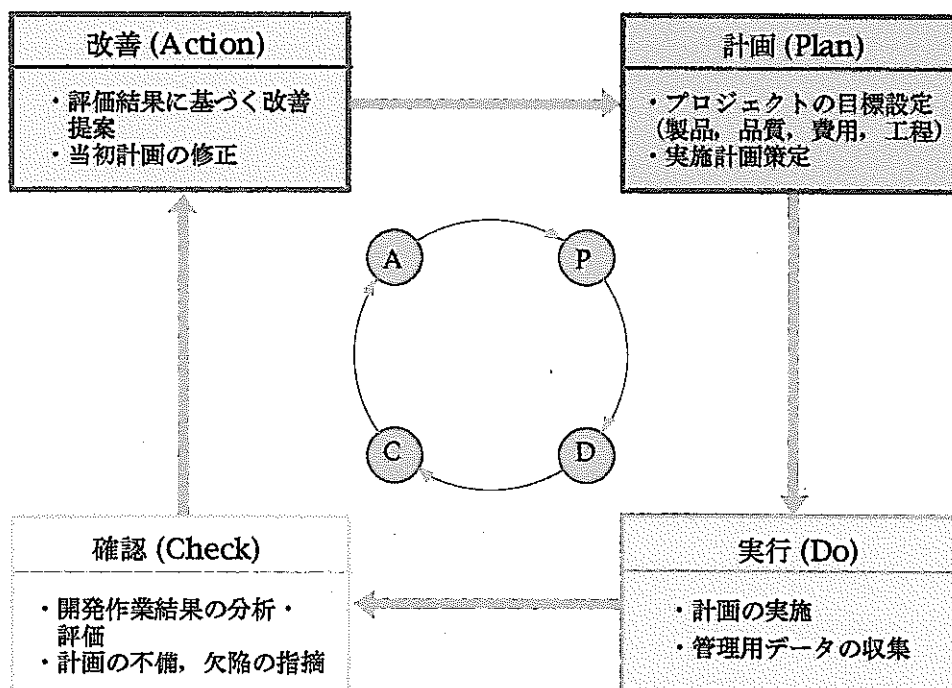


図5. 1-2 ソフトウェア開発管理に対する管理のサイクル

以上のことから、ソフトウェアの開発管理の主要な機能として、品質管理、工程管理、予算管理があり、これらの機能を効果的かつ効率的に果たすために管理のサイクルと標準化により支援する必要があることがわかる。前述したように、ソフトウェア工学の目的は、ソフトウェアの生産性と品質の向上である。生産性と品質の関係は、「品質は開発工程で作り込め」と言われることに象徴されるように、品質が生産性に寄与する最大の誘因であるという関係である。

これは、次に挙げる理由から品質が向上することにより生産性が増加することを意味する。

- ・品質が高ければ、手直しや再作成という後戻り作業は減少し、総開発予算も減少する。
- ・品質が高ければ、ソフトウェアエラーは少ないので、テスト労力の減少につながる。
- ・良い品質は、できるだけ開発の上流工程でソフトウェアエラーを補足することにより確保され、しかも補足されたエラーの修正予算は下流工程よりも上流工程のほうが安くつく。

以上のように、品質管理がソフトウェアの生産性向上につながる最も重要な要因と考えられる。そのため、計画するシステムのソフトウェア開発管理の中核的機能となるものは、プログラム品質管理機能である。したがって、本年度はプログラム品質管理機能について具体的な試作を評価し設計に反映させた。

## 5.2 プログラム品質管理機能試作

ソフトウェアの品質を向上させるためには、開発プロセスの全工程でソフトウェアエラーが潜入しないように適切な生産技術を適用するとともに、潜入したソフトウェアエラーはできるだけ早期検出するよう組織的に管理していく総合的品質管理の考え方が重要であると思われる。そのため、ソフトウェアの品質特性の定義、および品質の定量的尺度とその計測法が必要となる。たとえば、一般的なソフトウェアの品質特性は、図5.2-1のようなものである。

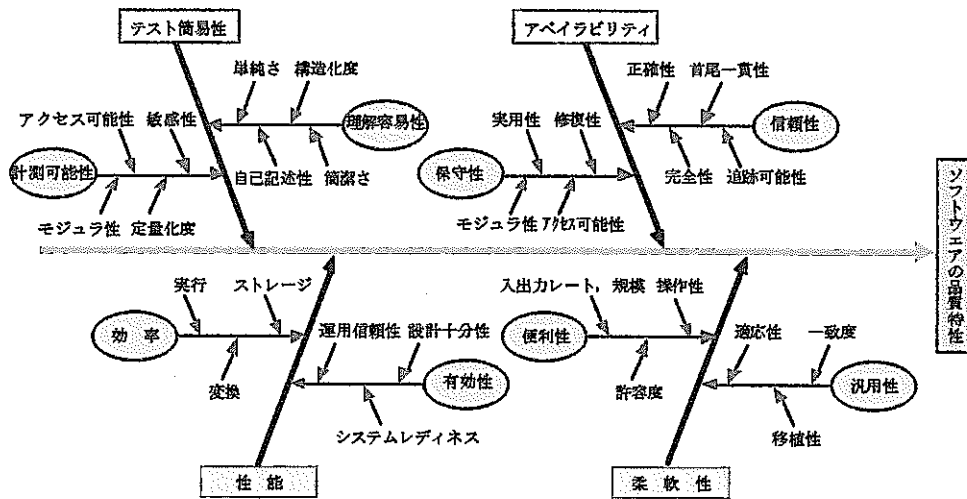


図5.2-1 特性要因図による階層化されたソフトウェアの品質管理

このようなソフトウェアの品質特性を考える上で、品質の構成要素を次のようにプロダクト品質とプロセス品質に分けると明確になり理解しやすい。

### ・プロダクト品質

開発作業中に潜入したソフトウェア内の誤りや欠陥である。これは、ソフトウェア開発の結果としての製品品質である。これを管理するには、要求仕様書・設計書・ソースコードの明瞭性、仕様のトレーサビリティ、信頼性、またテストの網羅性などの評価を自動化することが機能として必要と考えられる。

### ・プロセス品質

ソフトウェア開発環境が寄与する品質であり、開発技術の正確性、ツールの生産性、開発要員の能力・スキル、開発チームのコミュニケーション、開発設備（開発用マシンなど）の利用可能度などが含まれる。これを管理するためには、体制の整備や開発環境の評価等の技術が必要と考えられる。

さらに、ソフトウェア品質の構成要素を次のように動的品質と静的品質に分けて考えると、品質を計量化する際に有用であると思われる。

### ・動的品質

ソフトウェアをコンピュータ上で実行して、その挙動を調べることにより確認される品質であると考えられる。これを管理するためには、平均ソフトウェア故障時間間隔、ソフトウェア故

障の修復時間、利用可能資源の利用率などをパラメータとする手法が考えられる。

・静的品質

ソフトウェアを実行せずに、各工程の作業結果および成果物を審査するレビューや検査により確認される品質である。これを管理するためには、モジュール化の程度、簡潔性、完全性などの評価が考えられる。

