

# 地層処分性能評価用解析システムの統合化研究(VI)

(動力炉・核燃料開発事業団 委託研究成果報告書)

技術資料		
開示区分	レポート No.	受領日
ズ	J1216 95-002	1995.7.20
この資料は技術管理室保存資料です 閲覧には技術資料閲覧票が必要です 動力炉・核燃料開発事業団 技術協力部技術管理室		

1995年3月

三菱重工業株式会社

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い合わ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

## 地層処分性能評価用解析システムの統合化研究 (VI)

房枝茂樹\*<sup>1</sup>、柳沢一郎\*<sup>1</sup>、鶴沢将行\*<sup>1</sup>  
松本弘之\*<sup>1</sup>、笠井雅夫\*<sup>1</sup>、石原義尚\*<sup>2</sup>  
土井英雄\*<sup>3</sup>、土井基尾\*<sup>4</sup>、江崎正弘\*<sup>4</sup>

### 要 旨

本研究では、ニアフィールドの複合現象を柔軟に解析できる連成解析用計算支援システム (CAPASA) の拡張と高度化を目的として、下記を実施した。

(1) 現象解析コードを連成解析システム上で連成させるための手順を、これまでの研究で得られた知見を基に、システム分析から連成実現にいたるステップ毎の手順としてまとめた。

(2) 今後の連成解析領域の拡張に関する作業効率と解析業務の効率向上のために、PLAN構築モジュールとPLAN実行モジュールの高機能化をプロトタイピングにより検討した。この結果、PLANにおけるデータフローならびに制御フローの表現について見直し、PLAN表現能力を拡張した高度化仕様を定めることができた。

(3) 複数の現象解析コードを連成した大型の解析を効率的に行うために、分散処理手法の導入について検討した。分散処理環境においては、PLANオブジェクト単位での並列処理により、解析の効率化が期待できるが、そのためにはオブジェクトの実行を制御するためのコントローラの実現が課題となる。

(4) ORIGEN2とHYDORGEOCHEMを対象とし、解析結果の登録、入出力、データのバージョン管理等の機能を拡張し、基本システムにおけるデータ管理機能の向上を図った。

---

本報告書は、三菱重工業株式会社が動力炉・核燃料開発事業団の委託により実施した研究の成果である。

契約番号：060D0383

事業団担当部課室および担当者：環境技術開発部地層処分開発室（牧野 仁史）

\*1：新技術開発部、\*2：原子力システム設計部、\*3：原子力応用技術部、

\*4：燃料サイクル技術部

## Integration Study (VI) for Performance Assessment System of Geological Disposal

Shigeki Fusaeda\*1, Ichiro Yanagisawa \*1, Masayuki Uzawa\*1,  
Hiroyuki Matumoto\*1, Masao Kasai\*1, Yoshinao Ishihara\*2,  
Hideo Doi\*3, Motoo Doi\*4, Masahiro Ezaki\*4

### ABSTRACT

In this study, the following tasks have been carried out to upgrade and expand the demonstration system of a fully integrated near-field performance assessment system (called CAPASA) .

- (1) Based on lessons from PLAN object's design so far, procedure for incorporating new codes into CAPASA system has been summarized into a series of step-by-step processes.
- (2) In order to improve the flexibility of PLAN object's design, the specification of PLAN construction and execution modules have been restudied based upon the discussions on semantics of PLAN networks. These two modules were re-designed according to the upgrade specification by reflecting the result of prototyping.
- (3) Application of distributed computation techniques has been discussed as a promising approach to improving computing efficiency. It was concluded that the concurrent execution of PLAN objects could be a practical approach of reducing the computing time of complicated PLAN.
- (4) The demonstration system has been upgraded from a practical point of view. Data storage function, data retrieval function, and version control function for data, process and PLANs have been re-designed by considering the use of ORIGEN2 and HYDROGEOCHEM codes on the demonstration system.

---

Work performed by Mitsubishi Heavy Industries, Ltd. under contract with Power Reactor and Nuclear Fuel Development Corporation.  
PNC Liaison: Waste Technology Development Division, Geological Isolation Technology  
Section, Hitoshi Makino.

\*1:Advanced Technology Development Department, \*2:Nuclear System Engineering Department  
\*3:Nuclear Application Technology Department, \*4:Nuclear Fuel Cycle Engineering Department

## 目次

1. はじめに	-----	1
2. 連成解析システム上での現象解析コード連成における作業内容・基本手順の知識化	-----	3
2.1 ORIGEN2とHYDROGEOCHEM連成実現における作業内容の整理	-----	4
2.1.1 平成5年度作業の整理	-----	4
2.1.2 連成解析モデルの検討	-----	16
2.2 基本手順の作成	-----	49
3. 連成解析システムの基本フレーム機能の拡充に関する検討	-----	52
3.1 PLAN構築／実行モジュールの高機能化に関する検討・設計	-----	52
3.1.1 高度化の検討	-----	52
3.1.2 プロトタイピング	-----	59
3.1.3 モジュールの設計	-----	77
3.1.4 まとめ	-----	80
3.2 分散処理環境に関する検討	-----	81
3.2.1 クライアント／サーバ形式での利用	-----	81
3.2.2 マルチCPUによる分散処理環境	-----	85
3.2.3 PNG計算機環境での運用の検討	-----	88
3.2.4 まとめ	-----	91
4. 連成解析システム上でのデータ等の管理機能向上のための環境整備	-----	92
4.1 機能仕様の検討	-----	92
4.1.1 連成解析システムでのデータの体系的な管理方法について	-----	92
4.1.2 ユーザ支援環境	-----	100
4.2 アプリケーションの設計・製作	-----	107
4.2.1 データベース仕様	-----	107
4.2.2 ソフトウェア仕様	-----	114
4.2.3 動作確認	-----	116
4.2.4 拡張性について	-----	121
5. あとがき	-----	123

添付資料1 連成解析システム上のデータの体系的な管理のための環境整備

ORIGEN2/Hydrogeochem Workbench 仕様書

## 図リスト

- 図2.1.1-1 平成5年度作業におけるORIGEN2とHYDROGEOCHEMの連成実現のための作業手順
- 図2.1.1-2 問題領域のオブジェクトダイアグラム
- 図2.1.1-3 連成解析フロー図
- 図2.1.1-4 解析領域のオブジェクトダイアグラム
- 図2.1.1-5 PLANネットワーク記述例 ORIGEN2-HYDROGEOCHEM-DECAY連成問題（拡散-移流、崩壊問題）
- 図2.1.1-6 PLANオブジェクト仕様例 Hgchem\_Link\_Manager AVS Module
- 図2.1.1-7 グローバルデータ
- 図2.1.1-8 データベーススキーマ ORIGEN2関連オブジェクト オブジェクト・ダイアグラム
- 図2.1.2-1 ニアフィールド領域におけるカップリング現象の模式図
- 図2.1.2-2 (1/2) ニアフィールドのシステム評価モデル概念
- 図2.1.2-2 (2/2) ニアフィールドのシステム評価モデル概念
- 図2.1.2-3 ニアフィールド・システム評価モデルのカップリング・ダイアグラム
- 図2.1.2-4 現象モデル/解析コード/データ分類の対応関係
- 図2.1.2-5 解析内容およびデータ内容の時間的/空間的分類
- 図2.1.2-6 輸送-化学-崩壊-水理の連成評価におけるデータの接続方式
- 図2.1.2-7 輸送-化学-崩壊-水理の連成評価における制御モジュールの接続フロー
- 図2.1.2-8 (1/3) 輸送-化学-崩壊-水理の連成評価におけるデータ・フロー  
（ガラス固化体の初期組成計算および溶解計算）
- 図2.1.2-8 (2/3) 輸送-化学-崩壊-水理の連成評価におけるデータ・フロー  
（緩衝材領域における拡散輸送計算）
- 図2.1.2-8 (3/3) 輸送-化学-崩壊-水理の連成評価におけるデータ・フロー  
（周辺岩盤領域における移流輸送計算）
- 図2.1.2-9 初期組成計算モジュールの機能
- 図2.1.2-10 ソース崩壊計算モジュールの機能
- 図2.1.2-11 溶解計算モジュールの機能
- 図2.1.2-12 水理計算モジュールの機能
- 図2.1.2-13 緩衝材領域での崩壊計算モジュールの機能
- 図2.1.2-14 緩衝材領域での輸送化学計算モジュールの機能
- 図2.1.2-15 周辺岩盤領域の崩壊計算モジュールの機能
- 図2.1.2-16 周辺岩盤領域での輸送化学計算モジュールの機能
- 図2.2-1 連成解析コード連成のための基本手順
- 図3.1.1-1 グローバル情報
- 図3.1.1-2 データフローを用いたデータの流れの制御
- 図3.1.1-3 複数の出力データを扱うデータフロー
- 図3.1.1-4 複数の入力データを扱うデータフロー
- 図3.1.1-5 PLAN表現例
- 図3.1.1-6 PLAN構築/実行/参照 概念図
- 図3.1.2-1 トリガフラッグの判定
- 図3.1.2-2 ループ状PLAN (1)
- 図3.1.2-3 ループ状PLAN (2)

- 図3.1.2-4 アクティブフラッグの設定
- 図3.1.2-5 PLANの実行
- 図3.1.2-6 サンプルPLAN
- 図3.1.2-7 サンプルPLANにおけるデータの流れ
- 図3.1.3-1 PLAN構築イメージ
- 図3.2.1-1 グループワークを支援する計算機環境
- 図3.2.1-2 クライアント/サーバ対応DBMS 検索フロー
- 図3.2.1-3 クライアント/サーバ形式システム 構築例
- 図3.2.2-1 分散処理ハードウェア環境
- 図3.2.2-2 分散環境におけるPLAN実行
- 図3.2.2-3 スケジューリングが必要となるPLAN例
- 図3.2.3-1 PNC計算機環境下でのCAPASA
- 図3.2.3-2 PNC計算機環境下でのデータアクセス手順
- 図4.1.1-1 データベースに格納するデータ
- 図4.1.2-1 Workbench 概略図
- 図4.1.2-2 データ概念図
- 図4.1.2-3 データのバージョン管理
- 図4.1.2-4 データベース構成
- 図4.1.2-5 全体制御機能
- 図4.2.1-1 データベーススキーマ マスターデータ
- 図4.2.1-2 データベーススキーマ 解析関連
- 図4.2.1-3 データベーススキーマ ORIGEN2関連
- 図4.2.1-4 データベーススキーマ Hydrogeochem関連
- 図4.2.1-5 インスタンス図 ORIGEN2関連
- 図4.2.1-6 インスタンス図 Hydrogeochem関連
- 図4.2.2-1 モジュール全体構成
- 図4.2.4-1 抽象クラスの適用例
- 図4.2.4-2 Hydrogeochemにおけるデータ変換方法

## 表リスト

表2.1.2-1 連成解析用データの特徴分類

表3.1.3-1 操作モード組み合わせパターン

表4.1.1-1 入力データ、解析結果の保存の考え方

表4.1.1-2 CAPASA/PCDEEにおけるバージョン管理の考え方

表4.1.2-1 出力テーブル種別

表4.1.2-2 出力テーブルタイプ



## 1. はじめに

我が国における高レベル放射性廃棄物地層処分計画を進める上での、当面の重要課題は、地層処分の概念について広く社会的な理解を得るために、適切に設計された地層処分システムの長期的な安全性を示すことである。

地層処分システムの長期的性能を示すためには、長期間に生起すると想定される種々の現象をモデル化し、これに基づいた計算コード及び解析管理システムを開発するとともに、適切な入力データを準備して、地層処分システムの長期挙動の予測解析を実施し、地層処分システムの性能評価を行うことが重要である。

地層処分システムでは、他の工学システムとは異なり、その安全性をシステムの試作により実証することが困難であるため、予測解析による間接的な安全性の実証に成らざるをえない。このため、直接的な実証に代わる検証・確認プロセスを性能評価の体系に組み込み、性能評価研究の妥当性を示す必要がある。また、モデル、コード、データの階層構造を体系的に管理・保存し、要求に応じて即座にトレースできるようにしておくことは、性能評価結果の妥当性・信頼性を保証するために必要不可欠である。

地層処分性能評価において考慮すべき現象は、熱、水理、力学、腐食、地球化学、輸送等多岐にわたり、且つ複合的／連鎖的な関連性を持っている（連成現象）。これらの現象や、現象間の相互関係についての理解は、性能評価に対する科学的根拠を保証していく上で重要である。このため、個々の現象或は幾つかの現象の連成を忠実に表現することができる現象解析モデル（Research Model）の開発が行われている。さらに、地下水シナリオ等で考慮すべき長期的かつ複合的な現象群をモデル化して、地層処分システム全体の性能を定量化し、適切な指標との比較によってその妥当性を示すという地層処分システムの性能評価の最終目標のためには、地層処分システムに内包される多様な現象の組合せを包括的に評価できるシステム評価モデル（Assessment Model）の開発が不可欠である。

そのようなシステム評価モデルを開発していく過程の中で、現象解析モデルの開発成果を生かしつつ地層処分システムで生じ得る多様な連成現象を体系的に評価していける計算支援システム（連成解析システム）の開発が重要となる。これは、地下水シナリオ等に対応して、現象解析モデルや試験データ等を整備・統合し、これらを体系的に管理することにより、地層処分システムの総合的かつ詳細な評価を行うものである。連成解析システムを用いることにより、性能評価上関心の高い支配的な現象を整理し、安全性の裕度を確保しながら、現象解析モデルを統合化・簡素化することによって、信頼性の高いシステム評価モデルを開発していくことができる。

本研究では、これまでに、現象解析モデルの体系的な管理と結合を行う連成解析システムの開発を通じて、最終的なシステム評価モデルの中核とする性能評価用の統合化システムの拡充を推進するため、考慮すべき現象および使用するコードの調査、検証・確認の手法検討を一部実施し、モデル、コード、データの体系的な管理の手法検討、現象解析モデルの連成アルゴリズムの検討とそれに基づく連成解析システムの基本フレームの構築を行って

きた。

今年度の研究では、これまでの成果に基づき、下記項目を実施し、連成解析システムの基本システムの拡張と高度化を図った。

- (1)現象解析コード連成における作業内容・基本的手順の知識化
- (2)連成解析システムの基本フレーム機能の充実に関する検討
  - ①PLAN構築、PLAN実行モジュールの高機能化に関する検討・設計
  - ②分散処理に関する検討
- (3)連成解析システム上でのデータ等の管理機能向上のための環境整備

## 2. 連成解析システム上での現象解析コード連成における作業内容・基本手順の知識化

連成解析システムの能力は、システムに取り込まれた現象解析コードの数や解析に使用可能なデータ量に依存する。すなわち、多くの現象解析コードとそれらの連成に必要なユーティリティ・プログラムを取り込むことにより、連成解析領域を拡張することが可能となる。このため、連成解析システムの枠組みが固まった次のステップとしては、連成解析領域を拡張するために、様々な現象解析コード間の連成を連成解析システム上で実現していく必要がある。

新たな連成解析の実現を、できるだけ効率よく、容易に行えるようにすることが連成解析支援システム（CAPASA/PCDEE）に対する重要な機能要件の一つとなっている。このため、システムの開発には、分析、設計、実装の各段階でオブジェクト指向アプローチを採用し、ソフトウェアの再利用性を高めるものとしている。

しかしながら、連成解析システムの活用を図るためには、システム側での配慮だけでなく、システム・ユーザ側の能力の向上も必要である。近い将来には、連成解析支援システムの拡張は、性能評価解析の実施者が主体となって行うべきと考えられるが、平成5年度の作業では、分析、設計、実装の各段階で、問題領域の専門家の他に、システム開発のエキスパートが大きく関与している。連成解析システムを有効に活用して行くためには問題領域専門家が、ソフトウェア開発のエキスパートにあまり依存せずに、製作仕様を定められるようにすべきである。

全ての段階を問題領域の専門家のみが行えるようにすることは困難ではあるが、上流側の分析作業を問題領域専門家が主体となって実施し、設計、実装をソフトウェア開発の専門家に任せるのが、適切な作業分担であろう。

このためには、あらかじめ連成実現までの作業手順、分担を明確に定めておくべきである。ここでは、これまでの研究における現象解析コードの連成解析システム上での連成実現に関する知見を、特に問題領域・解析領域のシステム分析から連成実現に至る各作業の内容及び基本的な手順としてまとめた。

まず、平成5年度の研究において実施した ORIGEN2 と HYDROGEOCHEM の連成実現のための作業を整理した。平成5年度では、限定された範囲ではあるが、連成モデルの検討から PLAN オブジェクトの実装までの一連の作業をおこなっているため、その作業手順が、基本手順の検討の原型となると考える。

次に、分析作業の内容をより詳細に定めるため、ニアフィールド連成解析における連成モデルのより一般的な分析を実施した。この分析結果は連成モデル検討の一連のアプローチを示すものである。また、3章における基本フレームの高度化仕様の検討にも用いられる。

最後に、3章での検討結果も反映した形で、連成実現までの基本手順をステップ毎にまとめた。

## 2. 1 ORIGEN2 と HYDROGEOCHEM 連成実現における作業内容の整理

ここでは連成実現に必要な基本手順を定めるために、ORIGEN2 と HYDROGEOCHEM の連成を対象とし、連成実現までの作業内容の整理を行った。

まず、平成5年度に実施した ORIGEN2 と HYDROGEOCHEM の連成実現のための作業について述べる。次に、分析段階での作業内容をより詳細に定めるために実施したニアフィールド連成解析における連成モデルのより一般的な分析の結果について述べる。

### 2. 1. 1 平成5年度作業の整理

平成5年度作業における ORIGEN2 と HYDROGEOCHEM の連成実現までの作業手順を以下に記述する。平成5年度の作業は、連成解析システム の概念を固めながら実施した作業であり、かならずしも最適な手順でアウトプットをまとめられたわけではなかったが、連成問題の設定から、PLANオブジェクトの実装、動作確認までの一連の作業を実施しており、そこでの問題点も整理しながら、作業の流れをまとめる。

平成5年度における連成実現までの作業手順を図2. 1. 1-1に示す。各ステップでの作業の概要とアウトプットについて以下に述べる。

#### (1) 問題領域分析

##### ①問題領域の設定

問題領域分析では、まず、分析対象をニアフィールド問題と定めた。

分析の範囲を明確にするために、まず、問題領域を明確にする必要があった。

##### ②オブジェクト分析

次に、ニアフィールド解析に必要となるオブジェクトの同定(概念分析)をおこなった。概念分析の目的は、ニアフィールド問題を解くときに必要となるデータ構造をできるだけ実世界にマッチした形で整理することである。多くのソフトウェア開発においてデータ構造は最も変化しにくいものであり(手続きに比べた場合に安定と考えられる)、データ構造をさきに定めておくことが、長期的に安定なシステム開発のキーとなると考えられる。

分析は、オブジェクト指向分析を用い、以下の手順で行った。

ステップ1 問題領域専門家とシステム開発者の協議により、対象領域における概念を広い出す。

ステップ2 概念間 の関係を定める。特に、上位クラス-下位クラス、構成関係、さらに関連などを、実世界での関係を素直にモデル化するべく定めて行く。

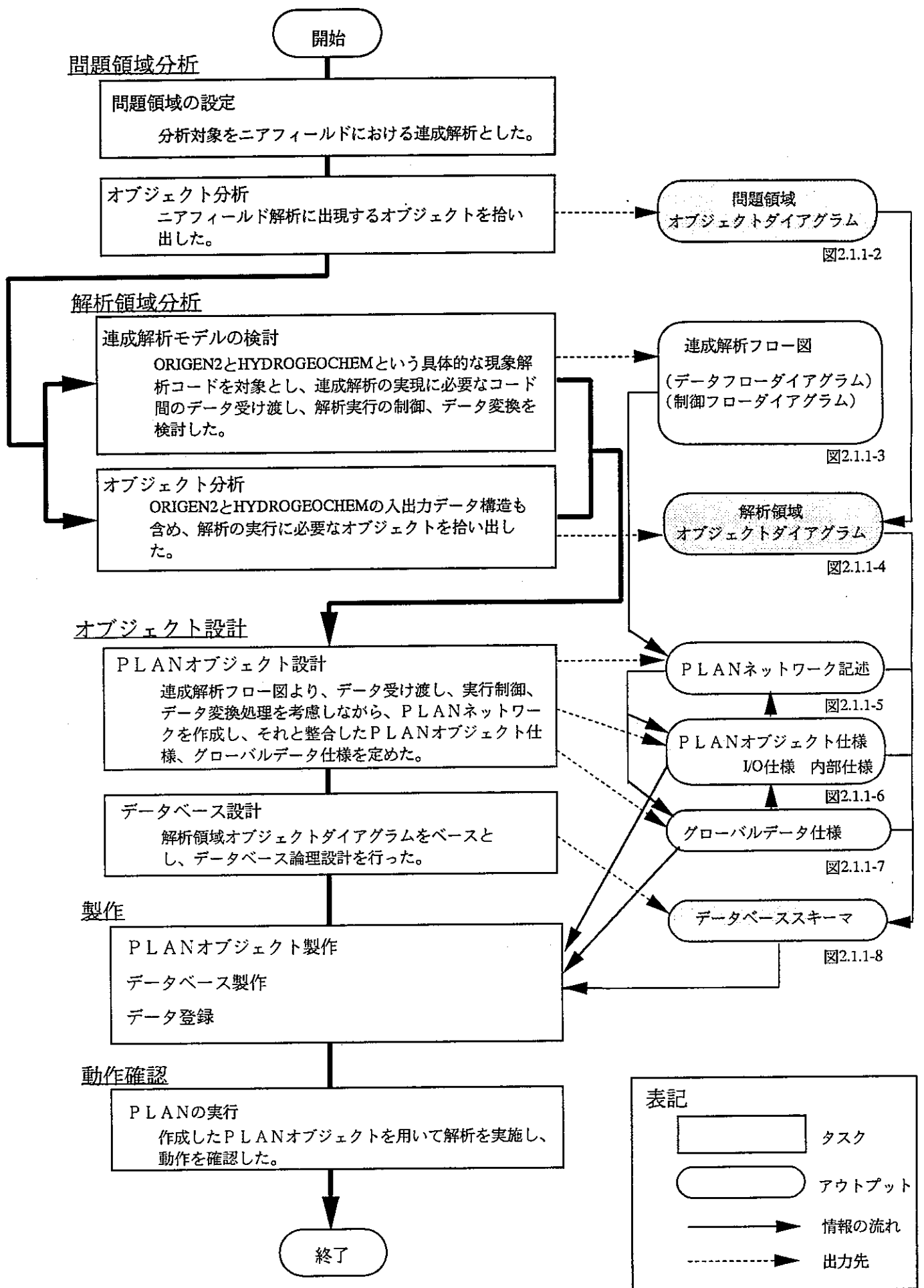


図2.1.1-1 平成5年度作業における ORIGEN2 とHYDROGEOCHEM の連成実現のための作業手順

ステップ3 抽出された概念に対して、その概念を特徴付ける属性を抽出する。

ステップ4 さらに、可能であれば、各概念に特有の操作を定義して行く。

オブジェクト分析のアウトプットは、オブジェクトダイアグラムである。

図2. 1. 1-2に平成5年度の問題領域分析により作成されたオブジェクトダイアグラムを示す。

問題領域分析のアウトプットであるオブジェクトダイアグラムは、解析領域分析でのオブジェクト分析において、ORIGEN2 と HYDROGEOCHEM の入出力データ構造を議論する際の参考となった。少なくとも、直接 ORIGEN2 あるいは HYDROGEOCHEM のデータ構造を議論し始めた場合に比べ、データ構造をより汎用的にできたはずである。

問題領域分析で得られたオブジェクトの、最終的な使い道は、CAPASAにおける解析/設計情報データベースを構築する際のデータ構造を与えることになろう。あらゆる現象解析コード間でのデータ受渡しを行うための、中間的なデータ構造を無理なく定めるためには、オブジェクト指向分析が有効な手法となろう。

問題領域分析のメリットとして、直接にアウトプットとしては見えないが、問題領域専門家とシステム分析者との間でテクニカルな概念に関して共通の認識が得られ、後の議論が効率に進められるようになったことも指摘しておきたい。

なお、問題領域分析では、特定の解析コードには言及しないことが、問題領域で扱うデータ構造を特定の解析コードに依存することなく定めるために重要なポイントである。

## (2) 解析領域分析

解析領域分析では、特定の解析コードを用いた連成モデルの検討と、オブジェクト分析を行い、PLANオブジェクトの設計のための情報として連成解析のフロー図と解析領域に対するオブジェクトダイアグラムを定めた。

### ①連成解析モデルの検討

連成解析モデルの検討では、ORIGEN2 と HYDROGEOCHEM を用いた連成解析の手順を検討し、そこでのデータ受渡し、解析実行の制御、データ変換処理を定め、連成解析フロー図を作成した。

図2. 1. 1-3に平成5年度作業で作成した連成解析フロー図の1つを示す。このようなフロー図は、連成解析に必要な下記の情報を整理するために、実現したい連成モデルのそれぞれに対して作成する。