上記のソフトウェア品質の構成要素を特性要因図にまとめたものが図5. 2-2である。図に示す品質特性の考え方は、自動化する場合のデータフローの基本概念を与えている。

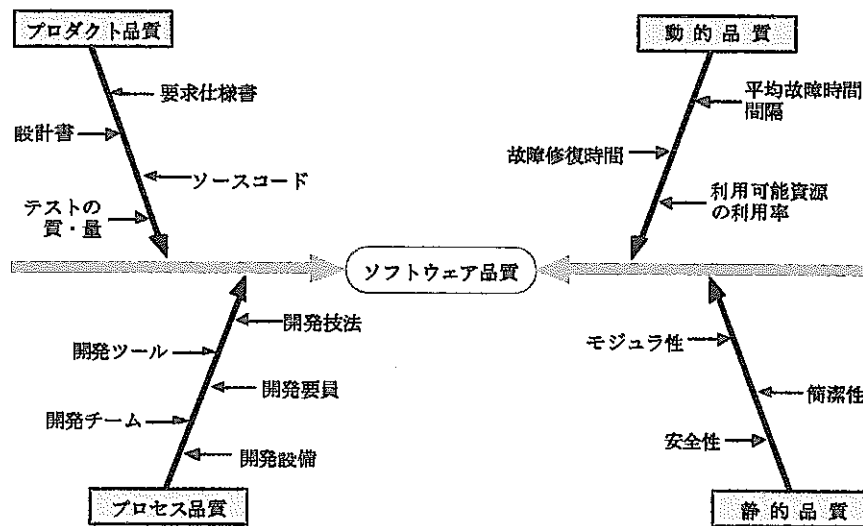


図5. 2-2 ソフトウェアの品質特性の考え方

ソフトウェアの品質はユーザの要求に基づくものであるから、これと開発プロセスで作り込まれる品質特性との関係を把握しておかなければならないと思われる。要求仕様として取り込まれるユーザ要求は、基本的には、機能と性能に関する運用特性と、修正・変更とその管理に関する保守特性からなる。表5. 2-1に、これらのユーザ要求と品質特性の関係を示す。

表 5. 2-1 ユーザ要求と品質特性

ユーザ要求	要求品質の定義	品質特性
機能	<ul style="list-style-type: none"> <li>・ 多少の誤りがあっても途中で止まることなく動作するか</li> <li>・ どのくらいの頻度でソフトウェア故障が発生するか</li> <li>・ 故障期間中に生き残れるか</li> <li>・ どの程度使いやすいか</li> </ul>	<ul style="list-style-type: none"> <li>堅固性 (integrity)</li> <li>信頼性 (reliability)</li> <li>残存性 (survivability)</li> <li>使用性 (usability)</li> </ul>
性能	<ul style="list-style-type: none"> <li>・ どのくらい資源を必要とするか</li> <li>・ 要求仕様どおりであるか</li> <li>・ 異常な状況を回避できるか</li> <li>・ 他のシステムに結びつけやすいか</li> </ul>	<ul style="list-style-type: none"> <li>効率 (efficiency)</li> <li>正確性 (correctness)</li> <li>安全全性 (safety)</li> <li>相互動作性 (interoperability)</li> </ul>
修正・変更	<ul style="list-style-type: none"> <li>・ 発生したソフトウェアエラーを修正しやすいか</li> <li>・ 機能拡張しやすいか</li> <li>・ 運用中のソフトウェアを変更しやすいか</li> <li>・ 他のシステムに移植しても正しく動作するか</li> <li>・ 他のシステム中に再利用が出来るか</li> </ul>	<ul style="list-style-type: none"> <li>保守性 (maintainability)</li> <li>拡張性 (expandability)</li> <li>柔軟性 (flexibility)</li> <li>移植性 (portability)</li> <li>再利用性 (reusability)</li> </ul>
管理	<ul style="list-style-type: none"> <li>・ 性能の検証は容易か</li> <li>・ そのソフトウェアは管理しやすいか</li> </ul>	<ul style="list-style-type: none"> <li>検証可能性 (verifiability)</li> <li>従順性 (manageability)</li> </ul>

### 5.3 ソフトウェア品質管理機能の試作

ソフトウェア品質管理機能の基礎的な部分を評価用に試作し、設計の参考とした。試作評価の結果、ソフトウェア品質管理機能は、全体を取りまとめる対話システムを開発することにより、個別処理機能は、ツールソフト等を組み込み構成することが可能と考えられた。

#### (1) プログラム品質管理機能試作

プログラム品質管理機能については、基本的な課題として静的解析機能を試作評価した。静的解析機能は、ソースプログラムの構造上の誤りやデータ参照の誤りなどの、形式的な誤りを評価する品質評価手法である。

図5.3-1に評価メニューを示す。

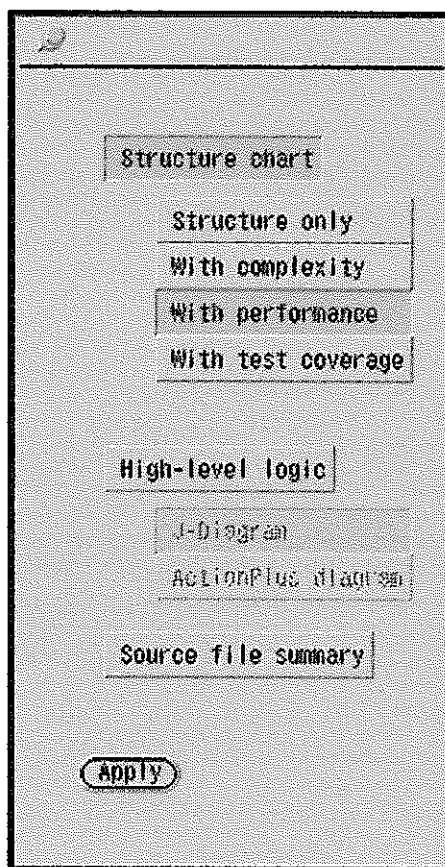


図5.3-1 静的解析機能メニュー

図5.3-1は、下記のソースプログラム品質の解析を行うメニューを示している。

#### Structure chart

- |                                    |                |
|------------------------------------|----------------|
| Structure chart only               | : 構造図表示        |
| Structure chart with complexity    | : 構造図と複雑度を表示   |
| Structure chart with performance   | : 構造図と性能を表示    |
| Structure chart with test coverage | : 構造図とテスト範囲を表示 |

## High-level ogic

- J-Diagram : J-Dアイアグラムを表示
- Action diagram : 動作ダイアグラムを表示

Source file summary : ソースの概要を表示

図 5. 3-2 に評価対象となるソースプログラムをブラウズするメニューを示す。

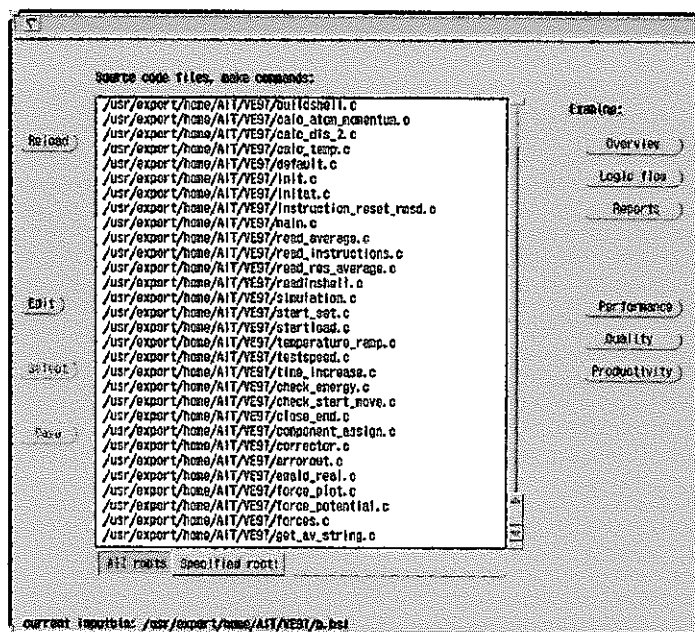
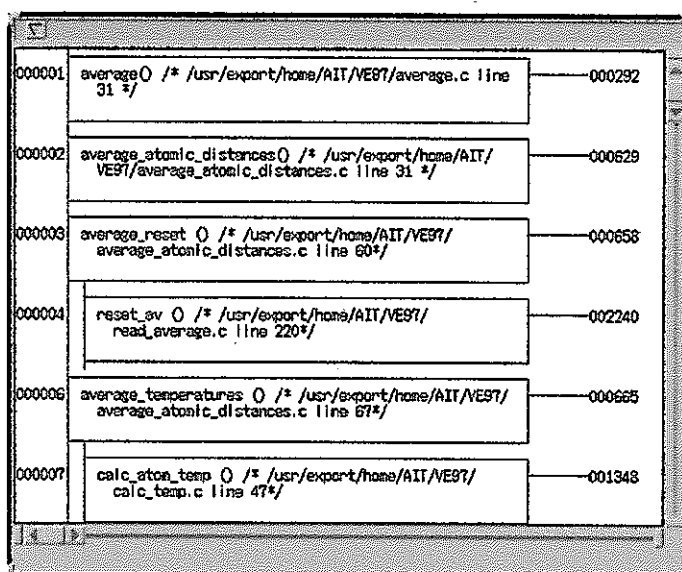


図 5. 3-2 品質評価ソースプログラム・ブラウザの例

図 5. 3-3 に、関数相互参照解析結果を示す。



5. 3-3 関数相互参照解析結果の例



図5. 3-4に、構造図と複雑度解析結果を示す。

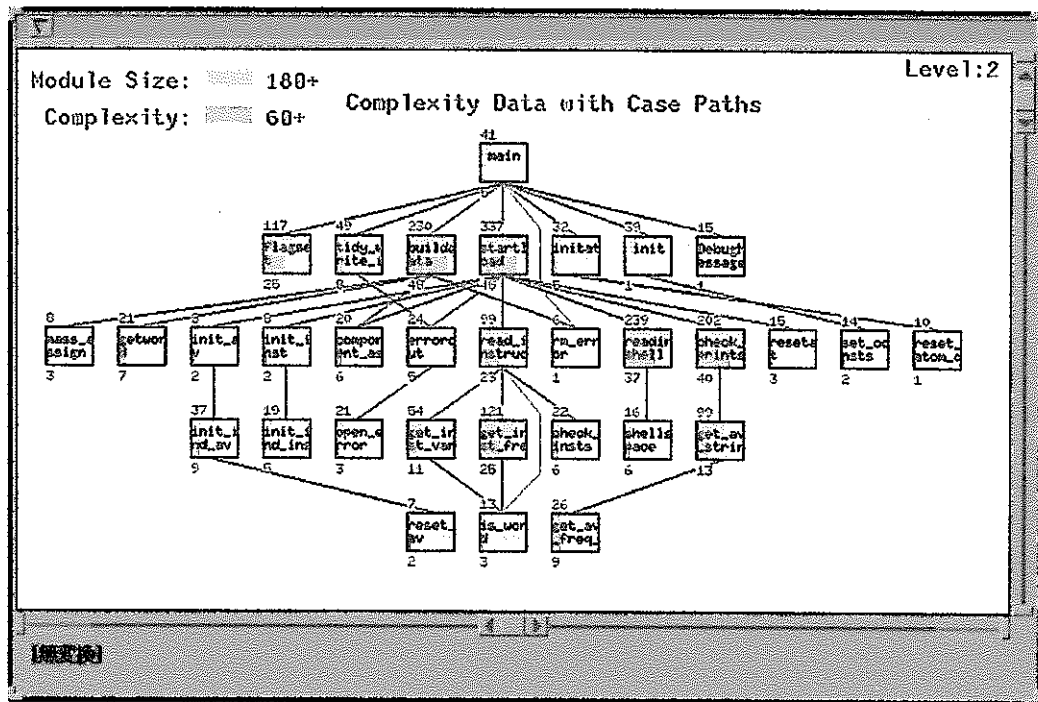


図5. 3-4 構造図と複雑度解析結果の例

(2) 数値データ品質管理機能試作

数値データの品質管理機能は、数値ごとと信頼性範囲を示す値をペアにして管理する方法を検討した。評価用事例としてRELEASE 解析結果を用いた。図5. 3-5にその例を示す。