・プロセス

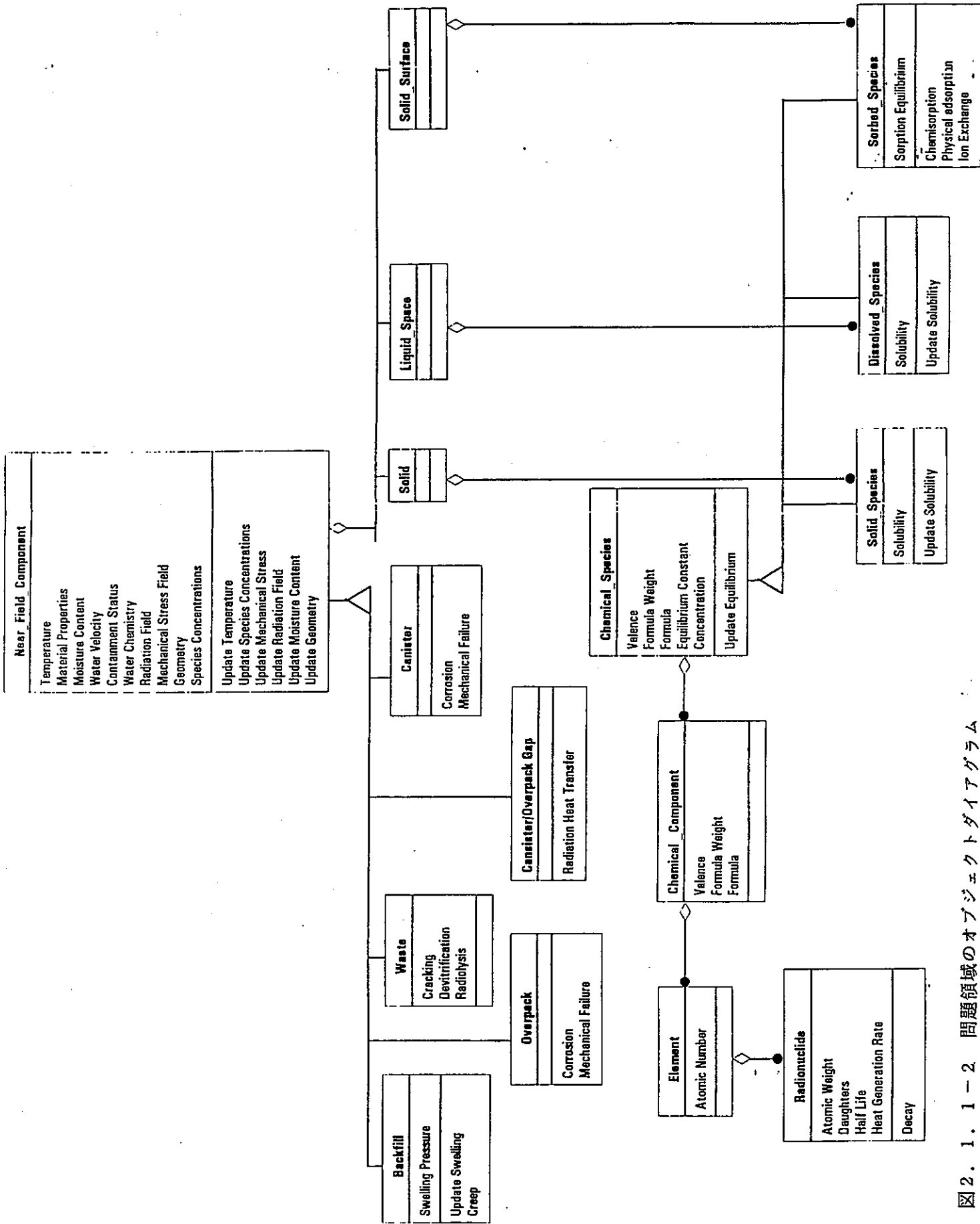


図 2. 1. 1 - 2. 問題領域のオブジェクトダイアグラム

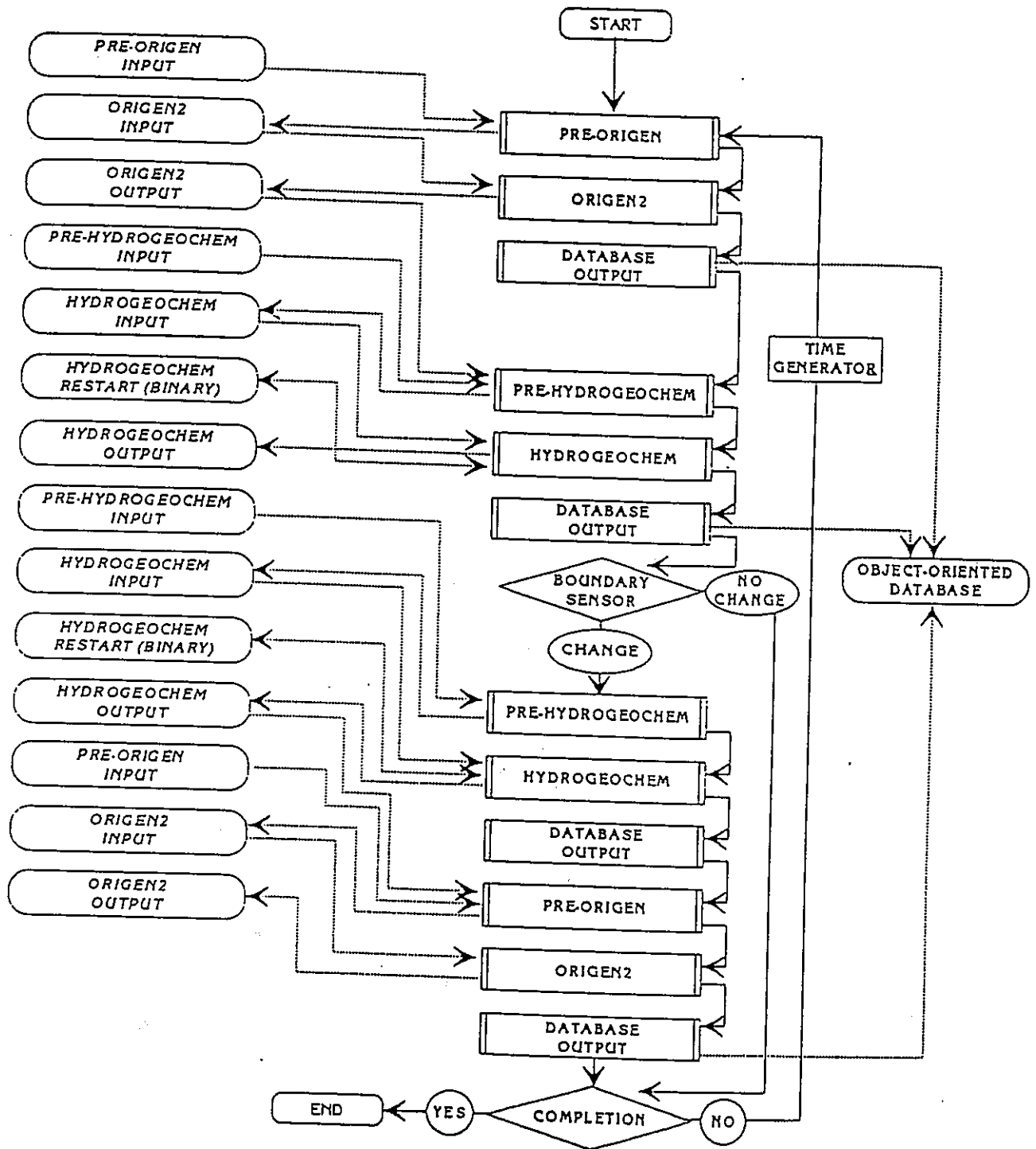


図 2. 1. 1-3 連成解析フロー図



- ・データフロー
- ・制御フロー

ここでの分析結果が、CAPASA/PCDEEにおけるPLANオブジェクトの設計の要求仕様となった。

なお、連成モデルの汎用性を保守するためには、連成解析モデルの検討において、できるだけ広い領域を分析してから、共通に利用できるプロセスや、データフローや制御フローにおける類似性、制約を拾い出すことが重要である。

なお、この観点からの知見を整理するために、連成解析モデルの検討については、今年度に新たな検討を実施した（2. 1. 2参照）。

## ②オブジェクト分析

ここでは、現象解析コードの入出力データも考慮に入れて、解析に出現するオブジェクトを拾い出した。この分析の結果は、解析領域のオブジェクトダイアグラムとしてまとめられた。図2. 1. 1-4に平成5年度に作成した解析領域のオブジェクトダイアグラムの一部を示す。

このオブジェクトダイアグラムは、データベース構造の設計に用いられる。

## (3) オブジェクト設計

次に、(1)(2)項での分析結果をもとに、PLANオブジェクトとデータベースの設計を行った。

### ①PLANオブジェクト設計

連成解析フロー図に表現された連成計算のデータや制御の流れから、どのようなプロセスをPLANオブジェクト（PLANの表現要素となるオブジェクト）として実現するかを定め、各オブジェクトの内部仕様や、グローバル情報（複数のPLANオブジェクトにまたがる情報）の仕様を定めた。

ここでの設計プロセスは、実装方式を念頭に置きながら、オブジェクト粒度や汎用性を考慮しながら行う必要がある。

ここでのアウトプットは以下の通りである。

- ・PLANネットワーク記述（図2. 1. 1-5）
- ・PLANオブジェクト仕様（図2. 1. 1-6）
- ・グローバルデータ仕様（図2. 1. 1-7）

なお、これらの3つのアウトプットの作成には、相互の整合性を保証するための繰り返し作業が必要である。

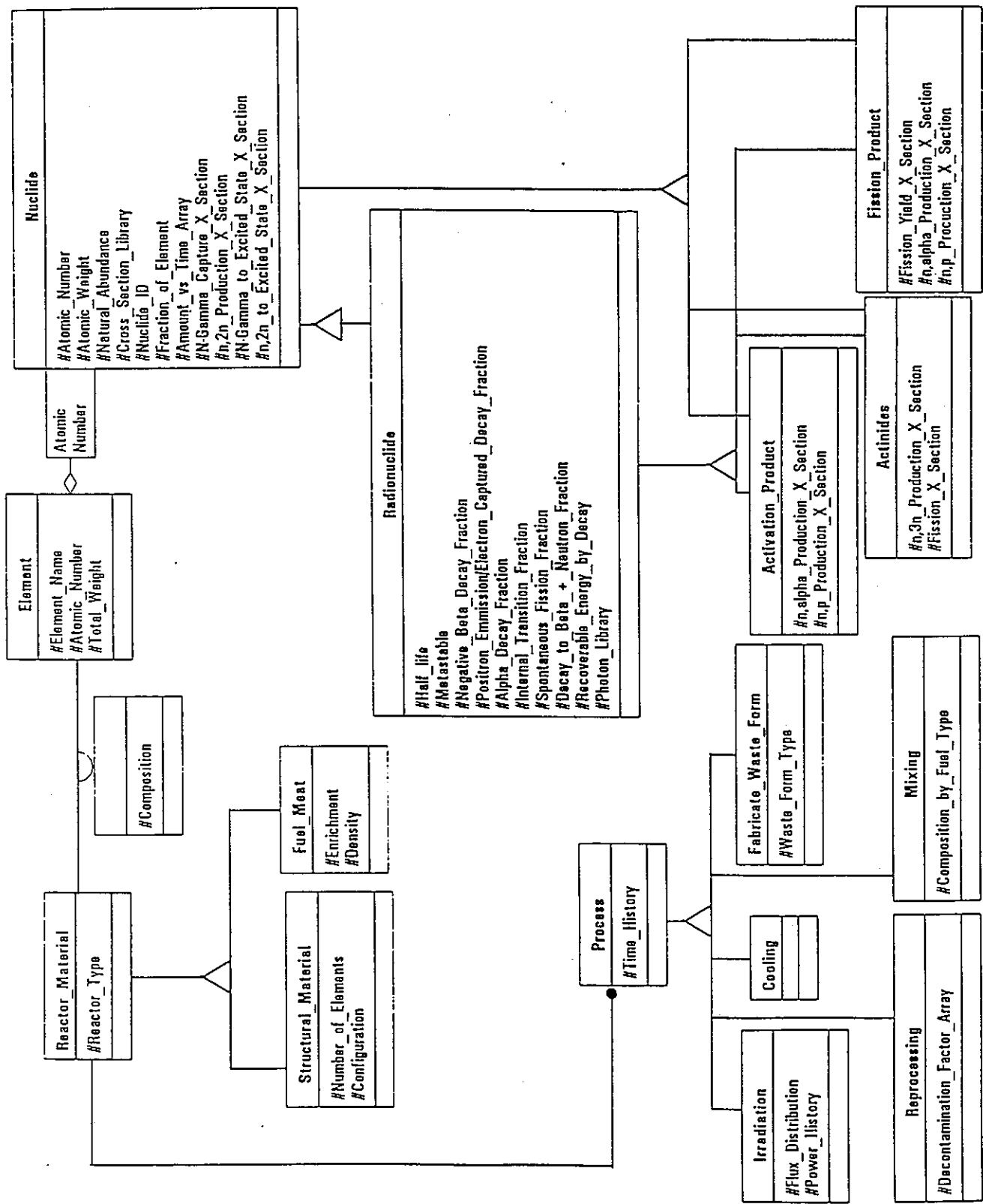


図 2. 1. 1-4 解析領域のオブジェクトダイアグラム

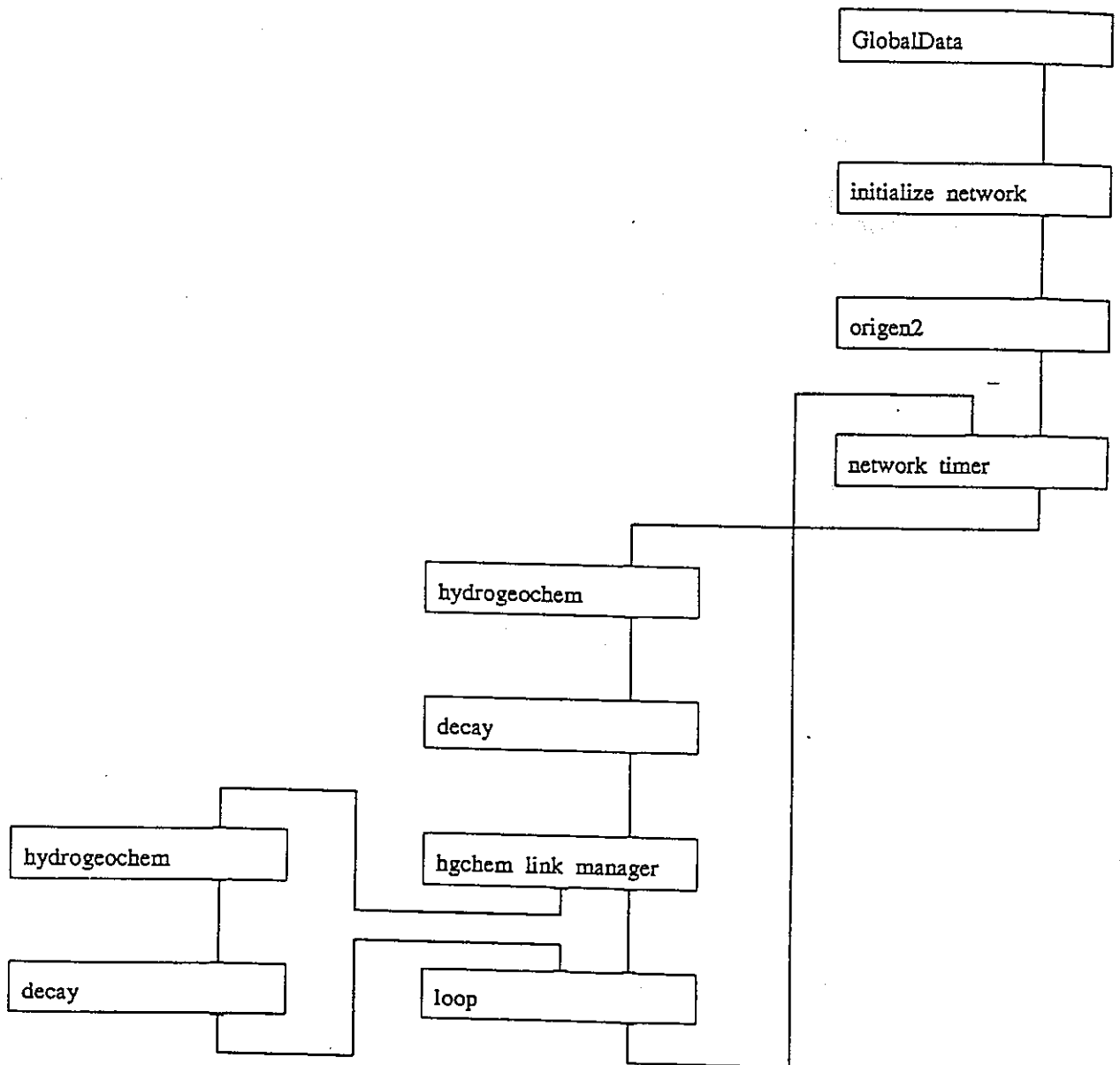


図 2. 1. 1-5 PLANネットワーク記述例

ORIGEN-HYDROGEOCHEM-DECAY連成問題 (拡散-移流、崩壊問題)

AVS Hydrogeochem Link Manager Module	
<b>Arguments:</b>	
Filename	Name of the initialization file
Instance1	Name of the first instance of hydrogeochem
Instance2	Name of the second instance of hydrogeochem
<b>AVS Integer Ports</b>	
input	Data input port
main_net	Data output port to main network
sub_net	Data output port to second hydrogeochem

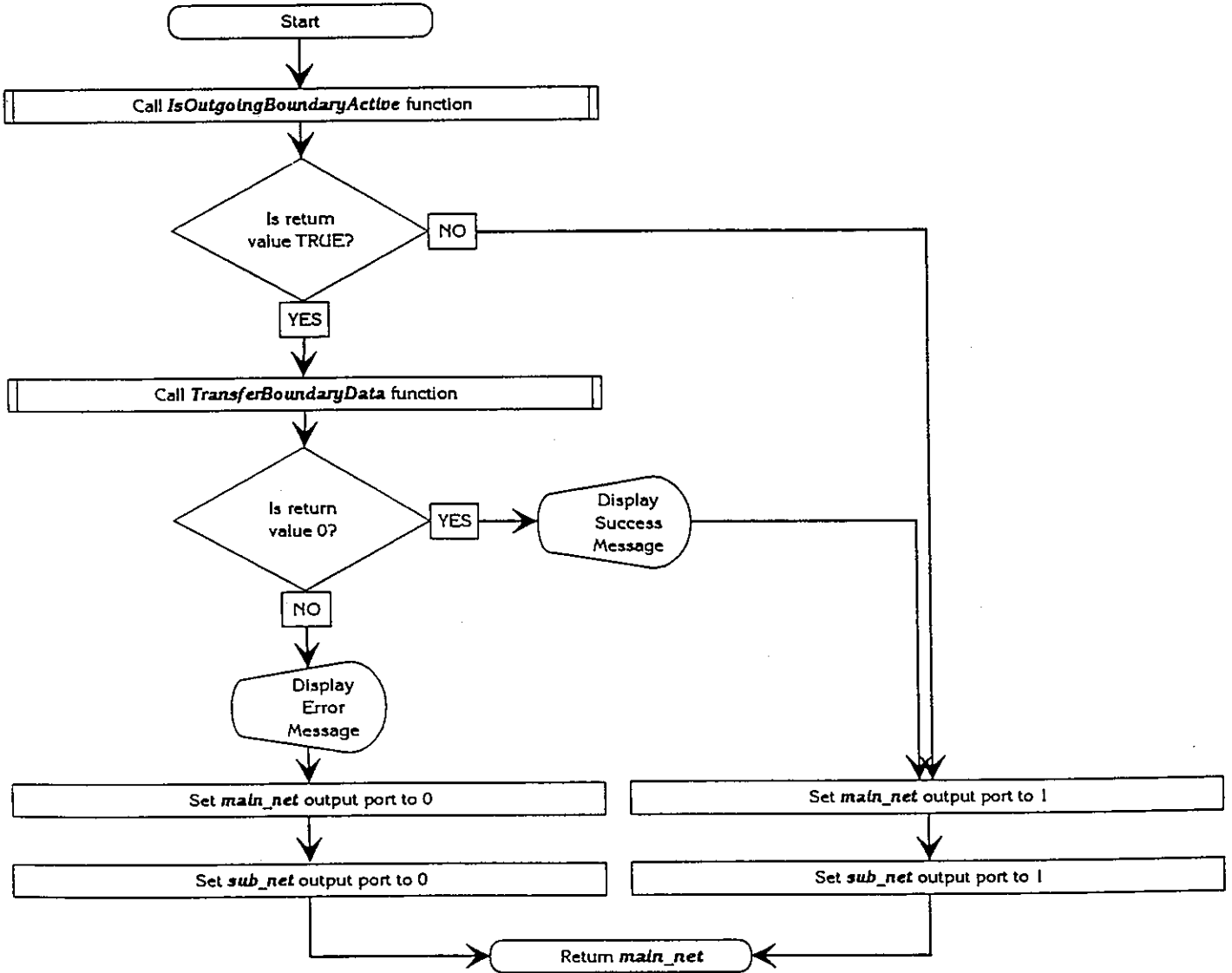


図 2. 1. 1 - 6 PLANオブジェクト仕様例

Hgchem\_Link\_Manager AVS Module

```

[problem]
hydrogeochem=hgchem1,hgchem2
decay=decay1,decay2
scale factor=100000.0

[hgchem1.hydrogeochem]
ascii input file=hgchem1.in
outgoing boundary=11,12
active=1

[hgchem1.hydrogeochem.components]
SODIUM=11,0
CALCIUM=20,0
HYDROGEN=1,0
ELECTRON=8,0
CHLORIDE=17,0
BICARBONAT=0,0
MAGNESIUM=12,0
POTASSIUM=19,0
SULFATE=0,0
AMERICIUM=95,2
NEPTUNIUM=93,1
PALLADIUM=46,1
PLUTONIUM=94,2
RADIUM=88,0
SELENIUM=34,1
TECHNETIUM=43,1
THORIUM=90,2
TIN=50,1
URANIUM=92,5
ZIRCONIUM=40,1
ACTINIUM=89,1
PROTOACTIN=91,1
CESIUM=55,1

[hgchem1.hydrogeochem.radionuclides]
AMERICIUM=952430,952410
NEPTUNIUM=932370
PALLADIUM=461070
PLUTONIUM=942390,942400
SELENIUM=340790
TECHNETIUM=430990
THORIUM=902320,902300
TIN=501260
URANIUM=922380,922360,922350,922340,922330
ZIRCONIUM=400930
ACTINIUM=892270
PROTOACTIN=912310
CESIUM=551350

[hgchem2.hydrogeochem]
ascii input file=hgchem2.in
incoming boundary=1,2
active=1

[hgchem2.hydrogeochem.components]
SODIUM=11,0
CALCIUM=20,0
HYDROGEN=1,0
ELECTRON=8,0
CHLORIDE=17,0
BICARBONAT=0,0
MAGNESIUM=12,0
POTASSIUM=19,0
SULFATE=0,0
AMERICIUM=95,2
NEPTUNIUM=93,1
PALLADIUM=46,1
PLUTONIUM=94,2
RADIUM=88,0
SELENIUM=34,1
TECHNETIUM=43,1
THORIUM=90,2
TIN=50,1
URANIUM=92,5
ZIRCONIUM=40,1
ACTINIUM=89,1
PROTOACTIN=91,1
CESIUM=55,1

```

```

; problem section
; hydrogeochem instances
; decay instances
; multiplier to convert time to years in this problem

; First hydrogeochem instance block

; Format is chemical=atomic number,number of radionuclides
; Sodium : atomic number 11 and 0 radionuclides
; Calcium : atomic number 20 and 0 radionuclides
; Hydrogen : atomic number 1 and 0 radionuclides
; Oxygen : atomic number 8 and 0 radionuclides
; Chlorine : atomic number 17 and 0 radionuclides
; Not an element
; Magnesium : atomic number 12 and 0 radionuclides
; Potassium : atomic number 19 and 0 radionuclides
; Not an element
; Americium : atomic number 95 and 2 radionuclides
; Neptunium : atomic number 93 and 1 radionuclide
; Palladium : atomic number 46 and 1 radionuclide
; Plutonium : atomic number 94 and 2 radionuclides
; Radium : atomic number 88 and 0 radionuclides
; Selenium : atomic number 34 and 1 radionuclide
; Technetium : atomic number 43 and 1 radionuclide
; Thorium : atomic number 90 and 2 radionuclides
; Tin : atomic number 50 and 1 radionuclide
; Uranium : atomic number 92 and 5 radionuclides
; Zirconium : atomic number 40 and 1 radionuclide
; Actinium : atomic number 89 and 1 radionuclide
; Protactinium : atomic number 91 and 1 radionuclide
; Cesium : atomic number 55 and 1 radionuclide

; Americium : atomic number 95 and 2 radionuclides
; Neptunium : atomic number 93 and 1 radionuclide
; Palladium : atomic number 46 and 1 radionuclide
; Plutonium : atomic number 94 and 2 radionuclides
; Selenium : atomic number 34 and 1 radionuclide
; Technetium : atomic number 43 and 1 radionuclide
; Thorium : atomic number 90 and 2 radionuclides
; Tin : atomic number 50 and 1 radionuclide
; Uranium : atomic number 92 and 5 radionuclides
; Zirconium : atomic number 40 and 1 radionuclide
; Actinium : atomic number 89 and 1 radionuclide
; Protactinium : atomic number 91 and 1 radionuclide
; Cesium : atomic number 55 and 1 radionuclide

; Second hydrogeochem instance block

; Format is chemical=atomic number,number of radionuclides
; Sodium : atomic number 11 and 0 radionuclides
; Calcium : atomic number 20 and 0 radionuclides
; Hydrogen : atomic number 1 and 0 radionuclides
; Oxygen : atomic number 8 and 0 radionuclides
; Chlorine : atomic number 17 and 0 radionuclides
; Not an element
; Magnesium : atomic number 12 and 0 radionuclides
; Potassium : atomic number 19 and 0 radionuclides
; Not an element
; Americium : atomic number 95 and 2 radionuclides
; Neptunium : atomic number 93 and 1 radionuclide
; Palladium : atomic number 46 and 1 radionuclide
; Plutonium : atomic number 94 and 2 radionuclides
; Radium : atomic number 88 and 0 radionuclides
; Selenium : atomic number 34 and 1 radionuclide
; Technetium : atomic number 43 and 1 radionuclide
; Thorium : atomic number 90 and 2 radionuclides
; Tin : atomic number 50 and 1 radionuclide
; Uranium : atomic number 92 and 5 radionuclides
; Zirconium : atomic number 40 and 1 radionuclide
; Actinium : atomic number 89 and 1 radionuclide
; Protactinium : atomic number 91 and 1 radionuclide
; Cesium : atomic number 55 and 1 radionuclide