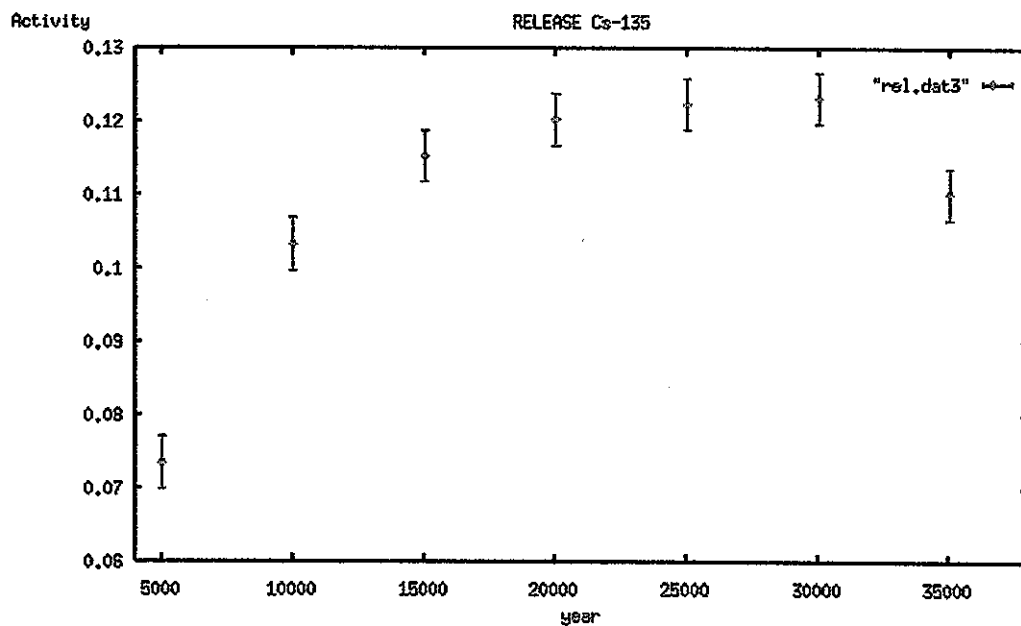


図5. 3-5 数値データ品質管理の例

### (3) 文書情報品質管理機能試作

文書情報品質管理機能は、文書を直接解析する方法と履歴等の関連情報を管理する手法が考えられた。文章を直接解析する手法は問題が複雑になることや、曖昧さを避けられないことなどから、今回の試作では、文書に関連する履歴情報を管理する手法をとった。評価用事例としてFEPシナリオのリストを用いた。

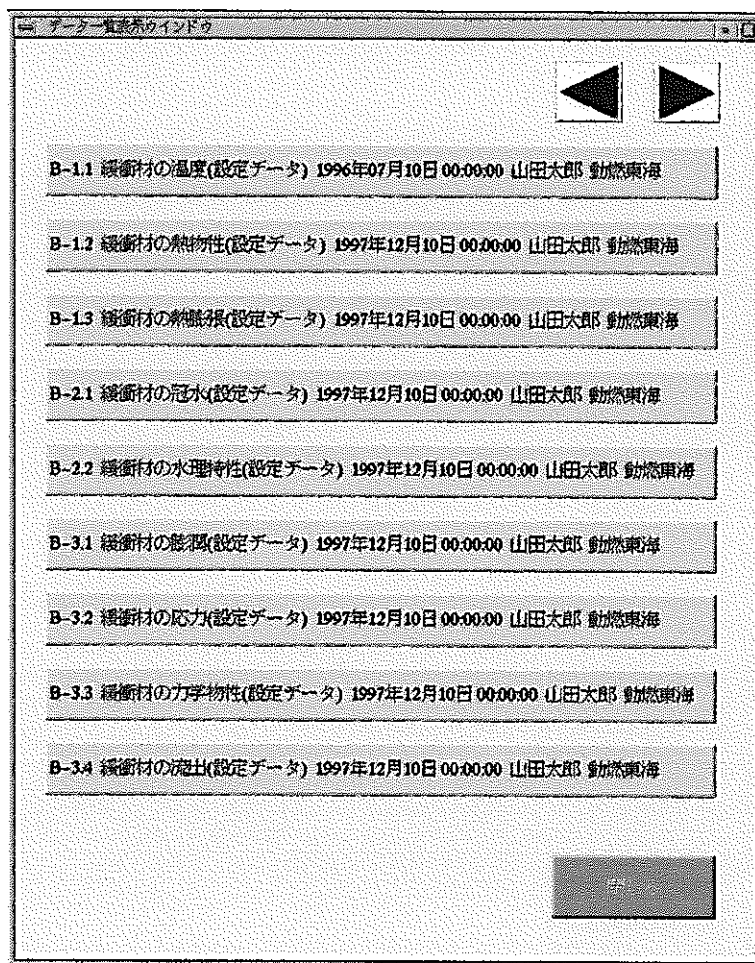


図 5. 3-6 文書情報品質管理画面

## 6 実証・評価用システムの改良整備

前年度の研究においては、個々の基盤技術を開発するための開発環境として実証・評価用システムを構築した。その結果、対話機能、シミュレーション機能、データベース管理機能およびソフトウェア開発管理機能に関わる基盤技術についての確認ができた。このため、本年度は前年度の成果をもとにシステムの設計を固めるための試作・評価等につき、下記項目の改良作業を実施した。

### 主操作システムの改良

主操作システムの改良は下記について実施した。

#### ア. ソフトウェアの改良

対話機能を提供するプログラムを本年度の設計に応じて改良した。

#### イ. ハードウェアの構成変更

別途、設計段階にて判断する

### シミュレーション用システムの改良

シミュレーション用システムは下記について実施した。

#### ア. ソフトウェアの改良

シミュレーション機能を提供するプログラムを本年度の設計に応じて改良した。

#### イ. ハードウェアの構成変更

別途、設計段階にて判断する

### データベース用システムの改良

データベース用システムは下記について実施した。

#### ア. ソフトウェアの改良

データベース機能を提供するプログラムを本年度の設計に応じて改良した。

#### イ. ハードウェアの構成変更

新たにデータベース管理用ワークステーションを1台本システムに増設した。

### パラメータ・ディスプレイ・システムの改良

パラメータ・ディスプレイ・システムは下記について実施した。

#### ア. ソフトウェアの改良

パラメータ表示機能を提供するプログラムを本年度の設計に応じて改良した。

#### イ. ハードウェアの構成変更

3次元オブジェクトをパラメータとして表示するためのOpenGL対応PCを2台本システムに増設した。

### オーディオ・ビデオ・システムの改良

オーディオ・ビデオ・システムは下記について実施した。

#### ア. ソフトウェアの改良

マウス、シミュレーション開始・終了、等のイベントを警告するビーブ音の編集再生が可能となるよう、PCMサンプル（デジタル音）機能を利用したレコーディングとプロセスでの再生アルゴリズムを組み込む。

イ. ハードウェアの構成変更

別途、設計段階にて判断した。

前年度の研究においては、個々の基盤技術を開発するための開発環境として実証・評価用システムを構築した。その結果、対話機能、シミュレーション機能、データベース管理機能およびソフトウェア開発管理機能に関わる基盤技術についての確認ができた。このため、本年度は前年度の成果をもとにシステムの設計を固めるための試作・評価等につき、下記項目の改良作業を実施した。

主操作システムの改良

主操作システムの改良は下記について実施した。

ア. ソフトウェアの改良

対話機能を提供するプログラムを本年度の設計に応じて改良した。

イ. ハードウェアの構成変更

別途、設計段階にて判断する

シミュレーション用システムの改良

シミュレーション用システムは下記について実施した。

ア. ソフトウェアの改良

シミュレーション機能を提供するプログラムを本年度の設計に応じて改良した。

イ. ハードウェアの構成変更

別途、設計段階にて判断する

データベース用システムの改良

データベース用システムは下記について実施した。

ア. ソフトウェアの改良

データベース機能を提供するプログラムを本年度の設計に応じて改良した。

イ. ハードウェアの構成変更

新たにデータベース管理用ワークステーションを1台本システムに増設した。

パラメータ・ディスプレイ・システムの改良

パラメータ・ディスプレイ・システムは下記について実施した。

ア. ソフトウェアの改良

パラメータ表示機能を提供するプログラムを本年度の設計に応じて改良した。

イ. ハードウェアの構成変更

3次元オブジェクトをパラメータとして表示するためのOpenGL対応PCを2台本システムに増設した。

オーディオ・ビデオ・システムの改良

オーディオ・ビデオ・システムは下記について実施した。

ア. ソフトウェアの改良

マウス、シミュレーション開始・終了、等のイベントを警告するビーブ音の編集再生が可

能となるよう、PCMサンプル（デジタル音）機能を利用したレコーディングとプロセスでの再生アルゴリズムを組み込む。

イ. ハードウェアの構成変更

別途、設計段階にて判断した。

前年度の研究においては、個々の基盤技術を開発するための開発環境として実証・評価用システムを構築した。その結果、対話機能、シミュレーション機能、データベース管理機能およびソフトウェア開発管理機能に関わる基盤技術についての確認ができた。このため、本年度は前年度の成果をもとにシステムの設計を固めるための試作・評価等につき、下記項目の改良作業を実施した。

主操作システムの改良

主操作システムの改良は下記について実施した。

ア. ソフトウェアの改良

対話機能を提供するプログラムを本年度の設計に応じて改良した。

イ. ハードウェアの構成変更

別途、設計段階にて判断する

シミュレーション用システムの改良

シミュレーション用システムは下記について実施した。

ア. ソフトウェアの改良

シミュレーション機能を提供するプログラムを本年度の設計に応じて改良した。

イ. ハードウェアの構成変更

別途、設計段階にて判断する

データベース用システムの改良

データベース用システムは下記について実施した。

ア. ソフトウェアの改良

データベース機能を提供するプログラムを本年度の設計に応じて改良した。

イ. ハードウェアの構成変更

新たにデータベース管理用ワークステーションを1台本システムに増設した。

パラメータ・ディスプレイ・システムの改良

パラメータ・ディスプレイ・システムは下記について実施した。

ア. ソフトウェアの改良

パラメータ表示機能を提供するプログラムを本年度の設計に応じて改良した。

イ. ハードウェアの構成変更

3次元オブジェクトをパラメータとして表示するためのOpenGL対応PCを2台本システムに増設した。

## オーディオ・ビデオ・システムの改良

オーディオ・ビデオ・システムは下記について実施した。

### ア. ソフトウェアの改良

マウス、シミュレーション開始・終了、等のイベントを警告するピープ音の編集再生が可能となるよう、PCMサンプル（デジタル音）機能を利用したレコーディングとプロセスでの再生アルゴリズムを組み込む。

### イ. ハードウェアの構成変更

別途、設計段階にて判断した。

## 7. まとめ

### 統合型シミュレーションへのアプローチについて

従来のシミュレーション研究は、ある現象を取り扱う時に還元された単純な現象をモデルとして考えてきた。これは、ある意味では他の不純物のない現象のみを扱うという考え方である。そういった数理モデルは、例えば、運動ならば運動だけというようなモデルを表す式を開発することにより得られてきたと考えられる。その結果、数理モデルを開発し、それを数値的に解くための道具としてコンピュータが位置付けられてきたと思われる。そのようにコンピュータを利用するならば、コンピュータはインプットを与えることにより高速に計算する機械という位置付けに過ぎない。

また、そのようにして、例えば熱、流体、構造、さらにその他諸々のひとつの現象の要素の領域のモデルが組み立てられプログラムされてきたと思われる。そして、それらのモデルが試験の結果を使って正しいかどうか確証を得る作業が継続して行われて来たと思われる。そのようにして確証が得られれば、全てのモデルが個別の現象に対して正しく記述しているために全体として記述は正しいという判断を得られることができる。それが従来のシミュレーションの一つの手段としてのシミュレーション、いわゆる戦略であったと考えられる。

そのように、個別に現象を分解・還元して、還元された領域の数理モデルを組み立てる活動を還元論的手法と呼ぶことにする。還元論的手法では、個々の事象・現象は単純な問題である。そして、それらの単純な問題が正しいから全体が正しいという仮説を用いている。しかし、複雑で分割の難しい連成問題等を考えると、還元論的手法成立する場合についてのみ従来型のシミュレーションの戦略が適用できるのではないかと思われる。たとえば、ニュートン力学が挙げられる。しかし、現実に対象となる熱・流動問題などでは熱力学的平衡問題のように非可逆的な問題が多く、単純なニュートン力学的アプローチだけでは研究を推進することが難しいと考えられる。そのため、そのような還元論的手法の数理モデルの研究開発というものが複雑な現象全体を記述できる、もしくは、そういった現象を記述するということに十分に戦略的に有効であるかという事に関する疑問が最近生まれてきていると思われる。たとえば、2つの物理量である熱と圧力や流れが相互に関連し合っ一つの現象を記述するような問題は、安全解析などでは多く見られると思われる。

そういった連成問題は、熱だけの現象や圧力だけの現象を表すモデルに比べて、はるかに複雑なものとなると思われるが、それら個別現象のモデルだけで記述し得る領域を超えた、新しい現象をシミュレートすることが可能になると考えられるため、研究開発上の意義は大きいと思われる。

還元論的立場からも連成問題は既にそれ以上分割可能な問題とは考えられない最小の単位として扱われてきた。連成問題を還元することによって、連成自身の性格を失ってしまうために、それ自身のシミュレーションとしての価値を完全に失ってしまう。従来、個別に還元されたシミュレーション問題として、そのような連成方程式が扱われてきたが、連成問題を解く、もしくは連成モデルを開発するといったことが、課題として還元論的な戦略の中から生まれてきた事は既に周知の通りである。そのような連成問題として、例えば構造物と水の相互作用などが挙げられる。

例えば薄い板に対する流れの衝撃、このような問題を解く場合、一つの力学的な問題だけを解

明するとしても、片方は流体の力学、片方は構造の力学に基づいたものでなければならない。流体の振る舞いは、一般にナビエ・ストークス方程式によって与えられる。一方、構造の振る舞いは有限要素法により考えられる場合が多い。ナビエ・ストークス方程式自身は、流れの場を記述するために、質量の保存則、運動量の保存則を代表とするものである。場合により、エネルギーの保存則を追加する。そのように連立させて解くことにより、流れと熱を含む挙動を示すモデルを開発する。また、有限要素法に基づく構造のモデルは、前述のように変分原理等に基づいたもののため、それらの流れのモデルと構造物の変形のモデルは基本的に異なっている。

連成問題に於ける議論は、そのように2つの異なったモデルを数値解析法などで結びつけ、一つのプログラムとして、どのようにインプリメントするかと言うことに重点が置かれていたと思われる。その結果、個別に還元されたモデルでは従来到達し得なかった相互作用の解析を行うことができるようになった。それは、一つの連成問題を通じたモデルの統合型シミュレーションのアプローチの成功例として考えられる。以上のように、従来からのシミュレーションに基づいた連成問題、統合型シミュレーションのアプローチは、従来とられてきた戦略として、まず個別のモデルを組み立て、それらのモデルを組み合わせるといふ、基本的に数理解析に基盤を於いた領域の活動であると考えられる。それはまた、各種の状態量を評価するためのモデルを開発し、それらを連立させて解くという科学手法であるとも考えられる。