```

また、アウトプットの内容も、PLANネットワーク記述仕様により変わってくるものである。平成5年度におけるPLANネットワーク記述仕様は、リンクは制御フローのみを表現し、ノードはプロセスを表現するという大きな制約に基づくものであった。このため、プロセス間のデータフローがグローバル情報として扱われたため、グローバルデータ仕様の設定がPLANオブジェクト設計の中で大きな比重を持つことになった。

PLANオブジェクト仕様は、対象がプロセスであることから、オブジェクトへのI/O仕様と内部処理の記述により行った。

## ②データベース設計

データベース設計では、解析領域オブジェクトダイアグラムとPLANオブジェクトの設計結果をもとにデータベーススキーマ（データ構造）を定めた。図2. 1. 1-8に平成5年度に作成したデータベーススキーマ例を示す。

## (4) 製作、動作確認

オブジェクト設計に従い、PLANオブジェクトならびにデータベースを作成し、データベースへ必要なデータの登録を行った。それらのオブジェクトを用いてPLANを構築し、PLAN実行モジュール上で実行させ、動作を確認した。

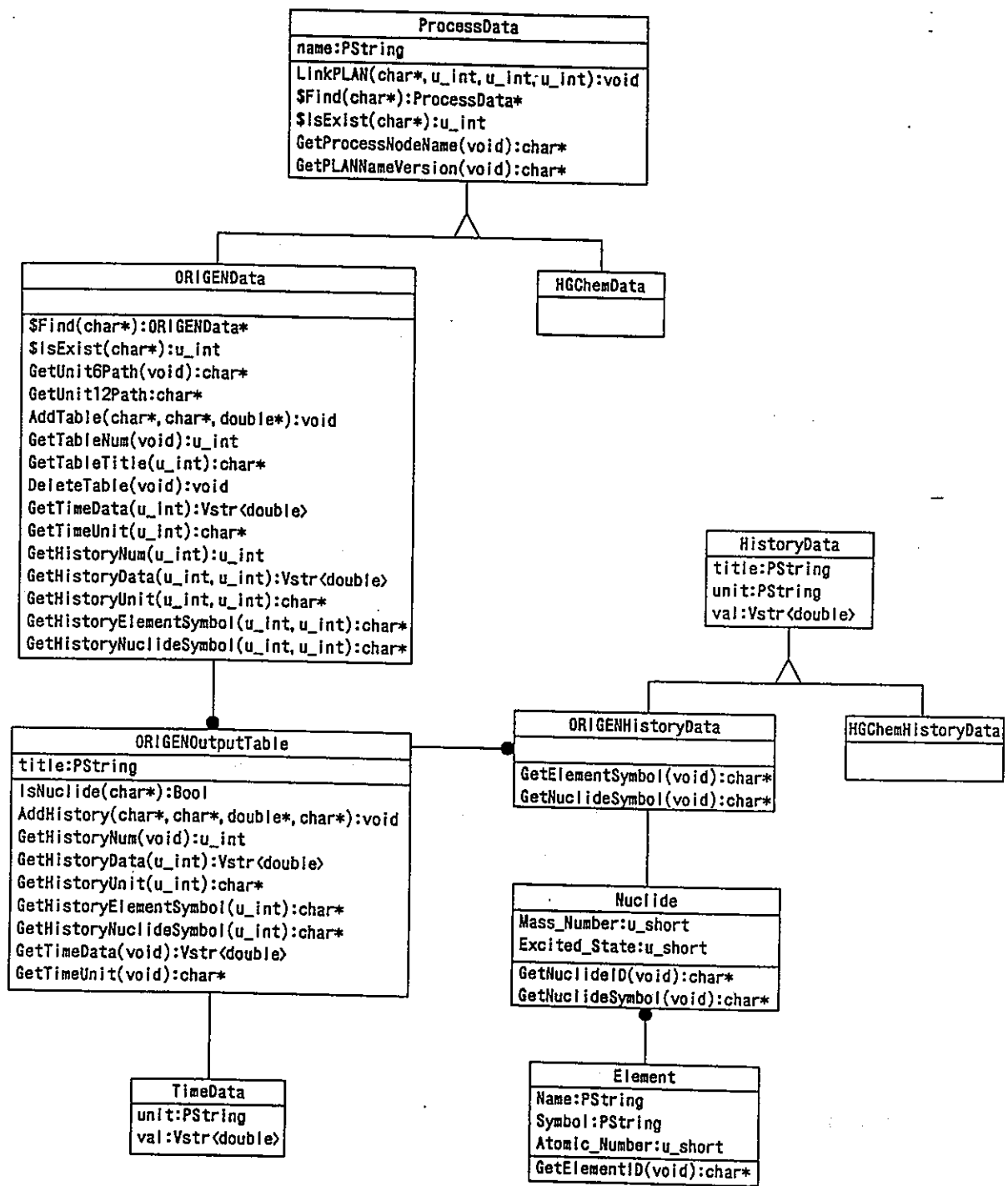


図 2. 1. 1-8 データベーススキーマ

ORIGEN2関連オブジェクト オブジェクト・ダイアグラム

## 2.1.2 連成解析モデルの検討

サイトを特定しない概括的性能評価研究で使われているニアフィールドにおける性能評価解析は、図2.1.2-1に示すように、ガラス固化体に接近してくる地下水の化学特性の変化プロセスと、ガラス固化体から浸出・溶解した核種の移行プロセスが中核となっている。

このような地下水シナリオに基づいて、平成3年度研究においては、処分場の性能に何らかの影響を及ぼすと考えられる現象メカニズムを、核的、力学的、熱的、化学的、および物質移動のプロセスに区分して同定し、図2.1.2-2および図2.1.2-3に示すように、それらのハイアラキと関係をまとめた。さらに、連成解析実行のため、各現象メカニズムに実体を与える現象解析コードおよび必要なデータについて、図2.1.2-4に示すように、各プロセスのハイアラキと対応づけて整理した。

また、解析対象のプロセスを表現するために必要なデータは、計算モデルへの入力データ、および計算モデルの代替となるデータである。これらのデータを大別すると以下のようなになる。

- ①伝熱特性 (Material Heat Transport Properties)
- ②力学特性 (Material Mechanical Properties)
- ③化学的データ (熱力学データ) (Chemical-Thermodynamic Data)
- ④浸入地下水データ (Inlet Water Chemistry)
- ⑤燃料の燃焼/冷却/装荷及びインベントリ・データ  
(Burnup/Cooling/Loading of Fuel&Waste)
- ⑥設計データ (材質, 寸法) (Design Data-Materials, Dimension)
- ⑦水理物性 (Material Hydraulic Properties)
- ⑧気候データ (Climatological)
- ⑨腐食データ (Corrosion : Bulk and local)
- ⑩地質構造と物性データ (Geologic Structure and Properties)

人工バリアの物理・化学的特性データ

- ⑪放射線/材質影響データ (Radiation/Material Effects)
- ⑫ガラス固化体の実験データ (Glass Testing)
- ⑬崩壊定数および崩壊系列 (Nuclear Decay Constants & Products)

ここで整理した現象メカニズム/解析コード/データのハイアラキ、および現象メカニズム間の相互依存関係は、平面的に表現されているため、各解析コード間の制御関



係およびデータの受渡し関係が、必ずしも明確に表現されていない。このため、連成解析を実現するためには、解析コードおよびデータの連成関係について、時間的・空間的な接続方法について検討する必要がある。

平成5年度に実施した連成モデルの検討作業は、HYDROGEOCHEMとORIGENの2つのコードのみを対象としているために、分析されたオブジェクト・ダイアグラム、連成解析フロー、PLANネットワーク等は、HYDROGEOCHEMとORIGENの2つのコードの連成だけに特定されており、他の現象解析コードに対する汎用性がほとんどない。このため、ニアフィールド連成解析における連成モデルのより一般的な分析を行い、連成モデルを検討するための一連のアプローチを明確にしておく必要がある。

そのため、解析コードおよびデータの連成関係について、時間的・空間的な特徴を踏まえた上で、コードとデータをどの様に接続していくかをまとめることによって、汎用的な連成モデルを検討するためのアプローチを示すことができると考えている。