ヴァーチャル・エンジニアリングとして考えているヴァーチャル・エンジニアリング・システムの領域は、そのような連成問題の領域と一部異なってくる新しい領域が考えられる。それは、モデルをどのように扱うかという事により、また、だいたいの意見が異なると思われるが、ヴァーチャル・エンジニアリング・システムにおけるアプローチは、個別のモデルをどのように連立させるかについて解析的に連立させるという考え方とは異なり、コンピュータの中で相互のモデルを通信させることにより、試験的に計算するという活動を想定していると考えられる。従って、従来の連成解析に於けるリサーチャー、研究者の立場は、そのモデルの記述の範囲を重視した研究であると考えられる。それは、連成解析のモデルを開発するための一つの活動であるという事もできる。従って、そのような観点から考えると、従来の連成モデルへのアプローチというものは、あくまで目的としては、連成モデル、数理モデルを開発することが目的であり、コンピュータの利用というものは、そのための確証、検証のための手段に過ぎないと考えることができる。

しかし、従来行ってきたシミュレーション活動は、現象の側面を捉え、それを記述するモデルを数理として示し、さらにその示された数理に従って実現されたプログラムというアルゴリズムをコンピュータ内部に開発する活動である。そして、その活動は工学的には成功してきたと思われる。従って、モデル開発をとり行う研究者にとって興味の核心となることは、まず観測を正確にとり行い、モデルとなる現象を還元化し、それを示す基礎方程式、支配方程式を導き出すことである。そのようなアプローチの結果、一つの領域のモデルが開発されてきた。我々はそういったモデルを持って解析コードを開発し、工学的分野で応用してきている。そのような応用は、特に要素設計等の現場で非常に成功した例だと考えられる。例えば、熱流力、構造、そういった分野に於いては、個別の設計条件を決定するためにモデルを使い、試験を行い、評価計算をとり行うが、その時にそのようなシミュレーションが非常に有効な手段である。以上のように、従来のシミュレーションという観点から考えた場合に、研究の領域は個別の現象をモデル式として記述



するというところに重点が置かれ、そのモデル式の持つ正確さ、不確実性、それらの検証確認という技法に非常に技術の核心が集中してきたと考えられる。

それらの活動の結果、多くのシミュレーションコードが開発されてきたと考えられる。従来のR&D等、段階に於いては動力炉の開発、核燃料サイクルシステムの開発、その開発を推進するためには、施設、機器などの設備の設計を固めるためにこれらのモデルが活用されてきた。さらに、設計側の観点からすれば、設計条件を決定する論拠を決定することが研究の核心に触れる部分であり、それを超えた範囲での影響を考慮することは、本来のモデル開発に対する外乱と考えられるため排除されなければいけないという基本的な姿勢がある。つまり、モデルの純粋性を保つことが重要と考えられる。そうしない限り、一つの個別現象レベルでのモデル開発を円滑に推進することは困難であると考えられる。そのため、従来設計評価用などの目的で個別に開発された現象解明コードを統合型シミュレーションするという事を考えた場合に、それらの間には個々に標準化、もしくは一つの基準となるような制度は設けられておらず、基本的に別個のものとして考えざるを得ない。従って、それらをヴァーチャル・エンジニアリング・システムのソフトウェア資源として扱うことを考えるならば次の問題が生じると思われる。

従来個別に自由に開発されてきたコードは、モデル記述、実行環境、さらにプログラミング、文法、それらが全て管理されておらず（ここで言う管理というのは標準化という意味である）、個人の意思で開発されてきたと思われる。従って、ある特定のコードの変数の領域や分岐など、そういった個別の詳細仕様決定やロジックの決定、アルゴリズムの決定に於いても、それは標準化されておらず、個々にプログラマの嗜好で決められてきたと考えられる。そのように、従来型の個別解析コードに於いては、それらは全く個々に別物であり、相互に関係を取り持つような標準化というものは初めから考慮されていなかったと考えられる。そのような観点から考えると、ヴァーチャル・エンジニアリングという要求に沿って、個別のモデルを統合型シミュレーションするという計画を円滑に推進するためには、その第一の問題点として、個々に開発されてきたコードの経歴上それらが相互に接続することを意識していなかったという事が挙げられる。この事実は、ヴァーチャル・エンジニアリング・システムを過去に開発する計算コードで実現する戦略に大きな壁となると思われる。従って、今後のシミュレーションコード、ないしは数理モデルの開発に於いては、その再利用を有効に実現するために標準化が要求される重要な課題として挙げられる。

ヴァーチャル・エンジニアリングのプロジェクトの計画と推進とともに、統合型シミュレーションという言葉が最近頻繁に使われるようになってきた。ここでは、その統合型シミュレーションについて考えてみる。特に、ヴァーチャル・エンジニアリングという一つのプロジェクト、その中に於いて、統合型シミュレーションという言葉がどのような意味を持って、どのような技術の領域について示されているか、さらに、その考え方はどのようなものであるか、そしてその問題点とはどのようなものであるかについて検討した。まず、統合型シミュレーションという言葉を使う場合に、どのような領域を意識しているのだろうか。それは情報の領域であると考えられる。少なくともここでは、今後のヴァーチャル・エンジニアリングの基盤技術としての統合型シミュレーションという事で考えてみたいと思う。

従って、統合型シミュレーションという言葉は、それが情報処理システムの中に於けるソフトウェアのインテグレーションという技術課題に結びつくのではないかと思われる。情報システム

のインテグレーションという一つの技術の領域の中に於いて、それを大別すると、技術面から見ればハードウェアのインテグレーション、そしてソフトウェアのインテグレーションが考えられる。一方、プロジェクトの管理面から見ると、資材購買やプロジェクトの管理などが考えられる。そのため、ヴァーチャル・エンジニアリング・システムを広義のシステム・インテグレーションの1領域であるソフトウェア・インテグレーションとして捉えてみる必要があると思われる。

ソフトウェアのインテグレーションという技術の領域は、非常に多くの情報の領域の一つに取りまとめるものと考えられるが、また単にそのような表現の場合には、漠然としていると思われる。従って、これらの用語を統一して統合型シミュレーションと言うことにしたが、統合型シミュレーションと言った場合には、対象とする情報処理システムやヴァーチャル・エンジニアリング・システムを実現するための情報をどのように統合型シミュレーションするかという命題が与えられるものと思われる。特に、本研究においてはシミュレーションという一つの課題として、従来蓄積されてきた、数理モデル開発、シミュレーション・コード開発、それらの遺産であるコードやデータをどのように接続し、より効率的に再利用することにより、今後生じるであろう複雑な問題・現象の解明に資することが出来るのかに関心が寄せられていると考えられる。