ここでは、解析コードおよびデータの具体的な接続関係をまとめるために、以下の項目について具体的な検討を行った。

#### ①解析内容／データの分類

連成解析に際して、各現象解析コードで実施する解析内容および連成に使用するデータについて、時間的特徴と空間的特徴の観点から分類して整理した。

#### ②データ接続の分類

連成に使用するデータの空間的特徴から、解析コードへ入力するデータの特徴を踏まえて、各データの接続方法について整理した。

#### ③制御モジュールの接続関係の分析

各現象解析コードの連成を実現するため、制御モジュールの接続関係について、データの受渡しを考慮して分析した。

#### ④データ・フローの分析

データの接続関係および制御モジュールの接続関係に基づいて、輸送－化学－崩壊－水理の連成解析におけるデータ・フローを整理した。

#### ⑤計算制御モジュールの機能分析

データ・フローに基づいて、解析コードの実行を制御する各制御モジュールの機能について、必要なデータと関連づけて処理フローを分析した。

# Coupling Processes on Base Case Scenario

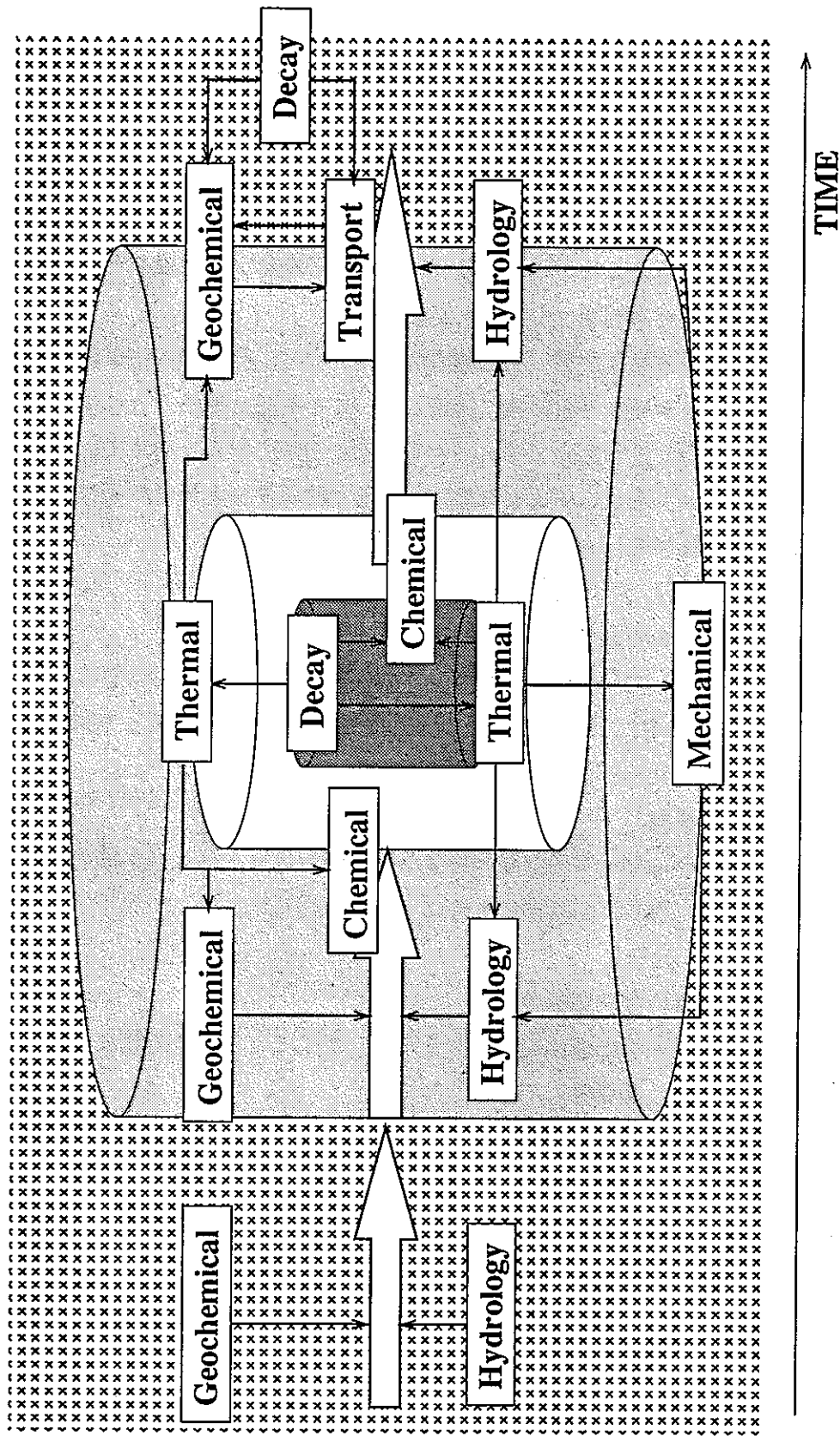


図2.1.2-1 ニアフィールド領域におけるカップリング現象の様式図

# HIERARCHY OF PROCESS IN NEAR FIELD

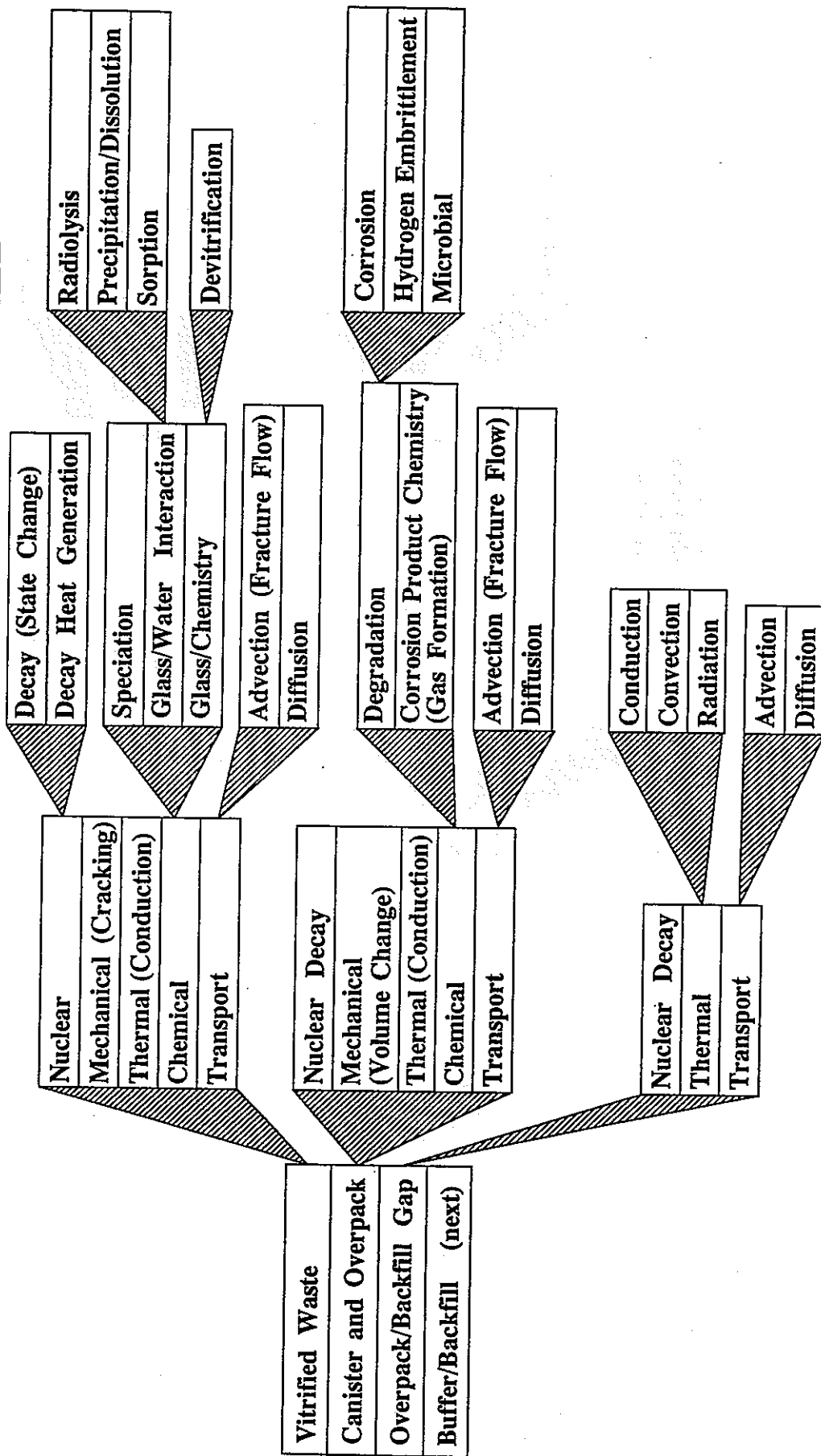


図 2.1.2-2 (1/2) ニアフィールドのシステム評価モデル概念

# HIERARCHY OF PROCESS IN NEAR FIELD (cont.)

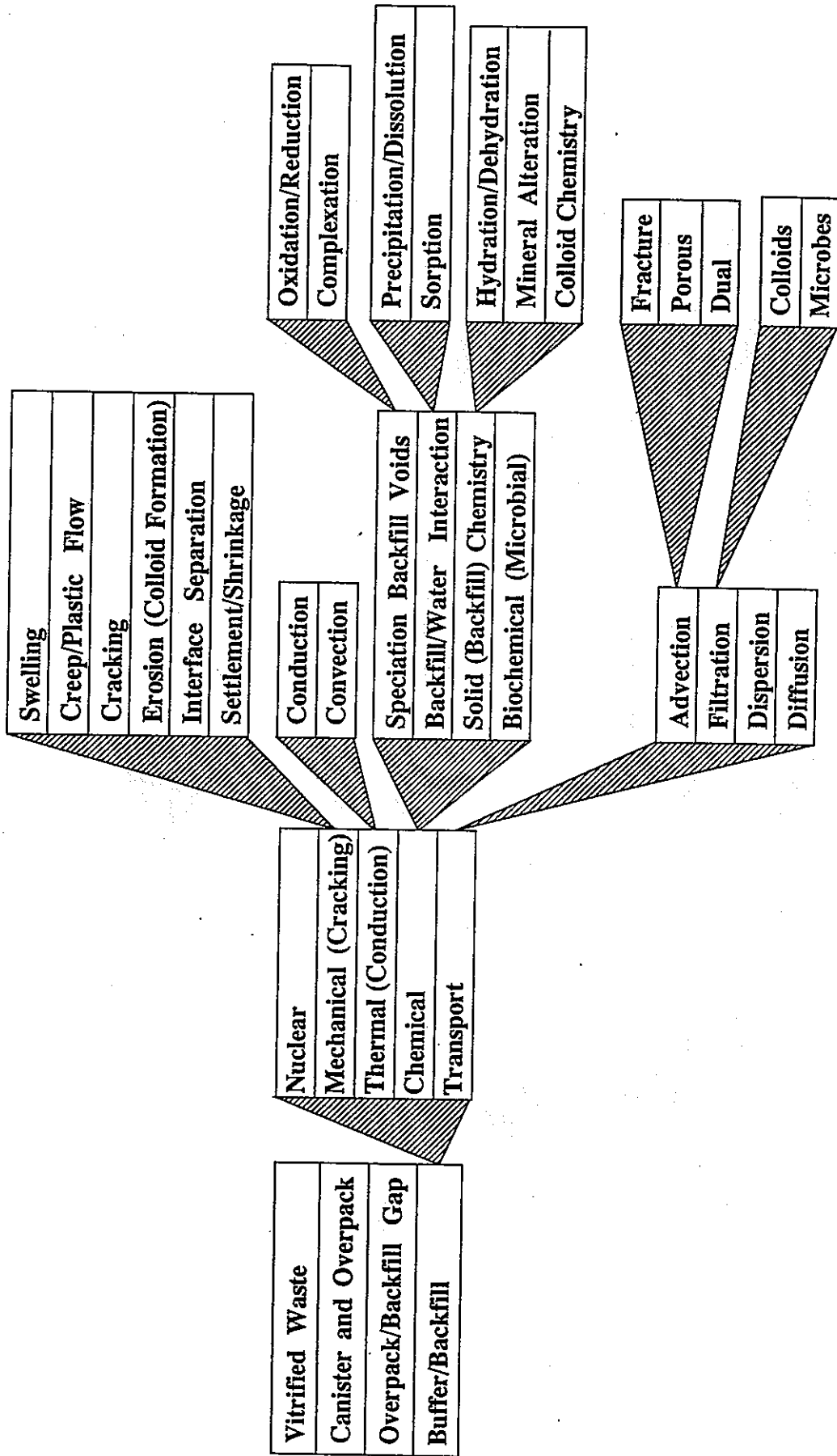


図2.1.2-2 (2/2) ニアフィールドのシステム評価モデル概念

# COUPLING DIAGRAM

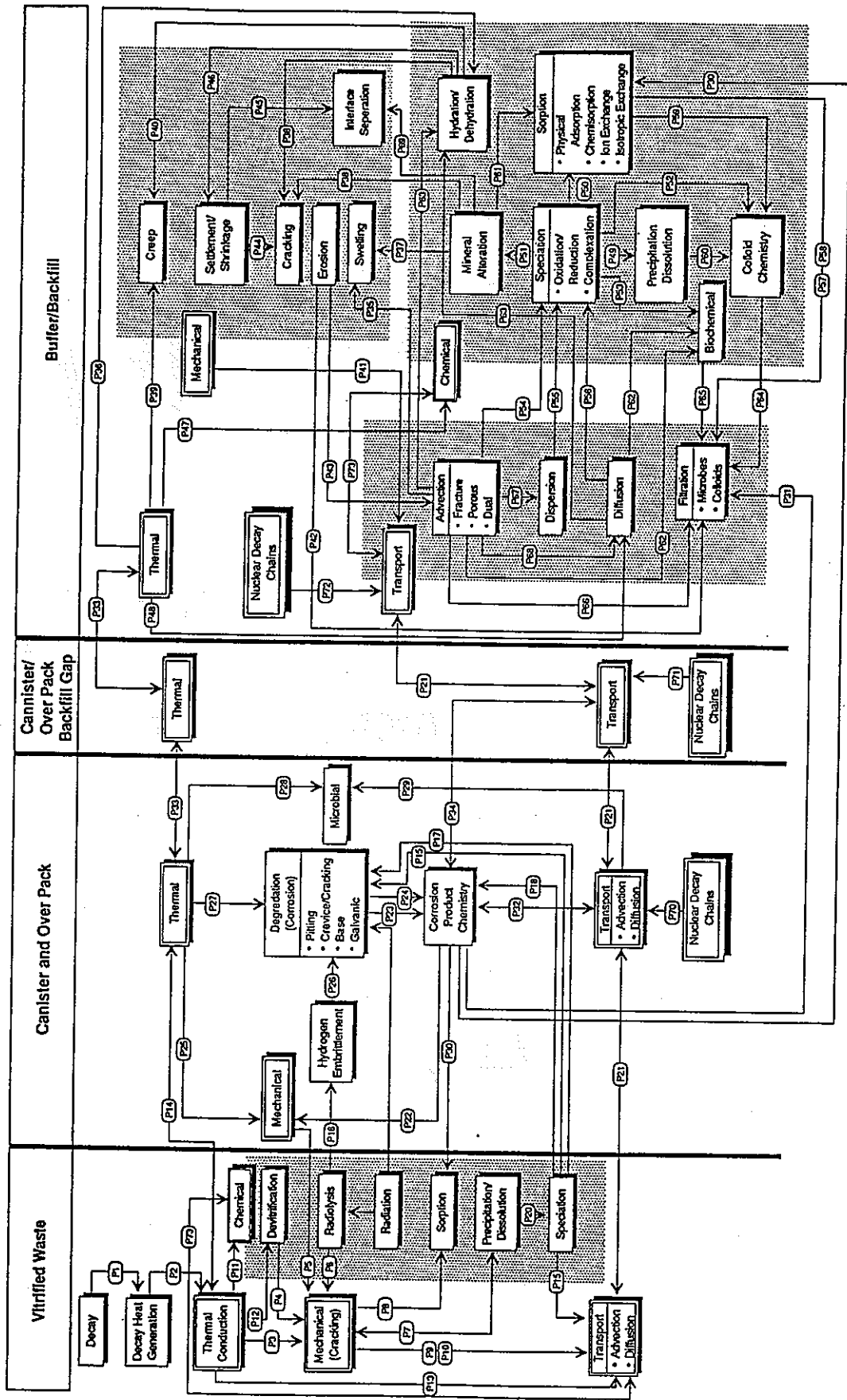


図 2.1.2-3 ニアフィールド・システム評価モデルのカップリング・ダイアグラム

# OVERLAYS ON HIERARCHY

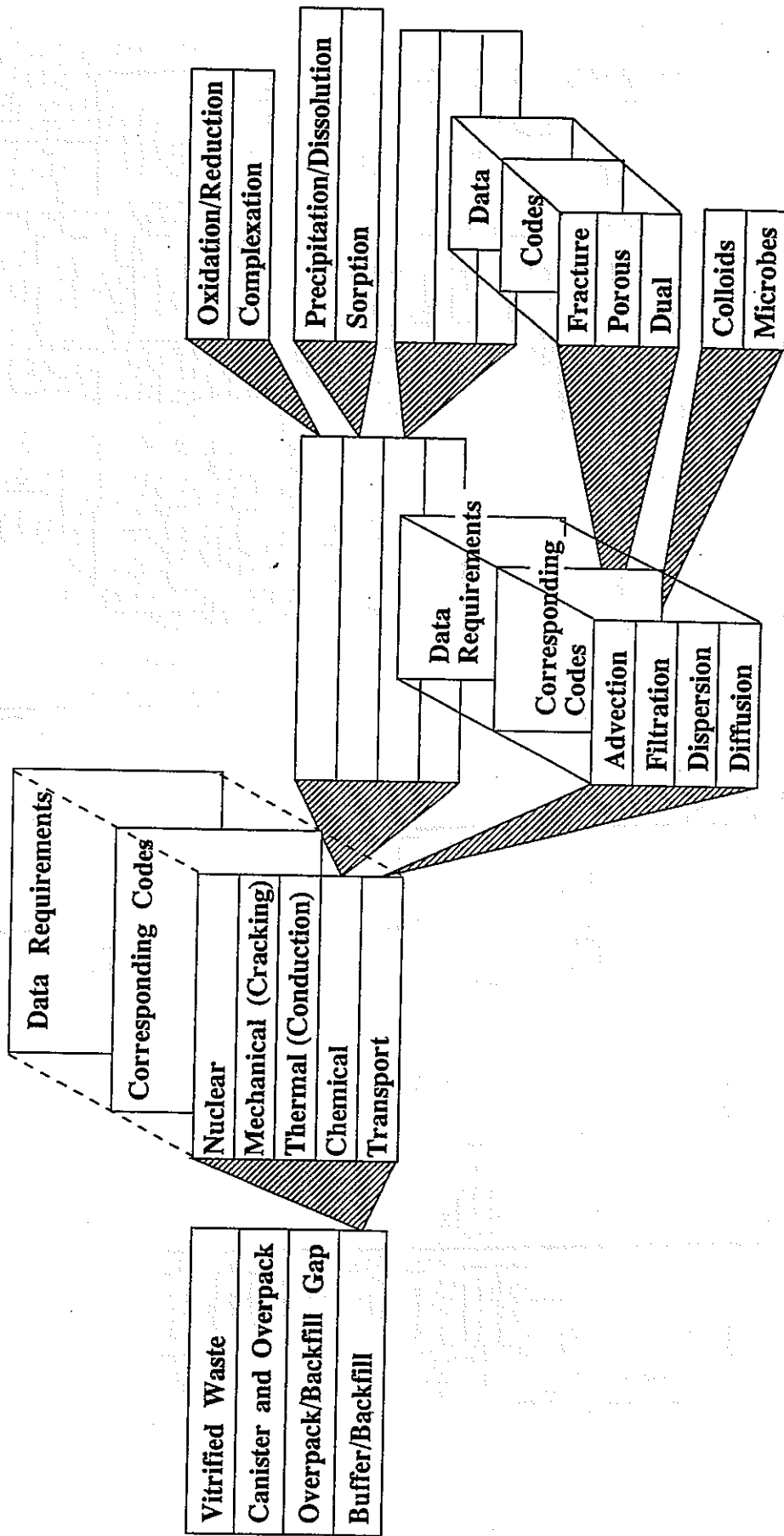


図 2.1.2-4 現象モデル/解析コード/データ分類の対応関係

## (1)解析内容／データの時間的・空間的分類

ここでは、現象解析コードをデータを介して連成し、ニアフィールドの性能評価解析を実現するために、コードの解析内容およびデータの時間的および空間的な連成について検討した。

連成解析に際して、各現象解析コードで実施する解析内容、および連成に使用するデータについて、時間的特徴と空間的特徴に分類して図 2.1.2-5 に示す。

解析内容については、時間的解析として、定常解析と非定常解析に分類でき、ほとんどの解析コードで両者の解析が実行できる。しかし、平衡論に基づく化学コードは、基本的には定常解析しかできない。これを非定常の連成解析に使用する場合は、1 タイムステップにおいて瞬時平衡が成り立つと仮定し、時間的に繰り返し計算を実行することにより、非定常解析すなわち化学種濃度の時間変化を求めることになる。

また、空間的解析としては、空間分布解析と点解析に分類できる。空間分布解析では、解析に空間メッシュを使用して、プロセス量の空間分布を求める解析である。点解析は、解析に空間メッシュが不要で、あるポイント（点）におけるプロセス量を求める解析であり、崩壊計算や化学計算がこれに相当する。このため、空間分布解析コードと点解析コードを連成する場合には、空間メッシュの全節点に対して、それぞれの節点において点解析コードを起動して計算を行う必要がある。

同様に、連成に使用するデータについても、時間的領域では、一定値（定数）の時間不変データと、時間変化をする時間依存データに分類できる。また、空間的領域においては、基本的にはすべて空間分布データに位置付けられる。しかし、点解析コードの入出力等に使用する場合は、あるポイントに関するデータのみを必要とするために便宜上点情報データとして分類することにする。これらの分類は、解析コードの出力だけでなく、実験データにも当てはまる。

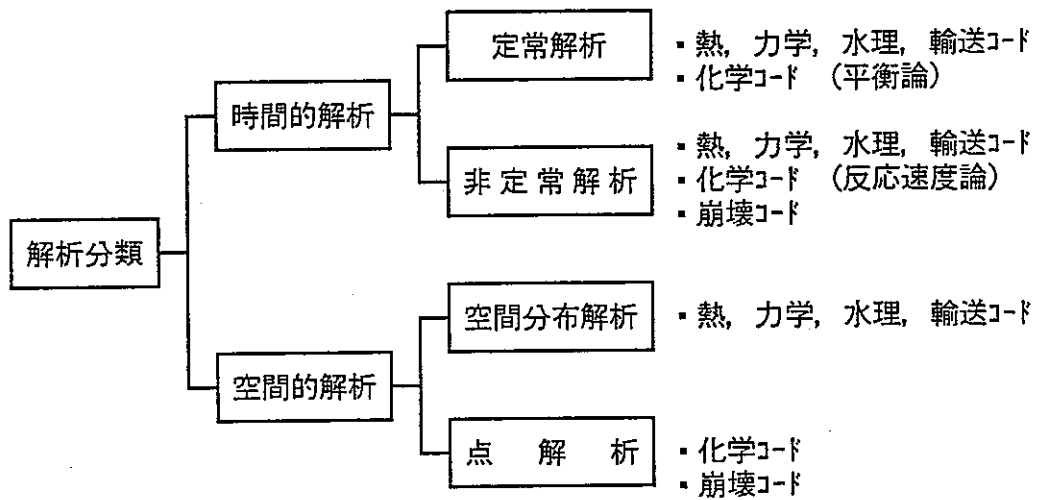
時間依存データは、非定常の連成解析において、計算時間に該当するデータを取り出して解析コードへ引き渡すため、補間処理を必要とする場合もある。また、空間分布データと点情報データの連成では、連成するデータの空間一点変換、すなわち空間メッシュの各節点データを取り出して点解析コードへ引き渡し、計算後の各節点データから再び空間分布を構築する処理が必要となる。

先に述べたデータ・カテゴリおよび解析コードの出力データに関して、連成解析に使用する具体的なデータ項目を時間的および空間的特徴で分類して表 2.1.2-1 に示す。

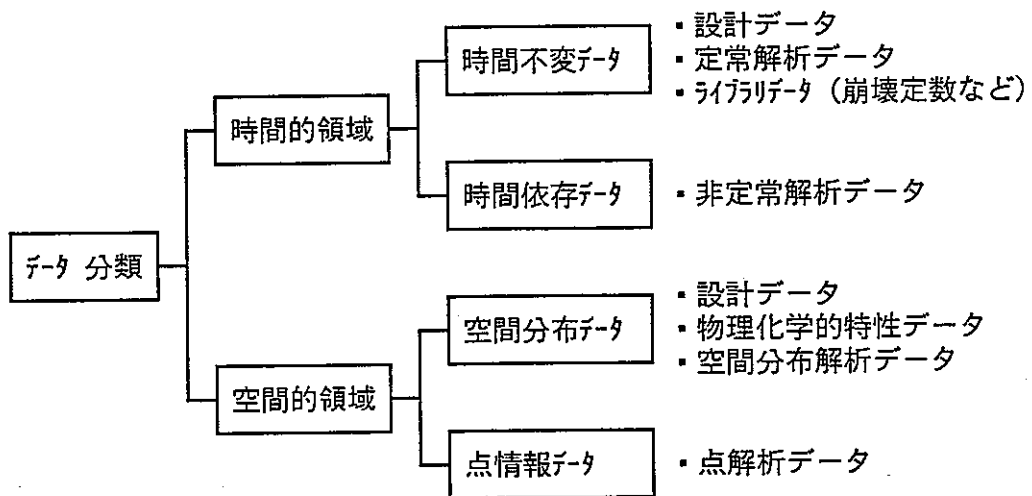
連成解析を実行するためには、使用する解析コードに入力するデータの時間的・空間

的特徴を把握し、それに対して必要十分なデータがあるかどうかを確認し、必要なデータの受渡し方法を決定することが重要と考えられる。





(a)解析実施内容の分類



(b)データ内容の分類

図 2. 1. 2 - 5 解析内容およびデータ内容の時間的/空間的分類

表 2. 1. 2 - 1 連成解析用データの特徴分類

連成用 データ		空間的領域	
		空間分布データ	点情報データ
時間 的 領 域	時間 不 変 デ ー タ	<ul style="list-style-type: none"> <li>▪ 物性値*<sup>1</sup> (伝熱, 力学, 水理等の物性)</li> <li>▪ 地層構造/地質データ</li> <li>▪ 解析形状 (空間メッシュ) (節点, 要素, 座標, 境界等)</li> <li>▪ 定常分布 (熱, 応力, 水理, 濃度)</li> </ul>	<ul style="list-style-type: none"> <li>▪ 化学的熱力学データ</li> <li>▪ 崩壊データ (崩壊定数, 系列)</li> <li>▪ 設計データ (物質, 次元)</li> <li>▪ ガラス固化体初期組成</li>   <li>▪ コード実行制御パラメータ</li> </ul>
	時間 依 存 デ ー タ	<ul style="list-style-type: none"> <li>▪ 空間分布データ (熱, 応力, 水理, 濃度)</li> <li>▪ 境界条件データ</li>   <li>{ ▪ 核種濃度 } *<sup>2</sup> ▪ 化学種濃度</li> </ul>	<ul style="list-style-type: none"> <li>▪ ガラス固化体イベントリ (ガラス溶解試験データも含む)</li> <li>▪ 崩壊熱 (発生熱)</li> <li>▪ 浸入地下水組成</li> <li>▪ 腐食データ</li> <li>▪ 核種濃度</li> <li>▪ 化学種濃度</li> </ul>

\*1: 物性値を定数として解析コードへ入力するものとして、時間不変データに分類した。しかし、ある種のプロセス量 (熱など) に応じて変化する場合は時間依存データとなる。この場合、解析コード内にプロセス量に対する補正式が組み込まれていなければ、それらの関係式をコード外部で定義する必要がある。

\*2: 核種濃度および化学種濃度は、解析形状に依存しない崩壊コードおよび化学コードの出力として得られるため、基本的には空間分布を持たない。しかし、連成解析、特に輸送コードとの連成を実現するためには、輸送コードの解析メッシュの全節点における計算を実行し、空間分布を求める必要がある。

## (2)データ接続の分類

地下水シナリオに基づく性能評価解析では、ガラス固化体から浸出・溶解した核種の移行プロセスの評価が重要な役割を担っている。このため、昨年度開発した CAPASA システムに実装した HYDROGEOCHEM と ORIGEN コードによる連成解析を基に、データの受渡しおよび接続方法について整理した。

昨年度の CAPASA システムで対象とした連成問題は、

- ・ガラス固化体のインベントリ評価：ORIGEN
- ・地球化学－輸送の連成評価：HYDROGEOCHEM
- ・空間領域における核種の崩壊連鎖：ORIGEN

である。また、輸送評価では緩衝材領域（拡散輸送）と周辺岩盤領域（移流輸送）に分けて考え、2つの HYDROGEOCHEM の連成解析を実現した。

しかし、ガラス固化体からの浸出・溶解評価と輸送評価の連成に関して、溶解濃度データの受渡しに課題があり、ガラス固化体の枯渇計算の評価までは至らなかった。そのため、ガラス固化体からの浸出・溶解評価を組み込んだ連成解析を行うためのデータ受渡しについて検討した。

また、昨年度は定常地下水流を仮定した輸送評価であったが、輸送評価に必要な地下水分布を得るための水理評価の組み込みについても検討した。

ガラス固化体から浸出・溶解した核種の移行プロセスの連成に必要な評価項目と解析コードとして、

- ・ガラス固化体：初期インベントリ評価           ：ORIGEN
- ・       "        ：核種の溶解評価                ：地球化学コード（PHREEQE 等）
- ・       "        ：核種の崩壊連鎖評価           ：ORIGEN  
                  （インベントリの経時変化評価）
  
- ・緩衝材領域    ：地球化学－輸送の連成評価：HYDROGEOCHEM
- ・       "        ：核種の崩壊評価                ：ORIGEN
- ・       "        ：水理評価                        ：例えば HYDROFLOW
  
- ・周辺岩盤領域：地球化学－輸送の連成評価：HYDROGEOCHEM
- ・       "        ：核種の崩壊評価                ：ORIGEN
- ・       "        ：水理評価                        ：例えば HYDROFLOW

が考えられる。核種の溶解評価に使用する解析コードは現状では選定されていないため（または溶解プロセスを厳密にモデル化した解析コードがないため）、溶解／沈殿反応による溶解度の評価解析が可能である地球化学コード（例えば PHREEQE）を対象として考えた。また、水理評価に関しても選定されたコードがないため、HYDROGEOCHEMで用いる水理場を評価するために開発された HYDROFLOWを対象に考えた。

各現象解析コードの連成に必要なデータの受渡しの接続関係を図 2.1.2-6 に示す。解析制御モジュールは連成解析実行の全体を制御するモジュールとして考え、ここで時間制御も行うとする。このため、非定常の連成解析では、全ての解析コードの時間ステップは、このモジュールで管理され、インプットファイルに設定される。従って、図中には示していないが、解析制御モジュールの時間制御パラメータ（特に時間ステップデータ）は、全計算プロセスに引き渡される。

各計算プロセス間のデータの受渡しについては、各解析コードが必要とするデータの空間的分類、すなわち空間分布データと点情報データにより異なり、以下のケースが考えられる。

#### ①空間分布データ接続（水理-輸送化学間のデータ受渡し）

どちらのコードも空間分布データを入出力するため、空間分布データを全て受け渡す接続方式である。ここで、有限要素コード間の接続では、対応する節点のデータをそのまま引き渡すことができるが、有限要素-差分コード間の接続では座標変換処理が必要となる。

#### ②空間-点変換接続（輸送化学-崩壊間のデータ受渡し）

空間分布解析コードと点解析コードを連成する場合には、空間メッシュの全節点に対して、それぞれの節点において点解析コードを起動して計算を行う必要がある。このため、連成するデータの空間分布データ-点情報データ変換、すなわち空間メッシュの各節点データを取り出して点解析コードへ引き渡し、計算後の各節点データから再び空間分布を構築する処理が必要となる。

#### ③境界データ接続（コード間の境界におけるデータ受渡し）

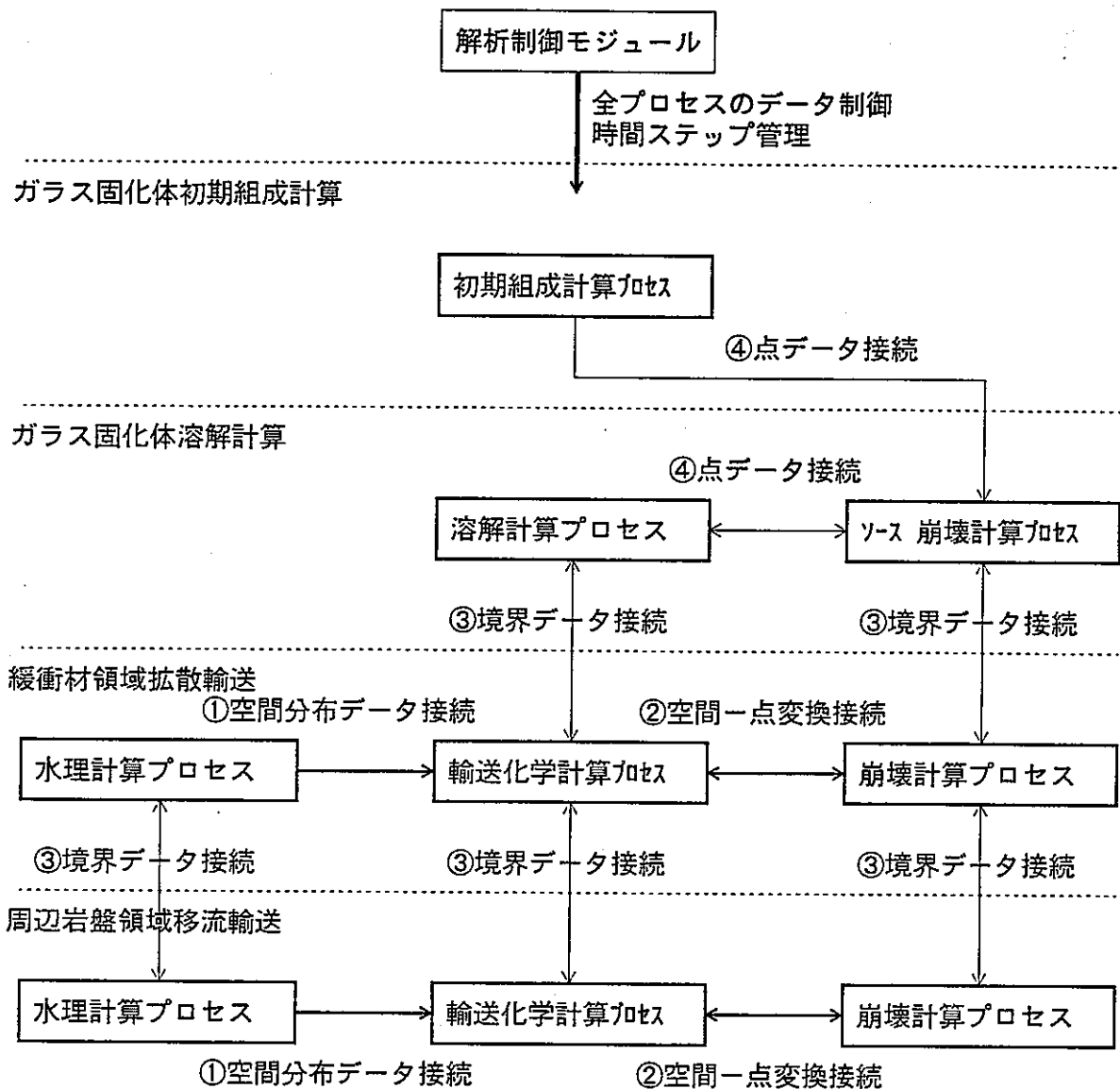
空間分布データおよび点情報データのいずれにおいても起こりうるデータ接続方式で、対応する境界節点データのみを受け渡す方式である。

#### ④点データ接続

点解析コード間を連成する場合のデータ接続方式で、点情報データをそのまま受

け渡す方式である。

また、崩壊計算プロセスは点解析コードのため、入出力は点情報データである。しかし、輸送化学計算との連成により、崩壊計算プロセスは節点データを空間分布に再構築して出力する。このため、崩壊計算プロセス間の接続方式は、点データ接続ではなく、輸送プロセスの空間メッシュにおける境界データ接続となる。



①空間分布データ接続

空間分布を持つ解析コード間の接続方式で、空間分布データを全て引き渡す。

②空間一点変換接続

空間分布を持つ解析コードと、形状データを持たないプロセスの接続方式で、空間メッシュの各節点毎に点プロセスを起動し、計算終了後、空間分布に戻して引き渡す。

③境界データ接続

対応する境界節点のデータを引き渡す。

④点データ接続

形状データを持たない解析コード間の接続方式で、点情報データを引き渡す。

図 2.1.2 - 6 輸送-化学-崩壊-水理の連成評価におけるデータの接続方式

### (3)制御モジュールの接続関係の分析

次に、データの受渡しを考慮し、各評価コードの連成解析を実現するための制御モジュールの接続フローについて整理して、図2.1.2-7に示す。

解析制御モジュールは連成解析実行の全体を制御するモジュールとして、対象核種、対象元素、空間メッシュ、境界節点の接続関係など、解析全体に係わる条件/データの設定を行うことを考えている。このため、解析時間に関するデータ（時間ステップ等）も同時に設定できる方が望ましく、非定常解析の時間制御機能を本モジュールに持たせることも可能である。

初期組成計算モジュールは、ガラス固化体のインベトリ計算を制御し、最初の時間ステップのみ実行される。これは連成解析の初期条件となるため、最初に接続する。

ガラス固化体の溶解計算は、ソース崩壊計算モジュールと溶解計算モジュールで実行される。ソース崩壊計算モジュールは固化体内の核種の崩壊計算を制御し、溶解計算モジュールは崩壊後の核種（元素）の溶解（度）計算を制御する。

溶解濃度は輸送化学計算の境界条件として引き渡されるため、溶解計算モジュールは必ず輸送化学計算モジュールの前に接続する。

また、ソース崩壊モジュールは、CAPASA (PLAN構築) の柔軟性から考えて、溶解計算モジュールの前後どちらに接続することも可能な構成とする必要があるが、溶解計算による化学種のバランスを保つためには、溶解計算モジュールの前に接続することが望ましい。

なぜならば、溶解計算モジュールの後に接続した場合、ある時間ステップにおいて、溶解計算で得られた濃度（化学種）に対して崩壊計算を実施すると、崩壊による核種（元素）濃度の減少により化学種（溶解/沈殿平衡）のバランスが崩れてしまう。一方、溶解計算の前に接続した場合は、ある時間ステップにおいて、まず崩壊後のソースの核種の濃度を求め、これに基づいて溶解計算するために化学種（溶解/沈殿平衡）のバランスが保たれた出力を得ることができると考えられるためである。

緩衝材領域拡散輸送の計算は、水理計算モジュール、崩壊計算モジュール、輸送化学計算モジュールにより実行される。

水理計算モジュールは地下水流速分布の計算を制御しており、流速分布が輸送計算に引き渡されるため、必ず輸送化学計算モジュールの前に接続する。また、連成解析の初期に全時間ステップ分の計算を実施しておくことも可能であり、初期組成計算モジュールのすぐ下に接続することもできる。ただし、この接続の場合は、水理計算の時間制御

と輸送化学計算の時間制御を分けて行うことになり、解析全体の時間制御が複雑になる恐れがある。

崩壊計算モジュールについては、昨年度 CAPASA では輸送化学計算の後に接続していたが、輸送化学計算前に接続する方が望ましいと考えられる。輸送化学計算の後に接続した場合は、ある時間ステップにおいて、輸送化学計算で得られた濃度（化学種）分布に対して崩壊計算を実施すると、崩壊による元素濃度の減少により化学種のバランスが崩れてしまう。一方、輸送化学計算の前に接続した場合は、ある時間ステップにおいて、まず崩壊後の核種（元素）の濃度を求め、これを輸送化学計算するために化学種のバランスが保たれた出力を得ることができると考えられるためである。

このため、崩壊計算モジュールは、モジュールの機能上は輸送化学計算モジュールの前後どちらに接続することも可能な構成とする必要があるが、輸送化学計算による化学種のバランスを保つため、輸送化学計算モジュールの前に接続することが望ましい。