従って、そのような背景から考えるならば、統合型シミュレーションとは、まずR&Dの成果と考えられる従来の個別現象解明、それにより得られた知見、個々の個別現象解明により得られた知見をどのように統合型シミュレーションするかという事だと考えられる。それは、少なくとも還元論的手法により、とり行われてきた個別現象解明の成果として得られた数理モデルの限界を打破するための技術的なアプローチであるため、基本的には連成問題への戦略と考えられる。たとえば、流れの場合には代表的なナビエ・ストークスの方程式が与えられる。これは、時間を無視して考えることはできない。しかし、構造物の静的解析に於いては、時間は考慮するパラメータとしてはそれほど重要ではない。従って、一般的には時間を含まない解析、力学系のみ解析、釣り合いの解析として問題をモデル化することができると思われる。そのような個別の領域に於ける理想化、モデル化に於いては、個々の研究者の判断により、どのようなパラメータが支配的であるかが判断される。その結果、先ほど示したように、構造物の静的解析といわれる分野では、時間よりむしろ釣り合いに着目してモデル化することに関心が寄せられると考えられる。

一方、流れの世界に於いては時間が支配的である。実際にナビエ・ストークスの方程式に示されるように、流れのモデルを記述するためには時間抜きには記述することができない。そのため、流体の問題を研究する研究者にとっては時間は常に支配的である。そのようにして、個別の領域のモデルが開発されてくるため、それらを統合型シミュレーションするという事を考えた場合に、非常に多くの問題点が生じると思われる。特に、初期の段階に於いて、将来連成問題として取り扱う、もしくは相互に情報交換するといった協調のモデルを考えていないモデル開発、先ほど示したような静的解析のモデル、単純なナビエ・ストークスの流れのモデルのようなモデルを接続するといった場合に、ほとんどの場合は現実的には不可能であると思われる。もしくは、実際にそういったプログラムを開発すること、プログラミングをすることはできるが、そのアルゴリズム自体が現象解明にとってなんら意味を示さないか、モデルとしての意味がないものになってしまうと考えられる。

例えば、静的解析の問題に流れの問題を連成させることを考えてみると、問題自身の意味が問題になると思われる。なぜならば、静的解析モデルは時間を含まないため、たとえば水圧や流れの加速度を時間に依存する形で境界条件として与えることができない。もちろん、最大値を静加重として与えることができるが、問題の関心が時間依存の境界条件に対する構造物の応答とした場合、あまり意味は無いと思われる。このような場合、いずれにしる構造物の応答モデルに時間を導入して動的解析する必要がある。しかし、本来の構造解析コードが静的解析のみのモデルで開発されたものであれば、時間依存のモデルを扱うにはプログラムの開発を全面的に行わなければならないことも予想される。

しかし、例えば有限要素解析に於いては、熱応力解析は代表的な連成問題として実績を持っている。有限要素解析の基本的なアプローチは、まず熱伝導方程式を対象となるモデルの領域に關して時系列的に解き、計画した時間内での温度分布を全て算出する。次に算出した温度分布を用いて変位を計算する。この時、先ほどの温度分布は変位を計算する前に計算しておき、各時間ステップ毎の変位計算のためのインプットとして扱われることがある。そのような同時じ生じる現象を個別に考えることは現実にはあり得ないが、ソフトウェアのアルゴリズムとしては考えられる。以上のように、有限要素法に於ける従来の熱応力解析は2パスの解析であり弱連成と考えられる。そして、これは成功した例である。

一方、流れの問題に於いて、ある物質移行のモデルを扱う世界があるが、そのような世界に於いては、初めから物質移行と流れ、これらが連成すると考える。例えば、放射性廃棄物処分に於ける広域地下水の汚染物質移行モデルが挙げられる。モデルは一般的に輸送方程式で与えられる。そのような考え方の中には、先ほど示した熱応力解析の場合も含め、モデル開発の当初から連成という考え方が取り込まれていると考えられる。したがって、研究者はある問題を自分で定め、それに従って試験をするわけであるが、試験の結果、その現象をモデル化するに当たり支配的な要因が何であるかという事に着目する。その時に支配的な要因が単一でない場合、例えば一つのパラメータでなく、複数のパラメータの組み合わせによって一つの現象を記述せざるを得ない場合、その研究者は連成問題としてそれを取り扱おうと思われる。このような連成問題という考え方は決して新しいものではない。

このように、従来のモデル化の基本は、前述したように還元論的な戦略を適応して行われてきたわけであるが、支配的なパラメータが一つに置き換えられない場合には複数になる。そのような現象、たとえば流体が層変化するような場合などが考えられる。水が蒸気になるといった場合、これは水と蒸気の連成になると考える。従って、従来のアプローチに於いて連成問題という考え方は普及しており、研究者は連成問題へのアプローチを理解していると考えられる。

そのため、ヴァーチャル・エンジニアリング・システムの技術開発において問題となるのは、連成問題ではなく、むしろ個別モデルの取り扱いであると思われる。さらに、個々に開発されたソフトウェアをどのように管理し統合型シミュレーションするか、接続するかということが統合型シミュレーションのための重要な論点となると思われる。ヴァーチャル・エンジニアリング・システムにおいては、各種のコードを複雑に組み合わせる必要があるだろう。少なくとも、ヴァーチャル・エンジニアリングを実現するための要件として、そのような要求があると思われる。したがって、統合型シミュレーション技術は、個別に開発されたコードをどのように接続するか、

という純粋にソフトウェア工学的な課題と、そのように接続するというアプローチを考えた場合のモデルの妥当性をどのように示すかという現象論的な課題が考えられる。

以上のように、ヴァーチャル・エンジニアリング・システムの実現には現在2つの大きな問題点が考えられる。一つはソフトウェアの標準化、インターフェイスの標準化、もう一つは工学的妥当性とその検証確証の手法ということである。後者については、本研究の領域を超えるものとも思われるが、非常に重要な課題であると考えられる。なぜならば、そのようなアプローチは、物理数学的に意味があるのかということを確認しなければいけないためである。その確認が得られない限り、そのようなアプローチは基本的に意味がないものと思われる。また、ヴァーチャル・エンジニアリング・システムという観点から考えた場合、既存のソフトウェアを単純に接続すること自体に、どれほどの意味があるのかについて十分検討することは最も重要な課題であると考えられる。

#### 今後の技術の研究開発要件

ヴァーチャル・エンジニアリング・システムという技術を開発する場合の要件についてまとめる。前述のとおり、統合型シミュレーションというアプローチはソフトウェア工学的には可能であると思われる。つまり、そういったプログラムを開発する事は可能である。しかし、それ自身が与える結果が、実際のモデルを反映しているかということについては全く不透明であると思われる。従って、ヴァーチャル・エンジニアリング・システムと言われる技術の領域をどのように定義するかということは非常に重要な課題であると考えられる。

ヴァーチャル・エンジニアリング・システム技術というものは、2つの技術の領域から成り立っていると考えることにした。その一つは、純粋にソフトウェア工学的な領域として捉える方法である。それは、従来ソフトウェア工学の領域、つまりプログラマー、システムエンジニア等といわれる人たちの活動の領域であったプログラミング技術にあると考えられる。それは、既に蓄積されているFORTRAN、C等の解析コードや、これらのコードの相互のインターフェイスをどのように組み立てるかに関する技術の領域である。さらにビジュアライゼーションも含まれると考えられる。インターフェースに於いては、例えばFORTRANコードによる計算結果、Cコードによる計算結果、これらの計算結果が相互に影響するモデルが考えられる。

例えば、AというコードはFORTRANで書かれており構造解析を行うとする。このコードは動的解析機能まであるので、時間依存のシミュレーションを行うことができるとする。次にC言語で書かれた流れの解析コードがあるとする。そのコードは、基礎的なナビエ・ストークスの方程式で解析されて示されており、一般的な流れをシミュレーションすることができるとする。この2つのコードをカップリングさせて構造物に及ぼす流れの影響、例えば津波のような影響を評価することを考えるとすると、そういった計算を実現することに於いては、少なくともそれら2つのコードが開発された段階に於いては、相互に接続することは意識されていないことが最大の障害になることが予想される。従って、実際にどのように接続するのか考えた場合、この2つを組み合わせるためには、FORTRANとCの2つのプログラムが相互に各タイムサイクル毎に（境界条件などの）情報を交換することが必要になると思われる。

一方、それをソフトウェア工学的に考えた場合に、どのように接続アルゴリズムを実現するかが問題になる。たとえば、UNIXシステムでは一種のパイプ、ないしはフィルタでそれを実現することが都合いいかも知れないが、仕様の統一という面では困難を伴うものと考えられる。なぜならば、構造解析だけでなく、流体解析もそうであるが、カップリングに於いて重要なパラメータに幾何学的な座標系がある。座標系が異なる場合には、その座標系の変換を含めて非常に複雑な手続きを行わなければならない。そのようなメッセージ交換のインターフェイスは、単純に情報を交換することだけではなく、幾何形状まで合わせたインターフェイスでなければならないだろう。そして、そのようなインターフェイス・コードを開発するというような手間は非常にかかるものと思われる。さらに、開発されたコードは、それら2本の間を接続するであろう。それは、たとえば特別の圧力境界条件のみのメッセージをパッシングできるインターフェイスとして取り扱うことができるかも知れないが、そのようにして開発されたインターフェイス自身も第三者のその他のコードに関しての適用性は全くないものなることが予想される。

また、得られた解析結果をどのようにしてその妥当性を評価したらよいかという問題が残されている。特に工学的なレベルでの検証は非常に難しいと思われる。なぜならば、連成問題に関しては、通常ハードウェアレベルの試験をとり行うことは、予算、期間等の面から非常に難しいと思われる。また、実際にそのような試験を仮定した場合に、どのようにして測定したらいいのか、その正しい計画を準備することも非常に難しいと思われる。従って、計算結果、解析結果を検証するという事は今後重要な技術開発要件であると考えられる。

以上のように、ヴァーチャル・エンジニアリング・システムの技術開発要件として、ソフトウェア工学上の開発要件がある。次に連成解析、実際に計算された結果をどのように検証するか、検証するかという検証確認の技術を開発するという要件があると考えられる。後者については、モデル開発や理論研究の領域の課題と思われる。

## まとめ

ヴァーチャル・エンジニアリングの目的は、非常に多くの従来の経験、知見、さらに、コード、それらのソフトウェアを統合することを要求しているものと考えられた。一方、ヴァーチャル・エンジニアリング・システムに於いては、従来個別の専門家の領域として、相互に影響のない活動であるモデル開発、それらを接続しなければいけないという、非常に厳しい条件が出されていると考えられた。

特に、従来開発されてきた解析コードを接続すること、そのための技術を開発すること、さらに、それら個別のコードを接続して計算することの工学的妥当性を評価すること、この2つの技術を開発することが、ヴァーチャル・エンジニアリング・システム技術の領域として非常に重要なものであると判断された。以上のような技術を開発することにより、今後推進されるであろうヴァーチャル・エンジニアリングの各アプリケーションや、各プロジェクトに共通な基盤技術として取り入れることができると思われる。例えば、標準化されたコードのテストの手法、標準化された視覚化のエンジン、手法、コード、また、インターフェイス、データベース、それらについては標準化されることにより、個別のヴァーチャル・エンジニアリング・システムを開発するための基盤技術として、非常に効率的に利用することができると考えられる。

本研究においては、ヴァーチャル・エンジニアリングの技術をアッセンブリ技術として捉えている。そのため、個々の要素技術は別途調達し、それらを最適に組み合わせて、目標とする一つの専門システムを構築するということから、プロジェクトの管理も重要な課題であると考えられた。そのような観点からみた場合には、類似の世界としては建築現場に於ける工法が挙げられる。従って、ヴァーチャル・エンジニアリングの共通基盤技術として、効率的に応用できる技術の領域は、その工法の領域だと思われる。それは、システム・インテグレーションという一つの情報処理工学の分野に、モデル研究、さらにモデル開発という、問題向けの科学技術の分野を融合させたものと考えられる。共通基盤技術のうち、今後整備する事が求められるものとしては、データベース、グラフィックス、通信、さらに品質保証、ソフトウェアの管理、そのような諸々のソフトウェア・エンジニアリングの領域に属した課題が中心になるものと思われる。従って、本研究に於いては、当初シミュレーションという概念、より複雑なモデルを動的に組み合わせてシミュレーションをすることにより、従来のような個別の専門領域に於ける現象解明の知見とは異なった、より動的な複雑な現象の解明の知見が得られるという一つの期待から、逆にシステム工学として、一つの技術を整備するというテーマに変換しつつあると考えられる。なぜならば、前述の通りシミュレーションを執り行うという一つの知的な活動は、ヴァーチャル・エンジニアリング、ないしは類似したソフトウェア・エンジニアリングの領域ではなく、これは科学の領域であると考えられるためである。従って、従来開発されてきたシミュレーション・コードは、実現されたコードとしてみることはできるが、実際にはそれは数理モデルであると理解し、そこから統合型シミュレーションの可能性を検討すべきであると思われる。そのため、複数の数理モデルをどのように接続するかといった問題や、それらの妥当性に係わる研究が重要であると考えられた。

#### 今後の課題

ヴァーチャル・エンジニアリングの基盤技術としてのヴァーチャル・エンジニアリング・システム技術の研究を考えた場合に、今後の研究課題としておよそ以下が残されている。

ア. ソフトウェアの品質管理手法を確立する必要がある。

ソフトウェア管理の手法、品質管理の手法、次に標準化、そしてインターフェイス、グラフィックス等を充実するべきであると思われる。

イ. シミュレーション結果の検証・確認手法をソフト的に行う手法を開発する必要がある。

シミュレーションをどのように評価するか、またそのようなコードの組み合わせの妥当性をどのように評価するかという事に関しては、定量的に評価する手法等を研究すべきである。

ウ. 具体的なシステムの概念設計を行うべきである。

具体的問題解決を目指したシステムの概念設計を行い、より具体的な研究を推進すべきである。