周辺岩盤領域輸送の計算は、緩衝材領域拡散輸送の計算の場合と同様である。



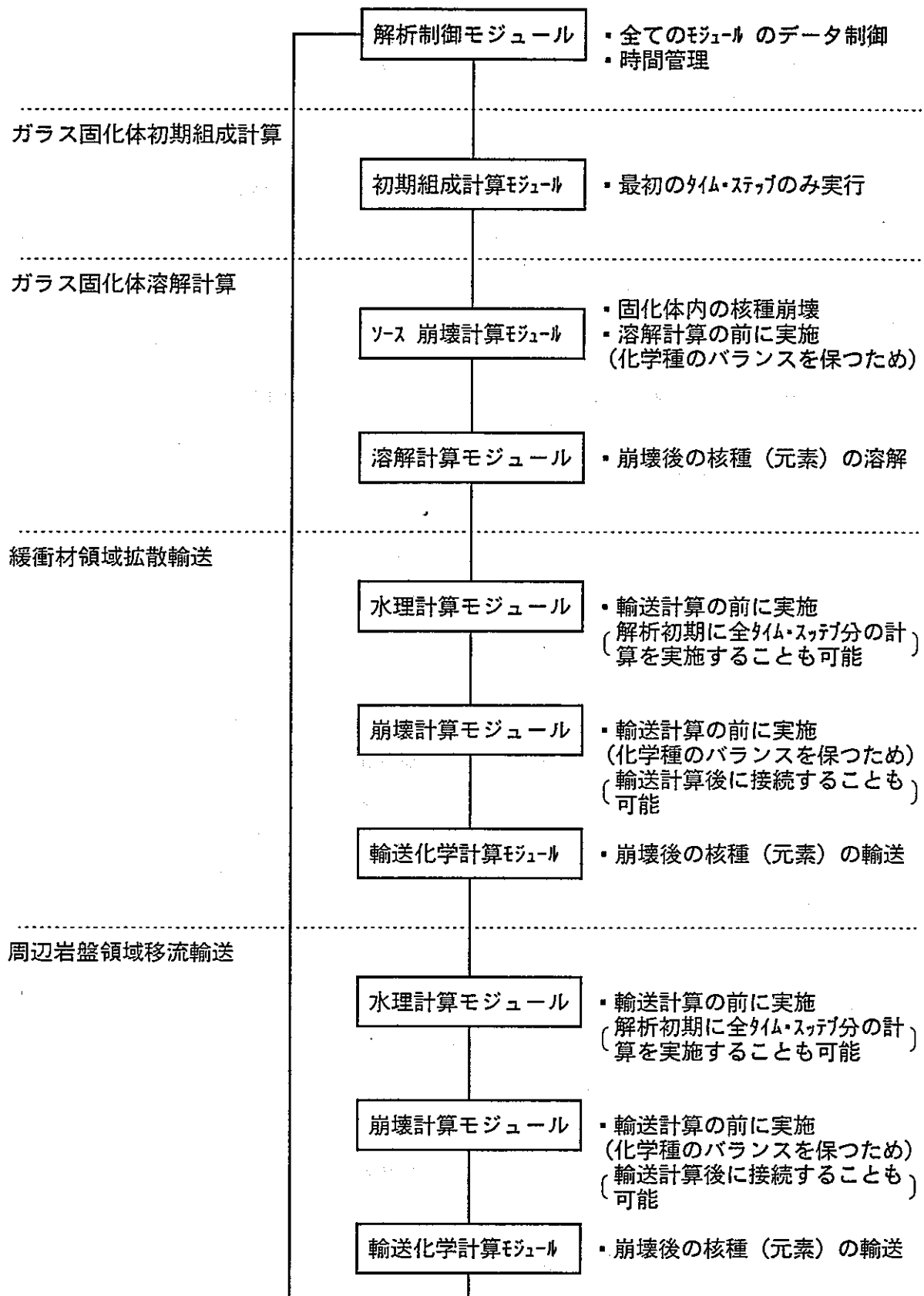


図 2.1.2 - 7 輸送-化学-崩壊-水理の連成評価における制御モジュールの接続フロー

#### (4)データ・フローの分析

前項でまとめた連成評価におけるデータ接続関係および制御モジュールの接続関係に基づいて、輸送-化学-崩壊-水理の各プロセスを連成解析する場合の具体的なデータ・フローを整理して、図2.1.2-8に示す。

図2.1.2-8(1/3)は、ガラス固化体の初期組成計算および溶解計算におけるデータの流れを示している。初期組成の計算では、解析制御部で設定された燃焼/冷却/再処理に関する通常の入力ファイルにより、ガラス固化体の初期組成データを出力する。このデータは解析の最初の時間ステップのときに、核種濃度データとしてソース崩壊計算の入力データとして使用される。

ソース崩壊計算では、崩壊計算を定義した入力ファイルと核種濃度データを入力データとして、固化体中の崩壊後の核種濃度データを出力する。ここで、入力に使用する核種濃度データは、前時間ステップにおける存在比データと固化体の元素濃度から計算する。出力された崩壊後の核種濃度データから、存在比データを求め、次時間ステップの計算のために保存する。さらに、存在比データは、溶解した元素中の核種の存在比として、輸送化学①計算(緩衝材領域)の境界節点に渡され、崩壊①計算で使用される。また、崩壊後の核種濃度データから求めた元素濃度は、溶解計算の入力データとして使用される。

溶解計算では、溶解/沈殿反応を定義した入力ファイル、ソース崩壊計算後の元素濃度データ、および前時間ステップの輸送化学①計算で得られた固化体周辺の地下水組成データを入力データとし、溶解濃度を出力する。この溶解濃度は、輸送化学①計算の境界条件として渡される。また、溶解濃度から固化体中の元素濃度を求め、次時間ステップの崩壊計算のために保存する。

図2.1.2-8(2/3)は、緩衝材領域拡散輸送解析におけるデータの流れを示している。水理①計算では、計算条件を定義した入力ファイル、前時間ステップの水頭分布、および水理②計算(周辺岩盤領域)で得られた境界節点データを入力データとし、流速分布および水頭分布を出力する。流速分布は、輸送化学①計算の入力データとして使用される。また、水頭分布は、次時間ステップの水理計算のために保存する。さらに、境界節点の水頭データは水理②計算(周辺岩盤領域)の境界節点に渡される。

崩壊①計算では、崩壊計算を定義した入力ファイルと核種濃度データを入力データとして、輸送化学①計算の空間メッシュの全節点に対して計算を実行し、崩壊後の核種濃度データを出力する。ここで、入力に使用する核種濃度データは、前時間ステップにおける存在比データおよび濃度空間分布、ソース崩壊計算で得られた境界存在比データ、さらに前時間ステップで得られた輸送化学②計算および崩壊②計算で得られた境界濃度

と存在比データから、各節点毎に計算する。出力された崩壊後の核種濃度データから、存在比データを求め、次時間ステップの計算のために空間分布データとして保存する。さらに、境界節点の存在比データは、輸送化学②計算（周辺岩盤領域）の空間メッシュの境界節点データとして渡され、崩壊②計算で使用される。また、崩壊後の核種濃度データから求めた元素濃度の空間分布データは、輸送化学①計算の入力データとして使用される。

輸送化学①計算では、計算条件を定義した入力ファイル、崩壊①計算後の元素濃度データ、水理①計算で得られた流速分布、溶解計算で得られた境界節点データを入力データとし、濃度空間分布を出力する。得られた濃度分布の中で、境界節点の濃度は輸送化学②計算（緩衝材領域）の空間メッシュの境界節点データとして渡され、崩壊②計算で使用される。また、ソース側の境界節点の地下水組成濃度は、次時間ステップの溶解計算の地下水組成データとして使用される。

図2.1.2-8(3/3)は、周辺岩盤領域移流輸送解析におけるデータの流れを示している。水理②計算では、計算条件を定義した入力ファイル、水理①計算で得られた境界節点データ、および前時間ステップの水頭分布を入力データとし、流速分布および水頭分布を出力する。流速分布は、輸送化学②計算の入力データとして使用される。また、水頭分布は、次時間ステップの水理計算のために保存する。さらに、境界節点の水頭データは水理①計算（緩衝材領域）の境界節点に渡される。

崩壊②計算では、崩壊計算を定義した入力ファイルと核種濃度データを入力データとして、輸送化学②計算の空間メッシュの全節点に対して計算を実行し、崩壊後の核種濃度データを出力する。ここで、入力に使用する核種濃度データは、前時間ステップにおける存在比データ、崩壊①計算で得られた境界節点の存在比データ、および濃度空間分布から計算する。出力された崩壊後の核種濃度データから、存在比データを求め、次時間ステップの計算のために空間分布データとして保存する。また、崩壊後の核種濃度データから求めた元素濃度の空間分布データは、輸送化学②計算の入力データとして使用される。さらに、境界節点の存在比データは、輸送化学①計算（緩衝材領域）の境界節点データとして渡され、崩壊①計算で使用される。

輸送化学②計算では、計算条件を定義した入力ファイル、崩壊②計算後の元素濃度データ、水理②計算で得られた流速分布、および輸送化学①計算で得られた境界節点データを入力データとし、濃度空間分布を出力する。得られた濃度分布の中で、境界節点の濃度は、輸送化学①計算（緩衝材領域）の空間メッシュの境界節点データとして渡され、崩壊①計算で使用される。

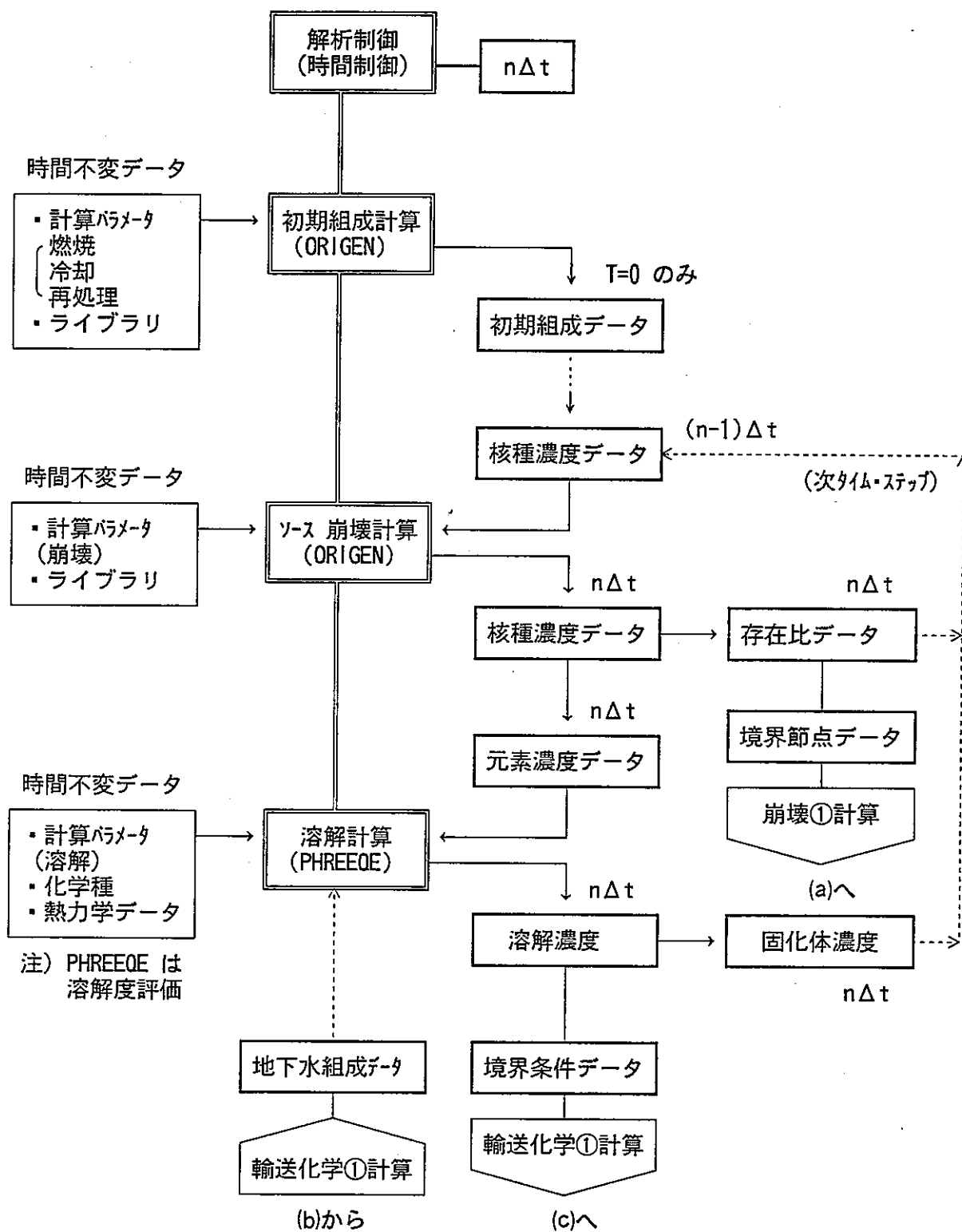


図 2. 1. 2 - 8 (1/3) 輸送-化学-崩壊-水理の連成評価におけるデータ・フロー (ガラス固化体の初期組成計算および溶解計算)

注) ①緩衝材領域  
②周辺岩盤領域

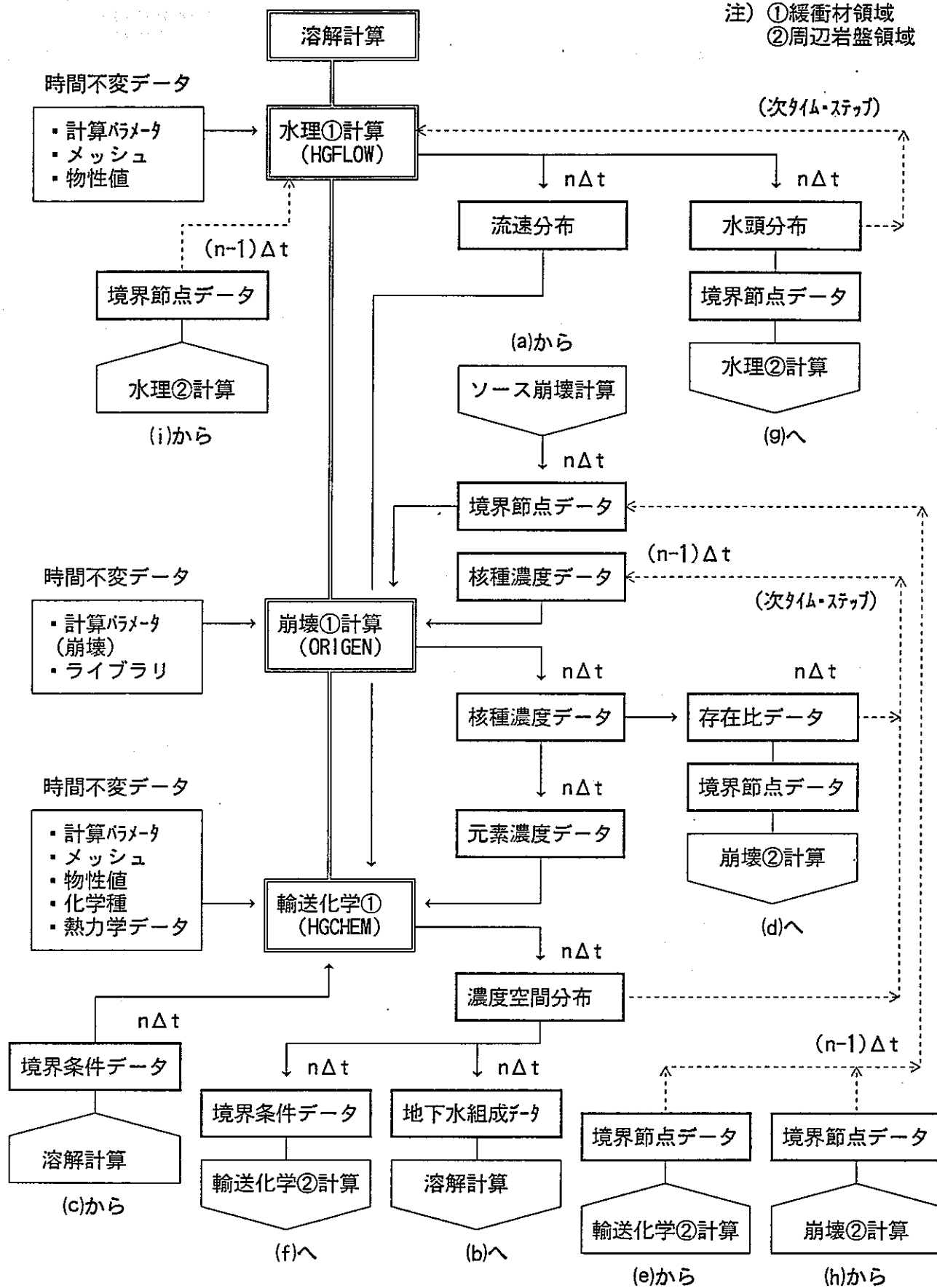


図 2.1.2 - 8 (2/3) 輸送-化学-崩壊-水理の連成評価におけるデータ・フロー (緩衝材領域における拡散輸送計算)

注) ①緩衝材領域  
②周辺岩盤領域

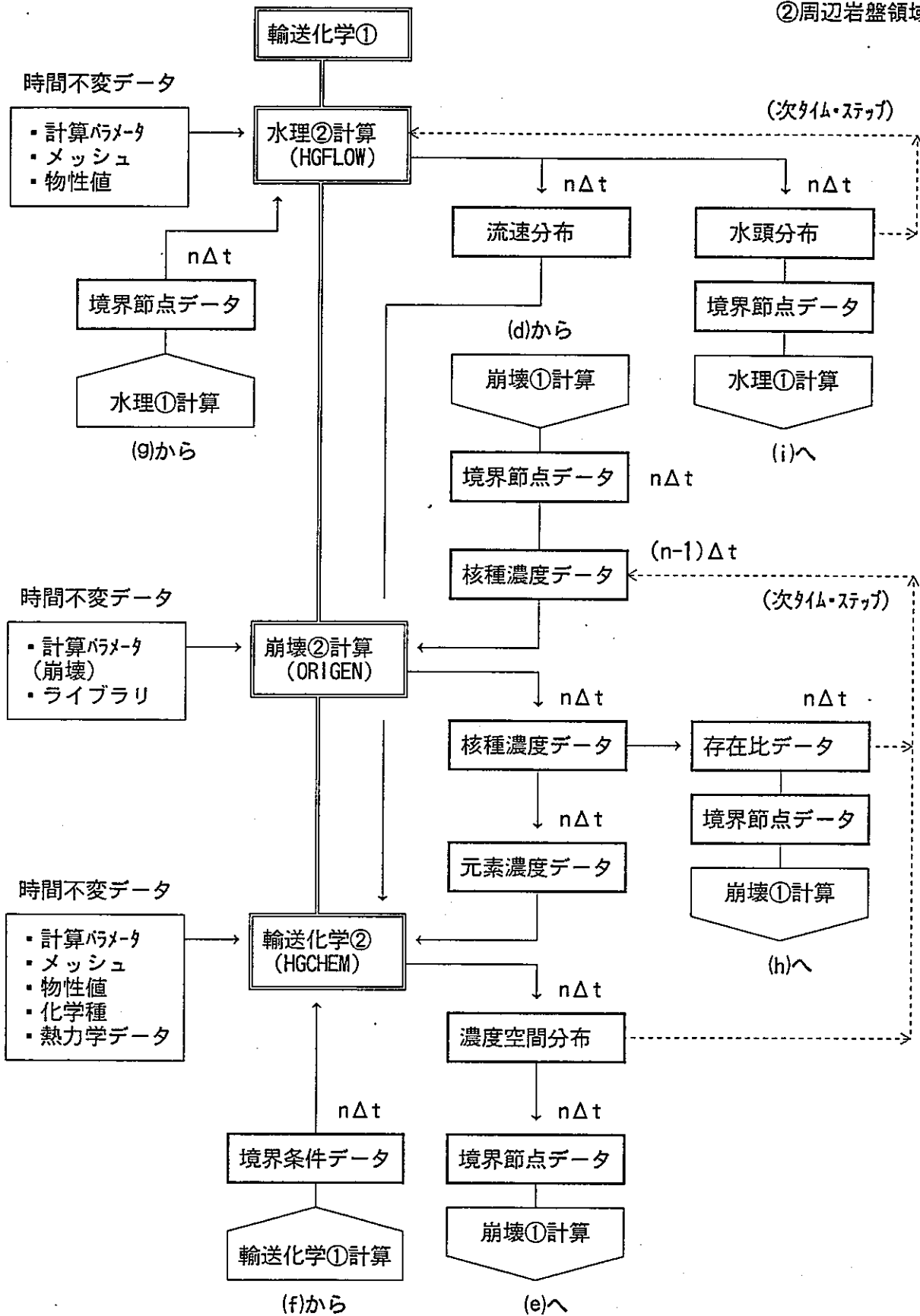


図 2.1.2 - 8 (3/3) 輸送-化学-崩壊-水理の連成評価におけるデータ・フロー (周辺岩盤領域における移流輸送計算)

#### (5)連成用計算制御モジュールの機能分析

輸送－化学－崩壊－水理の連成評価におけるモジュールの接続フロー（図2.1.2－7）およびデータのフロー（図2.1.2－8）から、各モジュールの機能について整理して、図2.1.2－9～16に示す。

初期組成計算モジュールの機能を図2.1.2－9に示す。このモジュールは、解析制御モジュールで設定した時間依存データ、時間不変データから入力ファイルを作成し、崩壊ライブラリ・ファイル等の入出力ファイルを指定して、コードを実行する。

ソース崩壊計算モジュールの機能を図2.1.2－10に示す。このモジュールでは、まず解析制御モジュールで設定した計算条件により入力ファイルを作成する。次に、使用する崩壊ライブラリ等を設定する。次に、前時間ステップの溶解計算およびソース崩壊計算の結果から、ソース濃度および存在比データを読み込んで核種濃度を計算し、組成（核種濃度）入力用にファイルを作成する。そして、ライブラリ等の入出力ファイルを指定してコードを実行する。最後に、計算された核種濃度から元素濃度および存在比を計算し、次時間ステップの計算のためにファイルに保存する。

溶解計算モジュールの機能を図2.1.2－11に示す。このモジュールでは、まず解析制御モジュールで設定した計算条件から、入力ファイルの計算パラメータを設定する。次に、ソース崩壊計算から元素濃度を、また輸送化学①計算から対応する境界節点の地下水組成をそれぞれ読み込んで化学条件（固相／液相濃度など）を設定して、入力ファイルを作成する。次に作成した入力ファイルや熱力学データベースなどの入出力ファイルを指定してコードを実行する。最後に、計算された溶解濃度からソース元素濃度を求め、次時間ステップのソース崩壊計算のためにファイルに保存する。

水理計算モジュールの機能を図2.1.2－12に示す。このモジュールでは、まず解析制御モジュールで設定した計算条件から、入力ファイルの計算パラメータを設定する。次に、境界が接続されている他の水理計算の境界水頭データを読み込んで境界条件を設定する。さらに、前時間ステップでの計算結果から水頭分布を設定し、入力ファイルを作成する。ここで、HYDROFLOW や FEMWATER などのコードにはリスタート機能があるため、リスタート・ファイルを使用すれば水頭分布を入力ファイルに設定する必要はない。最後に、入出力ファイルを指定してコードを実行する。

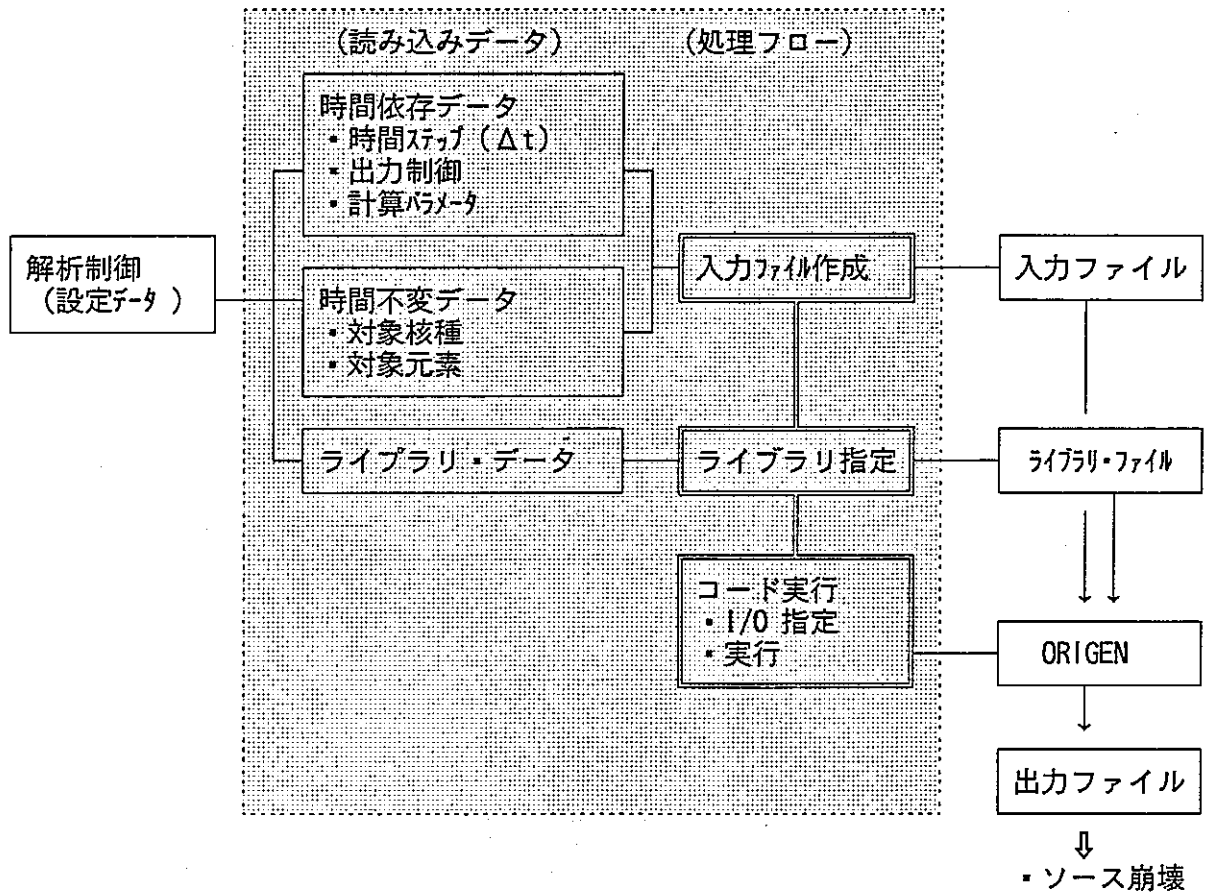
緩衝材領域での崩壊計算モジュールの機能を図2.1.2－13に示す。このモジュールでは、まず解析制御モジュールで設定した計算条件から入力ファイルを作成する。次に、

使用する崩壊ライブラリ等を設定する。次に、前時間ステップの輸送化学①計算および崩壊①計算、ソース崩壊計算、さらに岩盤領域での輸送化学②計算および崩壊②計算の結果から、濃度分布度および存在比データを読み込んで節点核種濃度を計算し、組成（核種濃度）入力用にファイルを作成する。次に、ライブラリ等の入出力ファイルを指定してコードを実行する。ここで、節点核種濃度の計算およびコードの実行については、輸送化学計算コードのメッシュ情報に基づいた分布一点変換処理が必要となり、メッシュの各節点でコードを実行する必要がある。最後に、計算された核種濃度から元素濃度および存在比を計算し、次時間ステップの計算のために空間分布の形でファイルに保存する。

緩衝材領域での輸送化学計算モジュールの機能を図2.1.2-14に示す。このモジュールでは、まず解析制御モジュールで設定した計算条件から入力ファイルの計算パラメータを設定する。次に、溶解計算の結果から、溶解濃度を読み込んで境界条件を設定する。なお、岩盤領域での輸送化学②計算から得られた境界節点濃度は、崩壊①計算の実行時に反映されているため、ここでは設定する必要はない。さらに、水理計算および崩壊計算の結果から流速分布および濃度分布を設定して、入力ファイルを作成する。ここで、水理計算に HYDROFLOWを使用する場合にはバイナリ・ファイルによる接続が可能であり、流速分布を入力ファイルに設定する必要はない。また、HYDROGEOCHEMのリスタート・ファイルを使用する場合には濃度分布を入力ファイルに設定する必要はないが、崩壊①計算モジュールでの出力ファイルをリスタート・ファイルと同じ形式のバイナリ・ファイルとして作成しなければならない。最後に、作成した入力ファイルや使用するバイナリ・ファイル等の入出力ファイルを指定してコードを実行する。

岩盤領域での水理計算モジュールは、緩衝材領域と同様に図2.1.2-12で表される。また、岩盤領域での崩壊計算モジュールの機能を図2.1.2-15に、輸送化学計算モジュールの機能を図2.1.2-16に示す。機能的な内容に関しては、読み込むデータファイルが異なるだけで、緩衝材領域でのそれぞれのモジュールと同じである。





注) ①緩衝材領域の計算

図 2.1.2 - 9 初期組成計算モジュールの機能

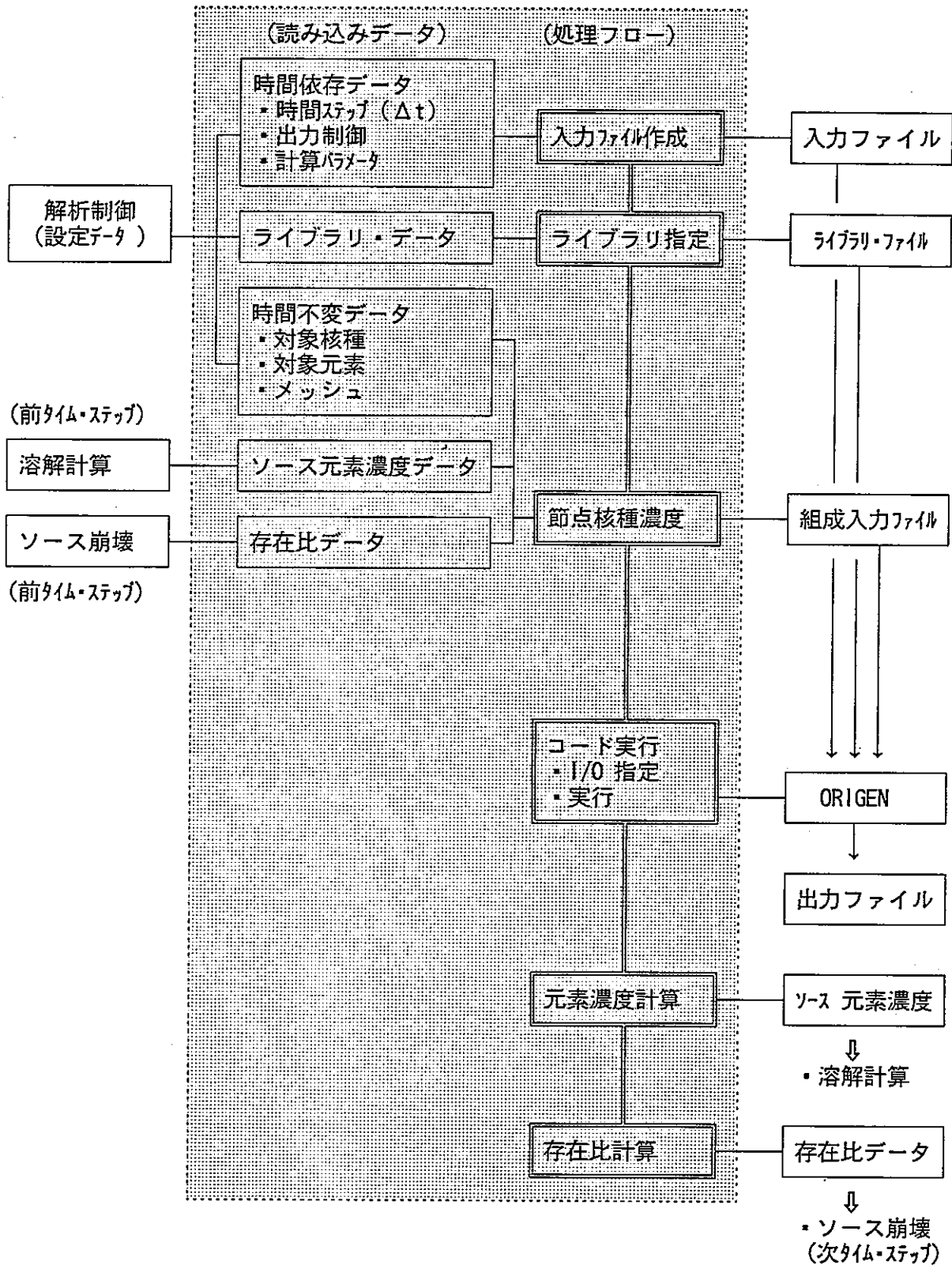
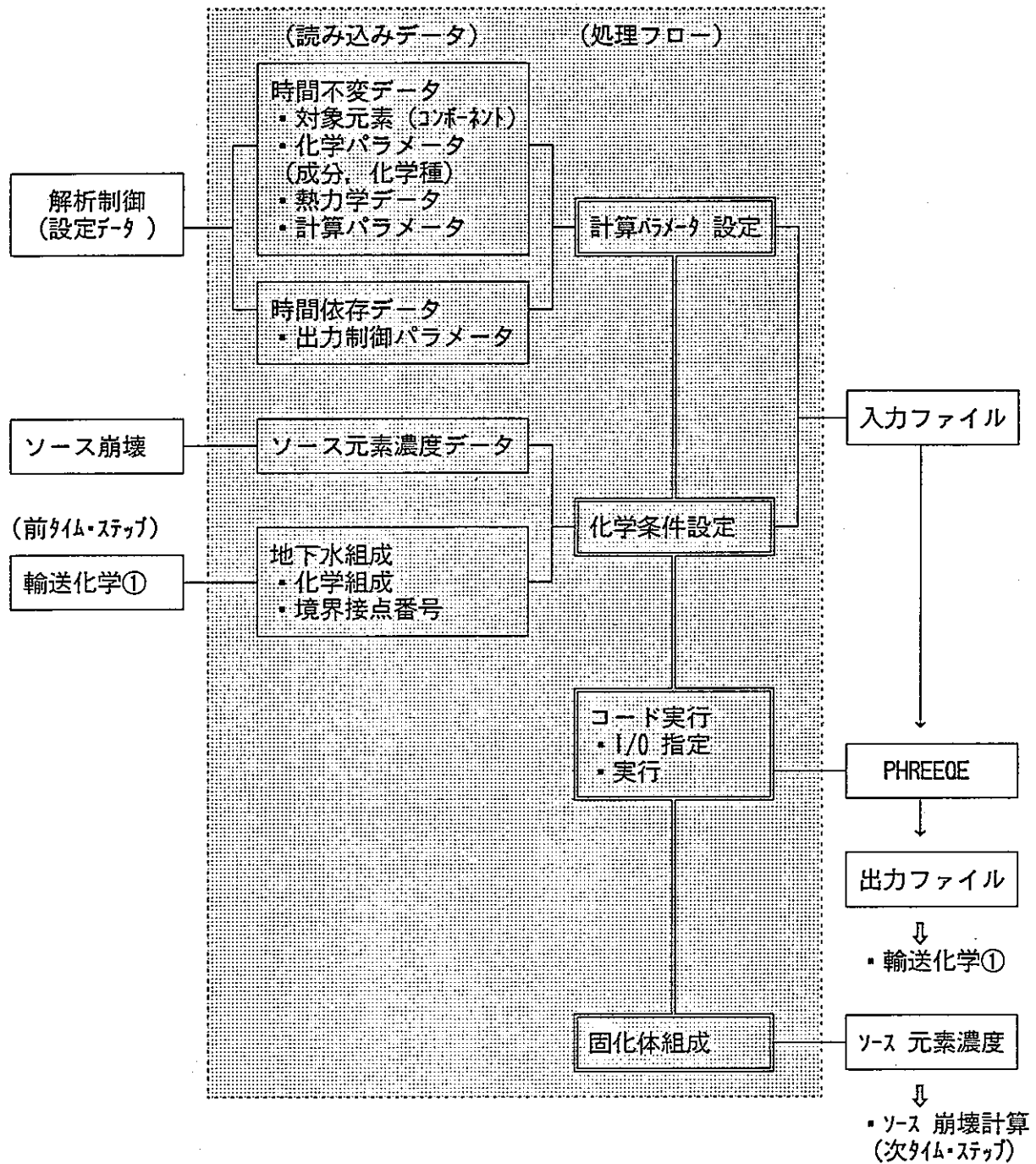


図 2. 1. 2 - 10 ソース崩壊計算モジュールの機能



注) ①緩衝材領域の計算  
PHREEQE は溶解度評価

図 2.1.2 - 11 溶解計算モジュールの機能

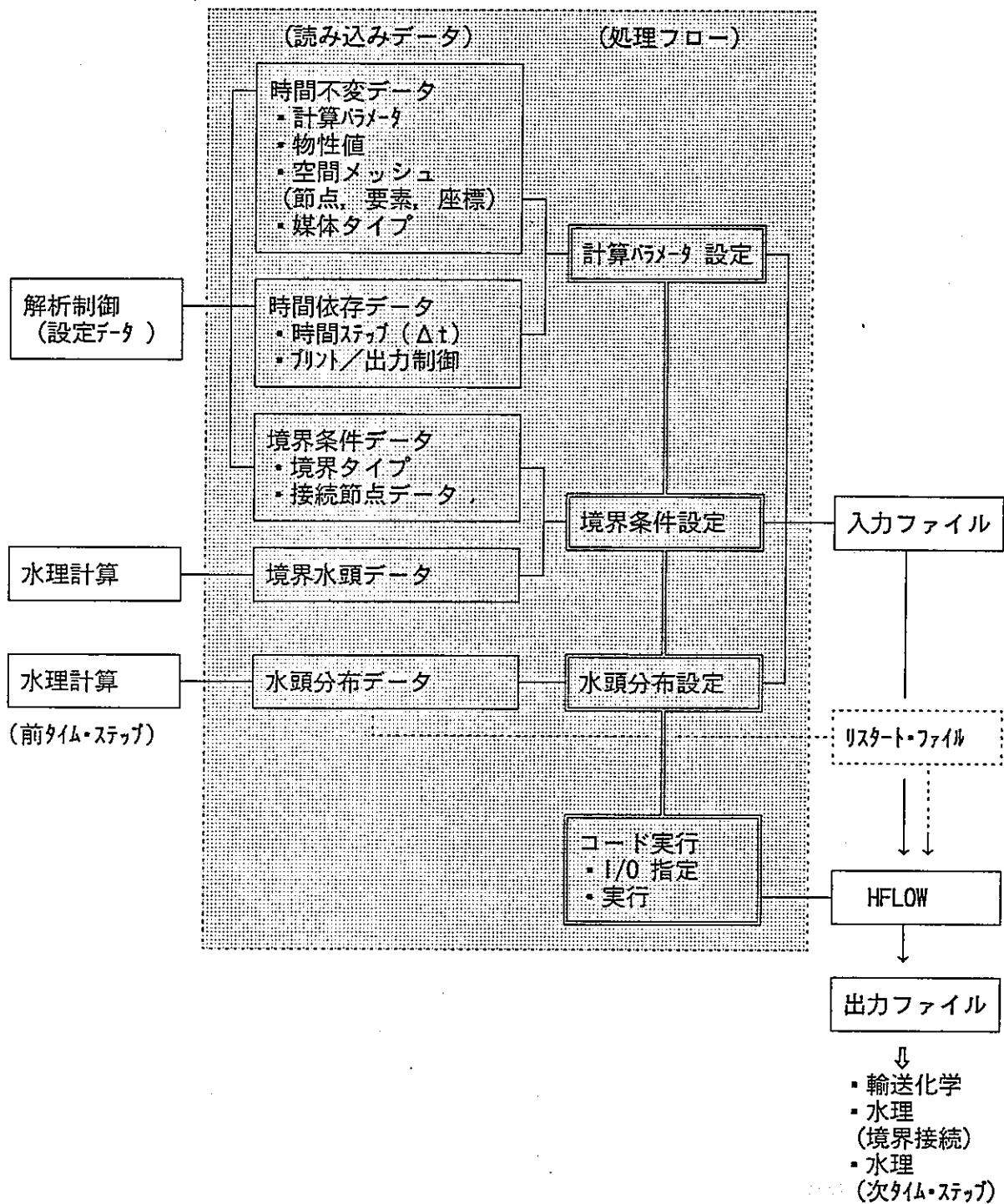


図 2.1.2 - 12 水理計算モジュールの機能

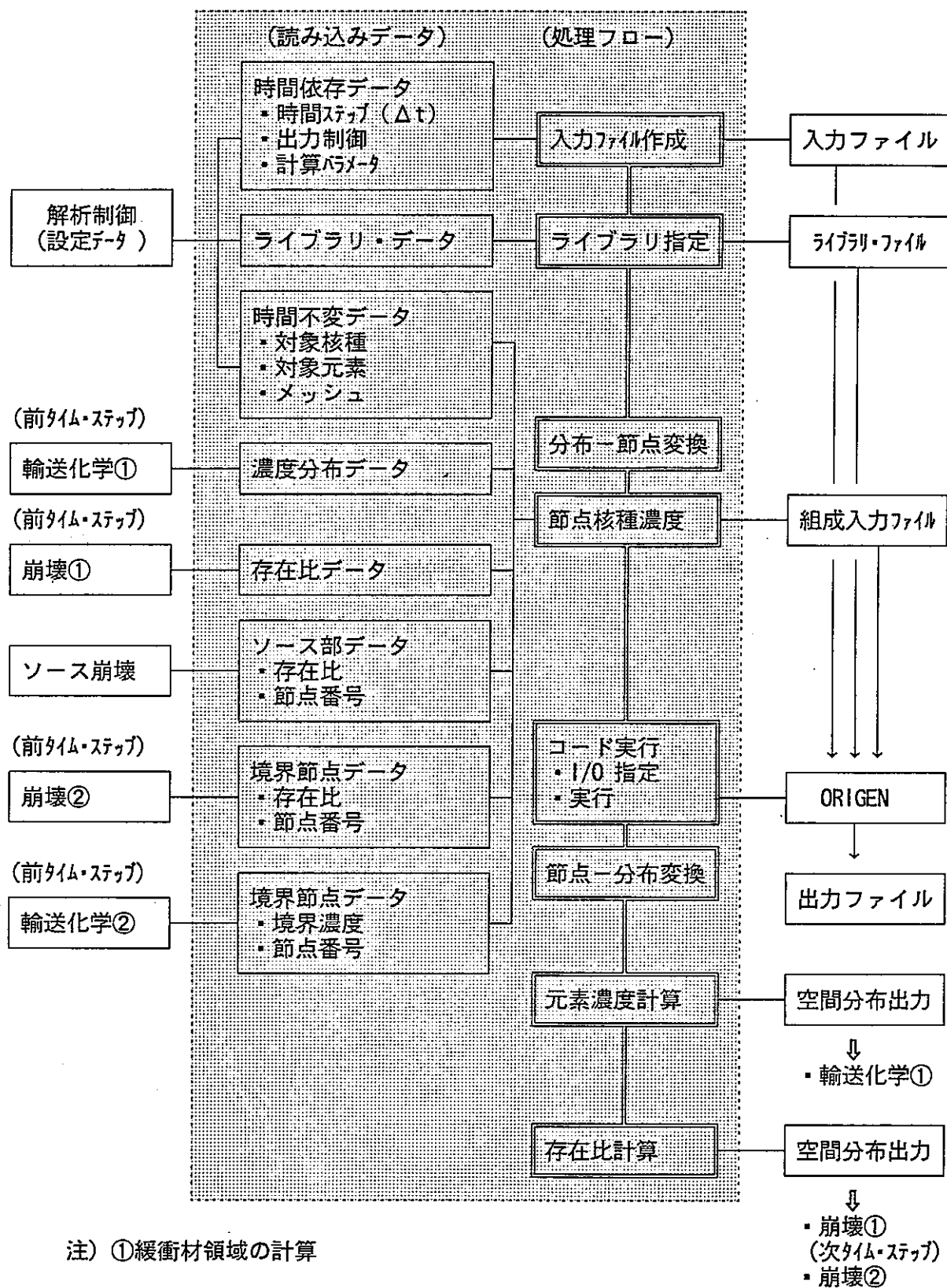


図 2.1.2 - 13 緩衝材領域での崩壊計算モジュールの機能

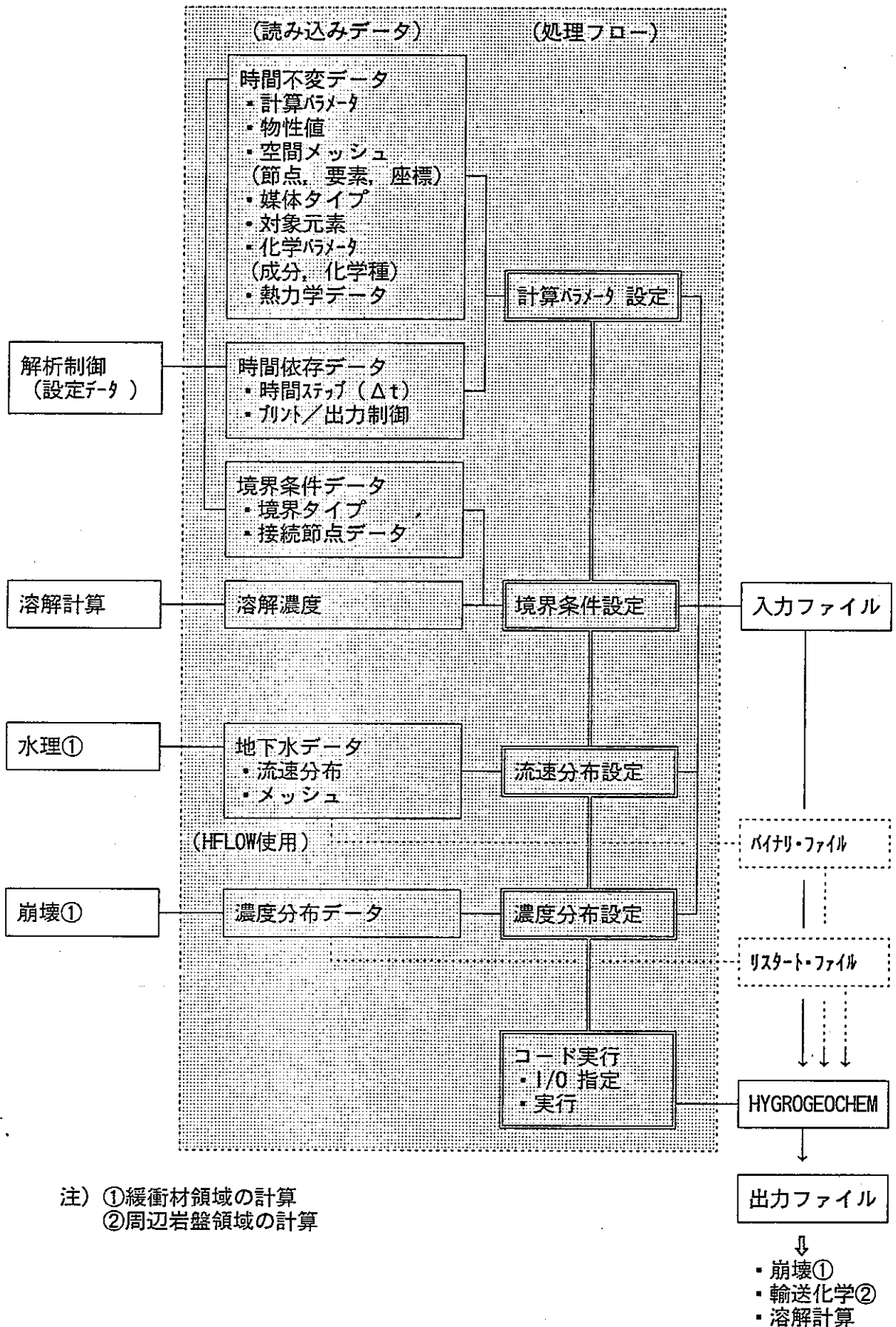


図 2.1.2 - 14 緩衝材領域での輸送化学計算モジュールの機能

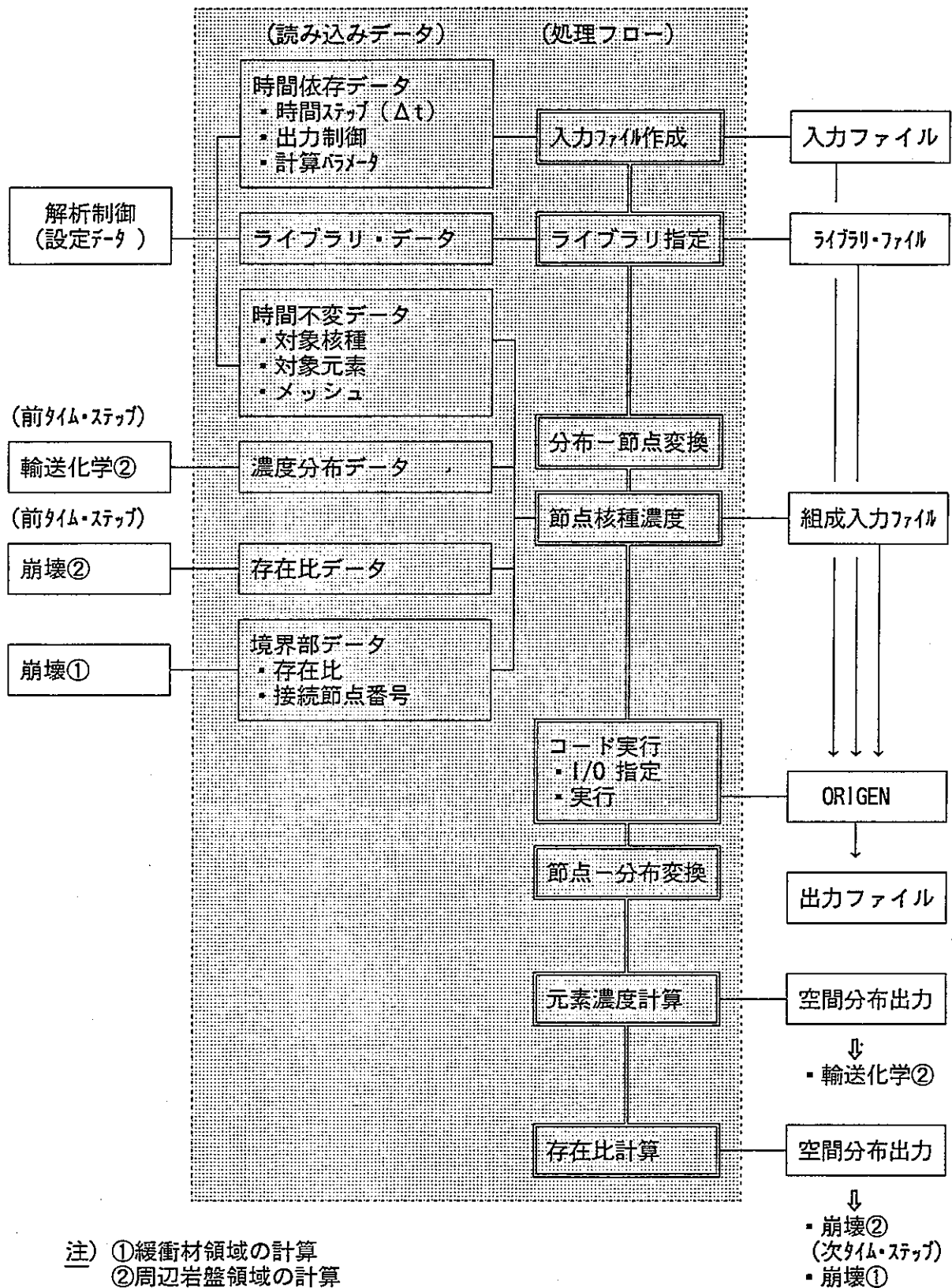


図 2. 1. 2 - 15 周辺岩盤領域の崩壊計算モジュールの機能

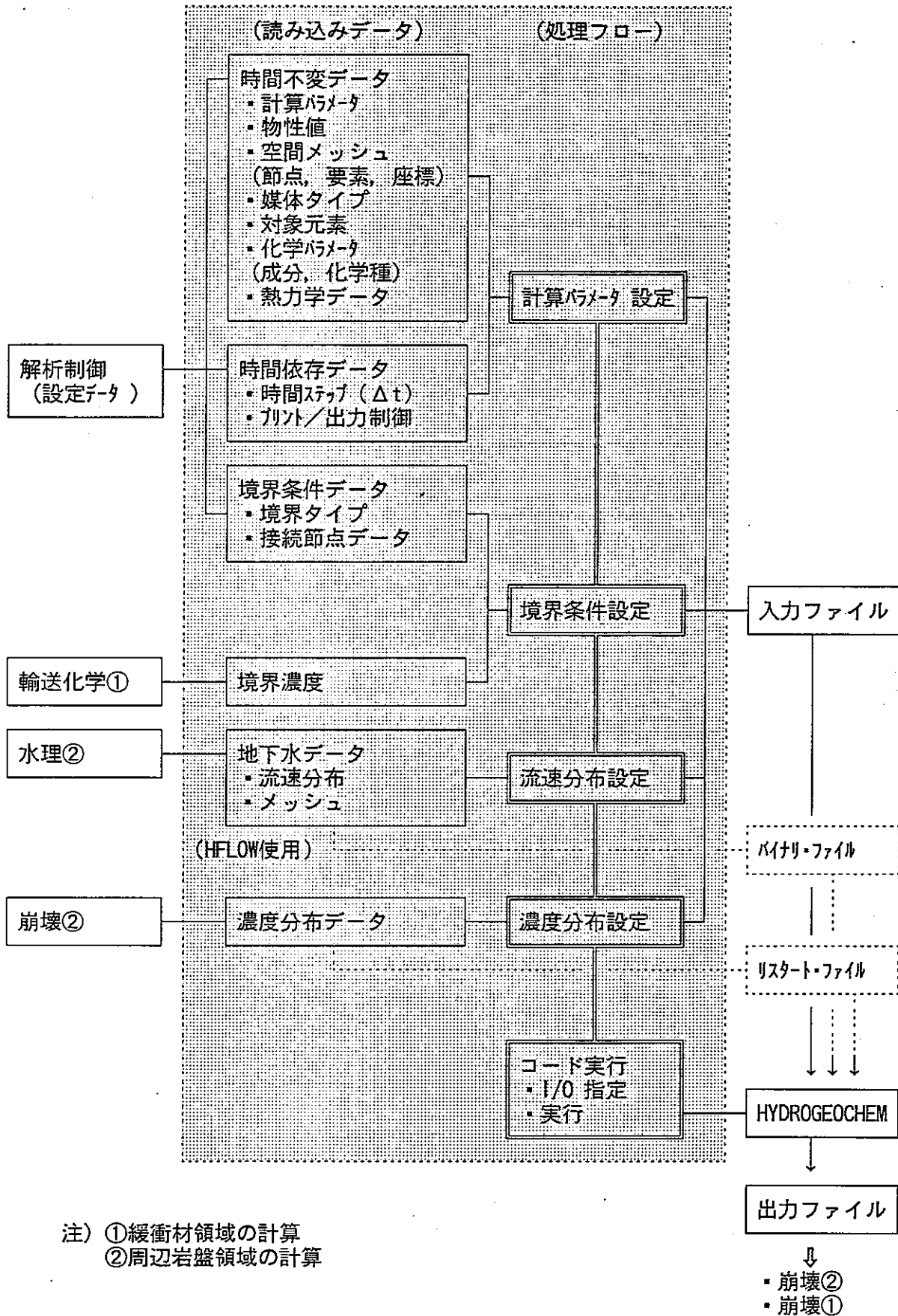


図 2.1.2 - 16 周辺岩盤領域での輸送化学計算モジュールの機能



## 2. 2 基本手順の作成

平成5年度の作業結果と今年度の研究結果をもとに定めた現象解析コード連成のための基本手順を図2. 2-1に示す。

### (1) 問題領域分析

#### ①問題領域に対するオブジェクト分析

- ・問題領域に対するオブジェクトダイアグラム

### (2) 解析領域分析

#### ①連成手順の分析

2. 1にて述べたように、連成モデルの分析においては、連成すべき各解析プログラム間のデータ接続関係を整理し、データフロー図を作成する。

- ・連成手順におけるデータ接続関係
- ・データフロー図

#### ②解析領域に対するオブジェクト分析

- ・解析領域に対するオブジェクトダイアグラム

### (3) PLANオブジェクト設計

連成解析手順のデータフロー図を基に、想定するPLANを定める(PPLANネットワーク仕様)。次に、そのネットワーク仕様から、各PLANオブジェクトの仕様とデータ変換処理仕様(データフローに対応する変換フィルター仕様)を定める。

- ・PLANネットワーク仕様
- ・PLANオブジェクト仕様
- ・データ変換処理仕様

### (4) データベース設計

- ・データベーススキーマ
- ・保存手続き

### (5) PCDEE拡張検討

必要に応じてPCDEEの関連モジュールの新規作成、機能追加の検討を実施する。

- ・機能仕様書

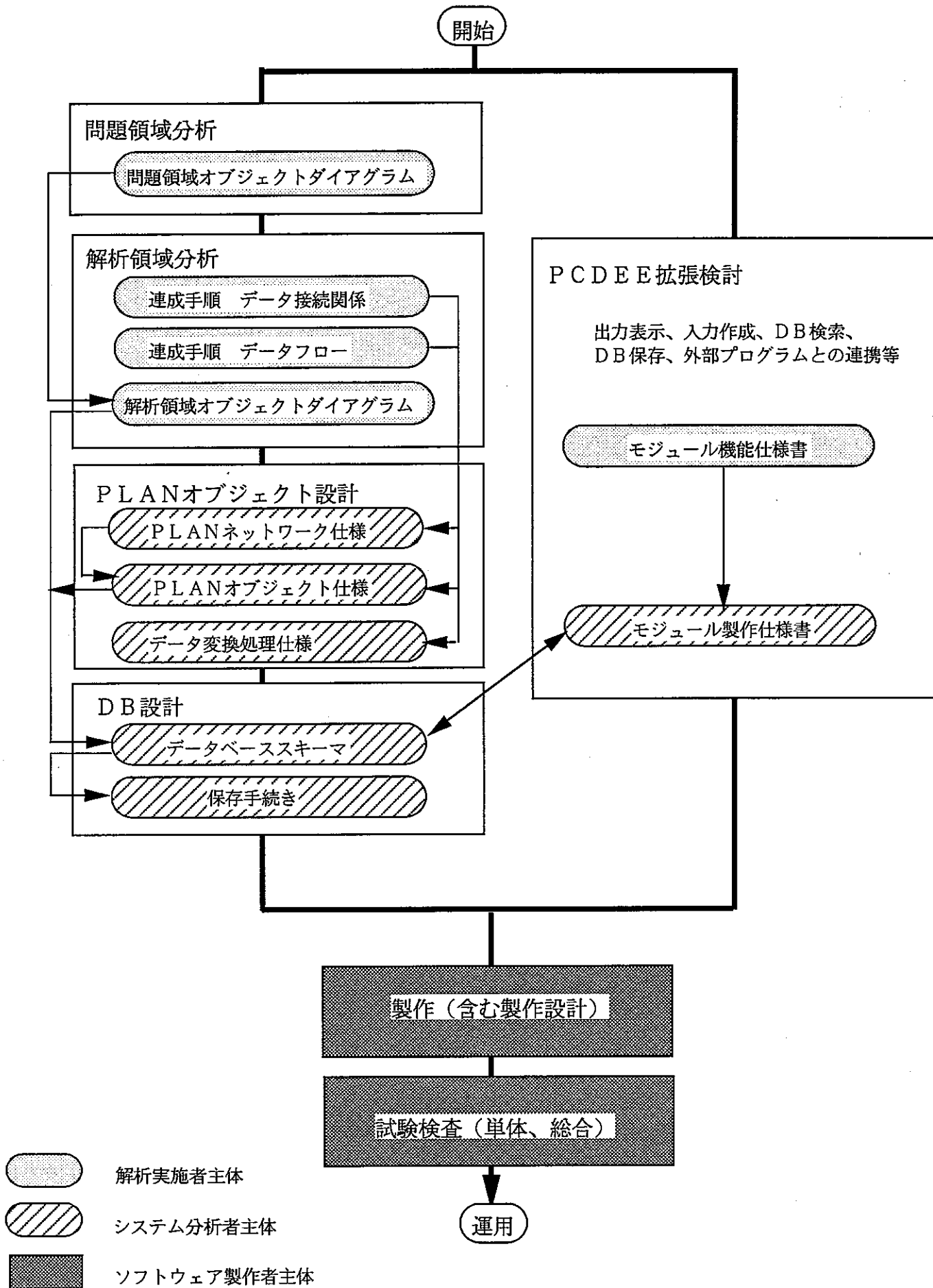


図2.2-1 連成解析コード連成のための基本手順

・製作仕様書

(6) 製作

PLANオブジェクト、データベース及びPCDEEの各部仕様に従い、製作する。

(7) 試験検査

製作した各モジュールについて試験検査を実施し、動作及び機能を確認する。

### 3. 連成解析システムの基本フレーム機能の拡充に関する検討

今後の連成解析の領域の拡張を念頭において、解析効率や拡張の際の作業効率の向上の観点から基本フレーム機能の高度化を検討した。

#### 3. 1 PLAN構築／実行モジュールの高機能化に関する検討・設計

PLAN構築／実行モジュールの開発でこれまで使用してきた市販アプリケーションAVSはモジュールの開発に有効な機能を有していたが、モジュールの機能や構成の自由度に大きな制約を与えていた。本節では、AVSの制約にとらわれることなく、解析効率や拡張作業の効率の向上の観点から、PLAN構築／実行モジュールの高度化を検討した結果について述べる。

##### 3. 1. 1 高度化の検討

平成5年度のシステムではAVSを使用したために、以下に示すようなPLANの表現や操作上の制約があった。

- ・データフローの表現ができないため、データの流れを直感的に理解できない。
- ・グローバル情報により、PLANに関するデータを設定していたため、各PLANオブジェクトのカプセル化が十分でなく、拡張性に劣る。
- ・PLANオブジェクトのグラフィカル表現が乏しい。全てのPLANオブジェクトは同一形状である。
- ・PLANオブジェクトに関連したデータの入力に操作性の制約がある。例えば、解析コードを表わしたPLANオブジェクトの入力データはPLAN構築／実行モジュール以外のモジュールを使用して入力する必要がある。

本年度は、これらの制約を改善するため、PLANの表現方法、及び、データの入力、参照方法について検討を行った。

まず、PLANの表現方法について述べる。AVSを用いた平成5年度のシステムでは、2.1.2節で述べたように、輸送－化学－水理の連成解析を行う際のデータの流れはグラフィカルには表現されずに、制御の流れのみが表現されている。そして、データの流れは図3.1.1-1に示すようなグローバル情報として定義された。例えば、HydrogeochemとDecayとのデータのつながりは、17、18行目に記述されている「decay1.decay hydrogeochem=Ex3Single」で表現されている。しかし、この方法では、データの流れを直感的に理解することは難しく、また、2つの解析コード間のデータの受け渡し方法については何ら記述がされていなかった。本年度は、これらの欠点を解決するため、PLANの表現に制御の流れを示すコントロールフローに加えて、データの流れを示すデータフローの概念を導入した。

1	[problem]	
2	origen=ORIGEN_MAP1	ORIGEN2で用いるデータの名称を設定
3	hydrogeochem=Ex3Single	Hydrogeochemでデータの名称を設定
4	decay=decay1	
5	scale factor=1.0	
6		
7	[Ex3Single.hydrogeochem]	Hydrogeochemの起動のための初期設定
8	active=1	
9		
10	[Ex3Single.hydrogeochem.components]	Hydrogeochemで対象とするコンポーネントの設定
11	AMERICIUM=95, 1	
12	SOH=0, 0	
13		
14	[Ex3Single.hydrogeochem.radionuclides]	Hydrogeochemで対象とする核種の設定
15	AMERICIUM=952410	
16		
17	[decay1.decay]	DecayとHydrogeochemのデータの関連を設定
18	hydrogeochem=Ex3Single	
19		
20	[origen]	
21		
22	[timer]	

図3.1.1-1 グローバル情報

データフローとは、複数のPLANオブジェクト間のデータの流れ、及び、データの受け渡し方法について記述したものである。データフローは以下の属性を持つ。

- ・上流PLANオブジェクト データを送るPLANオブジェクトの名称
- ・下流PLANオブジェクト データを受け取るPLANオブジェクトの名称
- ・フィルター データ変換を行うための外部プログラムの名称

データフローを用いたデータの流れを示す。データは図3.1.1-2に示すように、データフローの上流のPLANオブジェクトから下流のPLANオブジェクトにデータが受け渡される。その際、データ変換が、データフローの属性として与えられたデータ変換プログラム「フィルター」を介して行われる。つまり、上流のPLANオブジェクトにより出力されたデータは、「フィルター」によりデータのフォーマット等の変更が行われ、下流のPLANオブジェクトの入力データとなる。

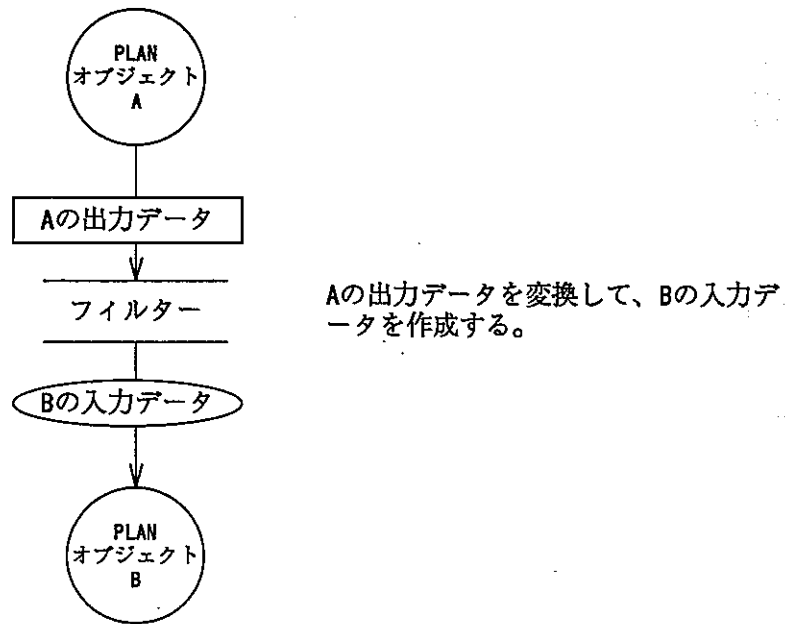


図3.1.1-2 データフローを用いたデータの流の制御

上記の例は、上流のPLANオブジェクトの1つの出力データが下流のPLANオブジェクトの1つの入力データになる単純な例である。しかし、より複雑な解析を行う際には、複数の上流のPLANオブジェクトの出力データが結合されて、下流のPLANオブジェクトの1つの入力データになる場合も考えられる。よって、データフローは、図3.1.1-3に示したようなデータの流を扱う必要がある。

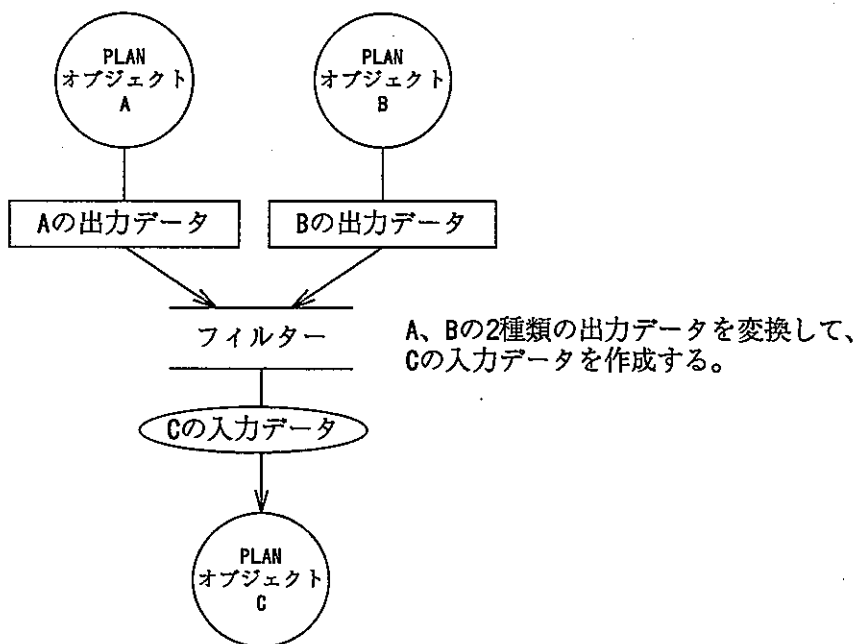


図3.1.1-3 複数の出力データを扱うデータフロー

また、図3.1.1-4に示すようなデータフローの下流の解析コードが複数の入力データを用いる場合は図3.1.1-2の変形として、表現することができる。

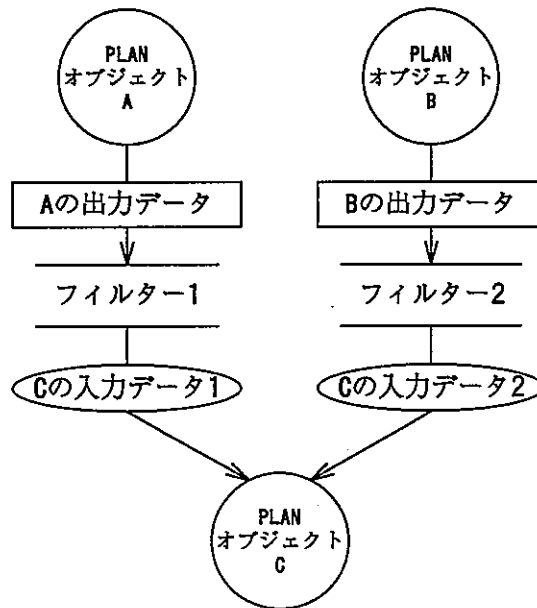


図3.1.1-4 複数の入力データを扱うデータフロー

次に今年度検討したデータフローの概念と平成5年度までに検討してきたコントロールフローやPLANオブジェクトの概念を加えたPLANの表現方法について述べる。

PLANは以下の3種類のアイテムより構成される。

(1)PLANオブジェクト

ORIGEN2やHydrogeochem等の解析コードを表わした解析コードオブジェクトや分岐条件等を表わしたユーティリティオブジェクトがある。PLANオブジェクトには、複数のコントロールフローとデータフローが接続される。なお、PLANオブジェクトは種類に応じた属性を持つ。

(2)コントロールフロー

制御の流れを規定したアイテム。上流、下流ともに各々1個ずつのPLANオブジェクトが接続される。

(3)データフロー

データの流れを規定したアイテム。上流側にN個、下流側に1個のPLANオブジェクトが接続される。データフローはフィルター（データ変換プログラム）の名称を表わした属性を持つ。

図3.1.1-5にPLANの表現例を示す。これは、以下の手順で実行が行われる。

(1)ObjAとObjBが実行される。

(2)ObjAの実行が終了すると、ObjCが実行される。また、ObjBの実行が終了すると、Filter1が実行され、ObjD用の入力データが作成される。

- (3)ObjCの実行が終了すると、Filter2が実行され、ObjD用の入力データが作成される。
  - (4)ObjB、ObjC、Filter1、Filter2の実行の終了を待って、ObjDが実行される。
  - (5)ObjDが終了すると、ObjEにてPLANの解析時間の判定が行われ、終了時間に達していない場合は、再度、ObjA、ObjCが実行される。
  - (6)ObjCの実行が終了すると、Filter2が実行され、ObjD用の入力データが作成される。
  - (7)ObjDが実行される。
- 以下、(5)から(7)までが終了時間まで繰り返される。

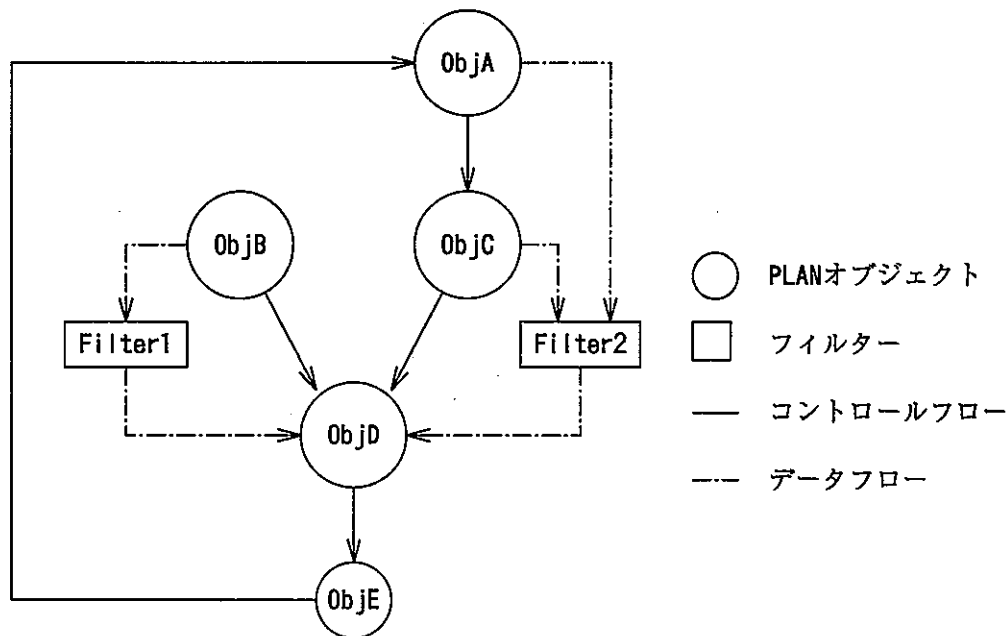


図3.1.1-5 PLAN表現例

最後に、PLANの作成、PLANオブジェクトに関連したデータの入力方法、及び、その入力データだけでなく出力結果も含めたデータの表示方法について述べる。平成4年度以降、PLANの構築や実行、入出力データの参照は、それぞれPLAN構築モジュール、PLAN実行モジュール、出力表示モジュールで実現される機能として検討されてきた。そして、平成5年度には、AVSの制約からPLAN構築モジュール、PLAN実行モジュールを統合して1つのモジュールとして構築し、参照機能だけは別のモジュールとして構築した。この平成5年度システムの操作性を考えると、PLAN構築、実行は1つのインタフェース上で行うことができる便利さはあるが、入力データの作成やデータの参照等は別のインタフェースとなりユーザにとって操作が複雑となっていた。このため、本年度検討のシステムでは、操作性向上の観点から、PLANの構築、実行、及び、各PLANオブジェクトに関連したデータの表示・参照を、全て1つのインタフェース上に実現することとした。

このインタフェースが持つべき機能は以下の通りである。



(1)PLAN構築機能

PLANオブジェクト、コントロールフロー、データフローを配置し、PLANを構築する機能。

(2)PLAN参照機能

PLANを参照する機能

(3)PLAN実行機能

PLANを実行する機能。実行状態が動的にグラフィカルに表現される。

(4)PLANオブジェクト入力データ作成機能

PLANオブジェクトに関連したデータを入力する機能。

(5)PLANオブジェクト入力データ参照機能

PLANオブジェクトに関連した入力データを表示する機能。

(6)PLANオブジェクト出力データ参照機能

PLANオブジェクトに関連した出力データを表示する機能。

図3.1.1-6にインタフェースの概念図を示す。

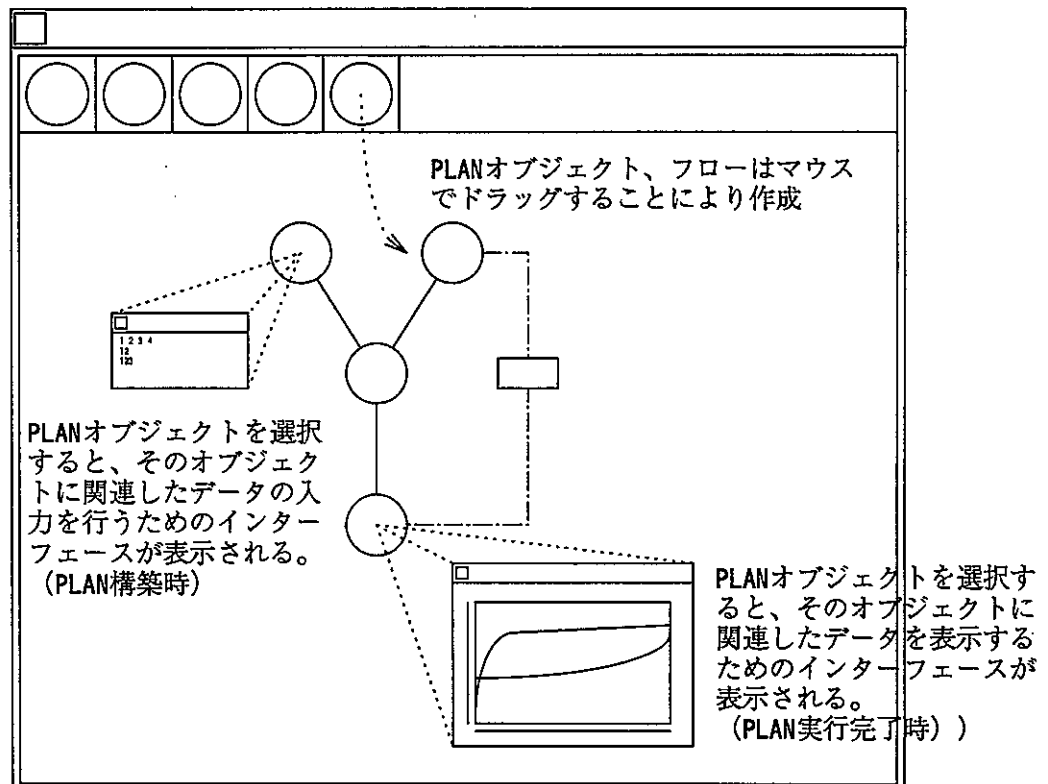


図3.1.1-6 PLAN構築/実行/参照 概念図

以上、データの流を規定したデータフローを取り入れたPLANの表現方法、及び、PLANの構築、実行、参照を効率よく行うことができるインタフェースについて述べた。今後は、さらに詳細な設計を進め、PLAN構築、実行だけでなくCAPASA/PCDEE全体の環境を整えていく必要がある。また、連成計算や新規解析コードの取り込みを行なうため、PLANオブジェクトやデータ変換を行なうフィルターについて検討することにより、広範囲な解析業務を行える環境を準備していく必要がある。

### 3. 1. 2 プロトタイピング

3.1.1節のPLAN表現に従って、各PLANオブジェクトを実行するための手順を検討し、その実現性を検証するためにプロトタイピングを行った。以下に、プロトタイピングを行ったPLAN実行のメカニズムについて述べる。

PLANの実行は、コントロールフローの上流側から下流側へと順次PLANオブジェクトを実行することにより行われる。このため、PLANオブジェクトが実行可能かどうかの判断は、該当するPLANオブジェクトが持つ全てのコントロールフローの上流側のPLANオブジェクトが実行済みかどうかを判定することにより行われる。

ここで、上記の手順を実現するために、以下の2種類の属性をPLANオブジェクトに設定した。

#### (1)トリガフラッグ

PLANオブジェクトが実行可能かどうかを判断するフラッグ。「オン」であれば実行可能とする。値の設定は以下の手順により行われる。

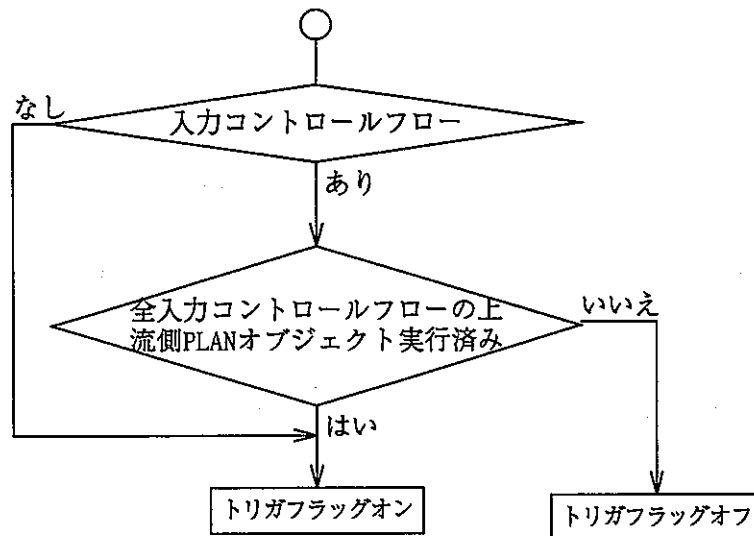


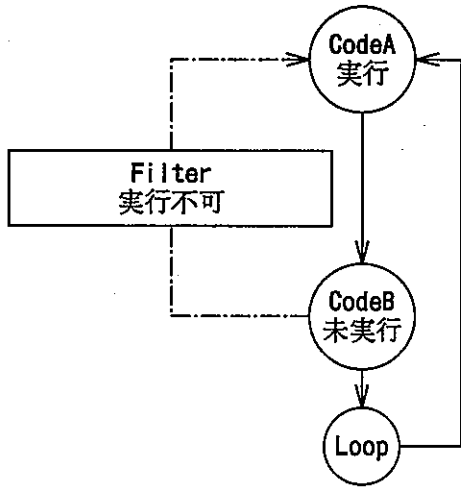
図3.1.2-1 トリガフラッグの判定

#### (2)実行フラッグ

PLANオブジェクトが実行済みかどうかを判断するフラッグ。「オン」であれば実行済みとする。PLAN実行は、すべてのPLANオブジェクトの実行フラッグが「オン」となった時に終了する。

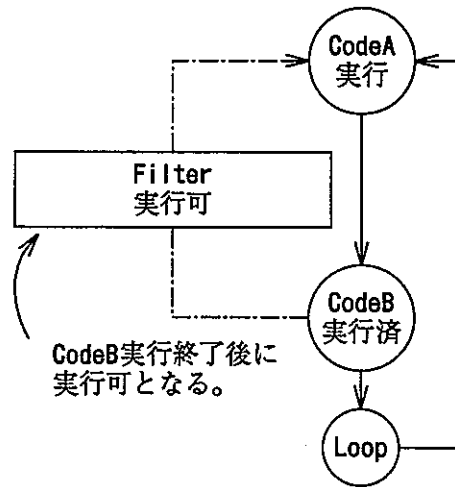
PLANオブジェクトがデータフローをもつ場合は、通常はデータ変換プログラムが実行され、その後、PLANオブジェクトに関連付けられた解析コード等の外部プログラムが実行される。ただし、図3.1.2-2に示すようなループ状のPLANの場合には、最初の「CodeA」実行時には、データ変換プログラム「Filter」は実行されずに、2回目以降からは実行される。また、図3.1.2-3に示すようなPLANの場合には、最初、「Filter」は実行されるが、2回目以降からは実行されない。このような例外処

CodeA実行時に、Filterは実行されない。



1回目のループ

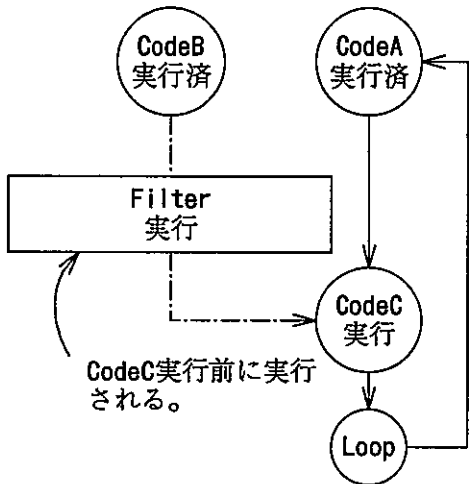
CodeA実行時に、Filterは実行される。



2回目以降のループ

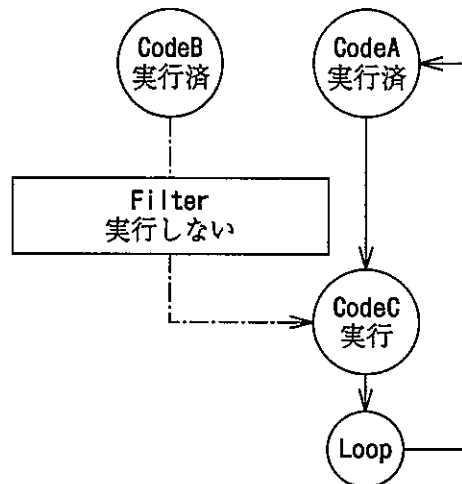
図3.1.2-2 ループ状PLAN (1)

CodeC実行時に、Filterは実行される。



1回目のループ

CodeC実行時に、Filterは実行されない。



2回目以降のループ

図3.1.2-3 ループ状PLAN (2)

理を制御するため、以下の2種類の属性をデータフローに設定した。

(1) アクティブフラッグ

データ変換プログラムが実行可能かどうかを判断するフラッグ。このアクティブフラッグは、図

3.1.2-4に示すようにデータフローの上流側のPLANオブジェクトが全て実行済みとなった時に「オン」となる。そして、PLANオブジェクト実行の際には、データフローの入力がある場合はアクティブフラグの値を調べ、「オン」の時にのみデータ変換プログラムが実行される。

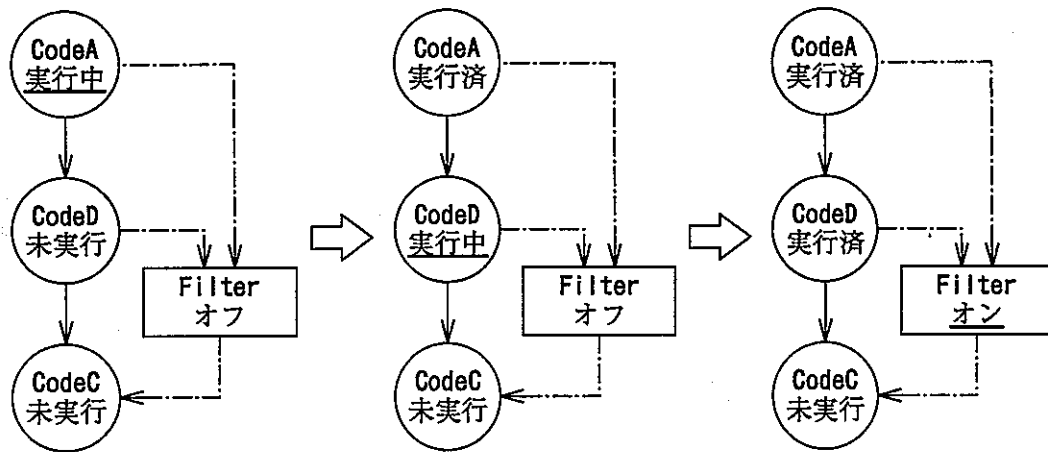


図3.1.2-4 アクティブフラグの設定

## (2) 実行フラグ

データフローがある場合、データ変換プログラムが実行済みかどうかを判断するフラグ。「オン」の場合には、データ変換プログラムは実行されない。

最後にPLAN全体の実行について述べる。PLAN実行は図3.1.2-5に示す手順で行われる。

最初、図3.1.2-1の手順でトリガフラグの判定が行われた後に、トリガフラグがオンのPLANオブジェクトが実行キューに追加される。その後、実行キューから順次、PLANオブジェクトが取り出され実行される。実行の際には、該当オブジェクトがデータフローの入力を持ち、かつ、そのデータフローのアクティブフラグがオンの場合には、外部プログラム実行前にデータ変換プログラムが実行される。実行キューが空になると、データフローのアクティブフラグの設定が行われる。以上の手順を全てのPLANオブジェクトの実行フラグがオンになるまで繰り返すことにより、PLANの実行が行われる。

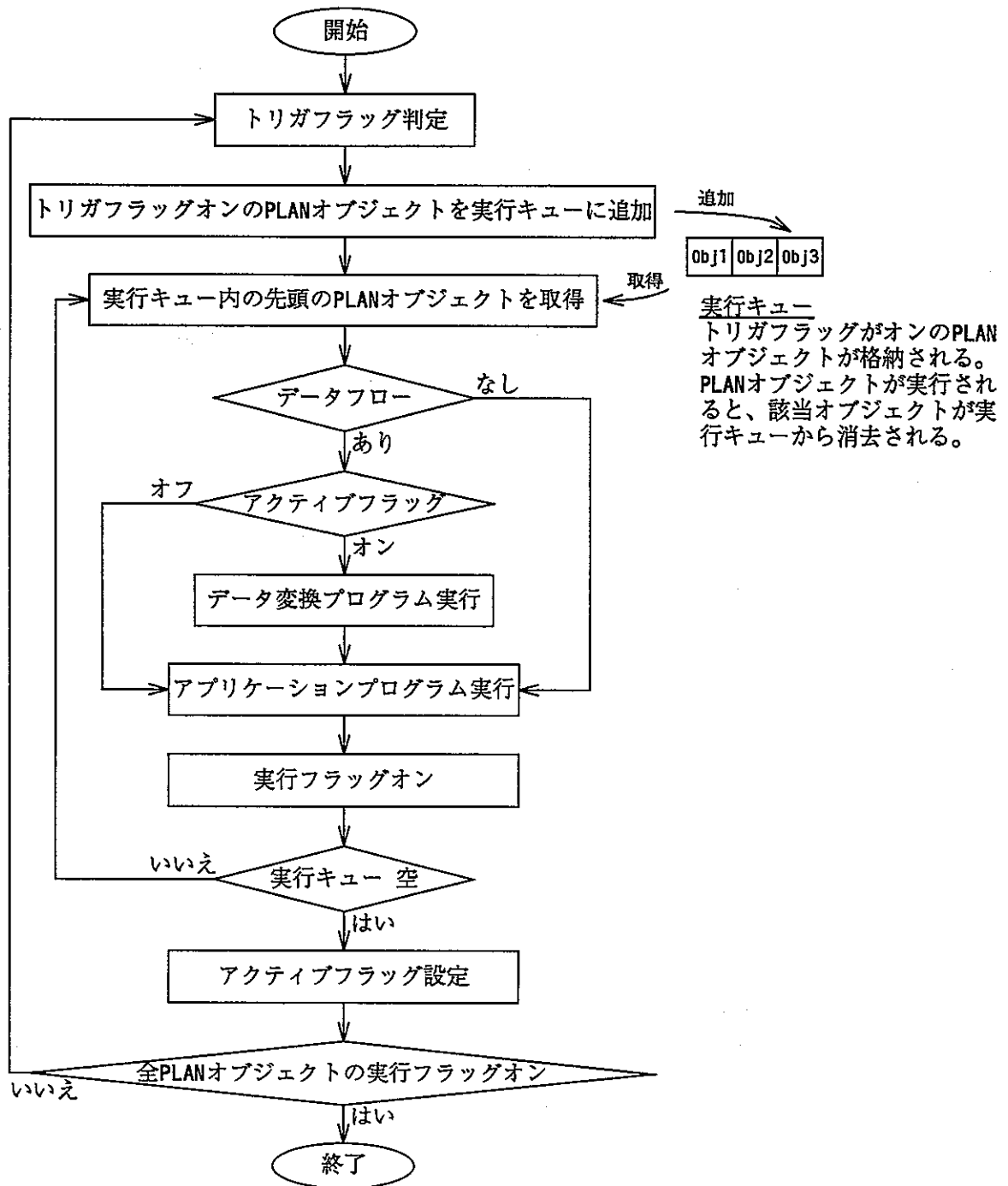


図3.1.2-5 PLANの実行

本年度は上記に示したフローに従ってPLANを実行するプログラム（以下、PLAN実行エンジン）をC++言語を用いてSUN SPARCstation上に製作した。以下にプログラムの概要、及び、実現性、機能性の検討結果について述べる。

PLAN実行エンジンにおいてPLANの各アイテムは以下の4種類のクラスによって表現されている。

・PLANクラス

PLANの中心となるクラス。各アイテムの管理やPLANの実行を制御する。

```
class PLAN {  
    public:  
        Boolean Execute(void);          図3.1.2-5の手順でPLANを実行する。  
    private:  
        char* name;                     PLANの名称  
        Array<PLANObject*> planObjs;    PLANObjectインスタンスが格納される配列  
        Array<ControlFlow*> cntIFlows;  ControlFlowインスタンスが格納される配列  
        Array<DataFlow*> dataFlows;    DataFlowインスタンスが格納される配列  
        void initTrigger(void);        トリガフラッグを初期状態に設定する。  
        void setTrigger(void);         図3.1.2-1の手順でトリガフラッグを判定する。  
        Boolean execute(void);        PLANオブジェクトを実行する。  
        Boolean isDone(void);         全てのPLANオブジェクトが実行済みかどうかを判定する。  
        void setLoop(PLANObject*, PLANObject*);  ループの設定を行う。  
};
```

・PLANObjectクラス

PLANオブジェクトを表わしたクラス。属性値の格納やPLANオブジェクトの実行を制御する。

```
class PLANObject {  
    public:  
        Boolean Execute(void);        外部プログラムを実行する。  
    private:  
        char* objName;                PLANオブジェクトの名称  
        char* execName;               外部プログラムの名称  
        PLANObjectType objType;       PLANオブジェクトのタイプ  
        ExecFlag execFlag;            実行フラッグ  
        TriggerFlag trigFlag;         トリガフラッグ  
        TriggerFlag initTrigFlag;     初期トリガフラッグ  
        PLAN* toPLAN;                 PLANインスタンス  
        Array<ControlFlow*> inControl;  入力しているコントロールフローの配列  
        Array<ControlFlow*> outControl;  出力しているデータフローの配列  
        Array<DataFlow*> inData;        入力しているコントロールフローの配列  
        Array<DataFlow*> outData;       出力しているデータフローの配列  
};
```

・ ControlFlowクラス

コントロールフローを表わしたクラス。

```
class ControlFlow {
    public:
    private:
        PLANObject* toUp;    上流側のPLANオブジェクト
        PLANObject* toLow;   下流側のPLANオブジェクト
        PLAN* toPLAN;       PLANインスタンス
};
```

・ DataFlowクラス

データフローを表わしたクラス。属性値の格納やフィルタ（データ変換プログラム）の実行を制御する。

```
class DataFlow {
    public:
        Boolean Execute(void);    フィルターを実行する。
    private:
        Array<PLANObject*> toUp;   上流側のPLANオブジェクトの配列
        PLANObject* toLow;         下流側のPLANオブジェクト
        char* execName;           フィルターの名称
        PLAN* toPLAN;             PLANインスタンス
        ExecFlag execFlag;       実行フラッグ
        ActiveFlag actFlag;      アクティブフラッグ
};
```

次にPLAN実行エンジンの検討結果について述べる。検討に用いたPLANは図3.1.2-6に示すように9個のPLANオブジェクト、及び、4個のフィルタにより構成されている。各PLANオブジェクト、フィルタの機能は以下の通りである。

・ InitTimeオブジェクト

解析時間の初期設定を行う。

・ IncTimeオブジェクト

解析時間を時間ステップ分進める。



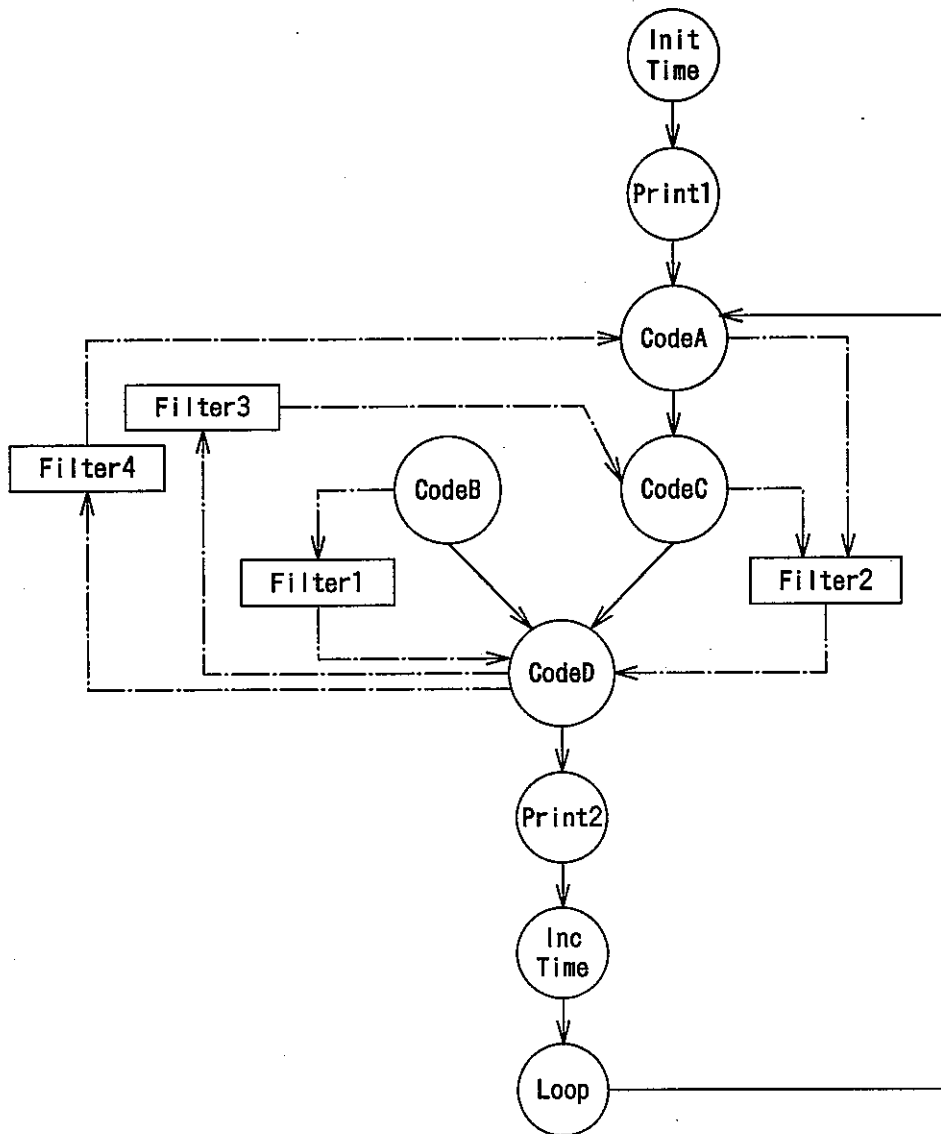


図3.1.2-6 サンプルPLAN

・ Print1オブジェクト、Print2オブジェクト

CodeA、CodeB、CodeC、CodeDのデータの内容を表示する。

・ Loopオブジェクト

現在の解析時間が解析終了時間を満たしているかどうかを判定する。満たしている場合は解析を終了し、満たしていない場合は、ループに依存したPLANオブジェクト (CodeA、CodeC、CodeD、Print2、IncTime、Loop) のトリガフラグと実行フラグ、及び、データフロー (Filter2、Filter3、Filter4) のアクティブフラグと実行フラグを以下に示した規則に従い、再設定する。

Loopオブジェクトからの制御が入力しているオブジェクト (CodeA) のトリガフラグはオンとなる。それ以外のPLANオブジェクトのトリガフラグはオフとなる。

Loopに依存した全てPLANオブジェクトの実行フラグはオフとなる。

順方向の制御の流れ (CodeA、CodeC、...) と同一のデータの流れたLoopに依存したデータフロー (Filter2) のアクティブフラグはオフとなる。

Loopに依存した全てのデータフローの実行フラグはオフとなる。

- ・ CodeAオブジェクト

入力ファイルに設定されている値を2倍する。

- ・ CodeBオブジェクト

乱数を発生する。

- ・ CodeCオブジェクト

入力ファイルに設定されている値を0.5倍する。

- ・ CodeDオブジェクト

入力ファイルに設定されている2つの値を足し算する。

- ・ Filter1データ変換

CodeBの出力ファイルの値をCodeDの入力ファイルの1番目のフィールドに設定する。

- ・ Filter2データ変換

CodeAの出力ファイルの値とCodeCの出力ファイルの値を乗算し、CodeDの入力ファイルの2番目のフィールドに設定する。

- ・ Filter3データ変換

CodeDの出力ファイルの値をCodeCの入力ファイルに設定する。

- ・ Filter4データ変換

CodeDの出力ファイルの値をCodeAの入力ファイルに設定する。

図3.1.2-7に図3.1.2-6のPLANのデータの流れたについて示す。CodeAは、入力ファイルの値を読み、その値を2倍した値を出力ファイルに書き出す。CodeBは乱数を発生し、出力ファイルに書き出す。CodeCは、入力ファイルの値を読み、その値を0.5倍した値を出力ファイルに書き出す。CodeDは、CodeBの出力値とCodeAの出力値とCodeCの出力値を加算した値が書かれた入力ファイルを読み込み、この2つの値を乗算した値を出力ファイルに書き出す。そして、この値はCodeA、CodeCの入力ファイルに設定される。

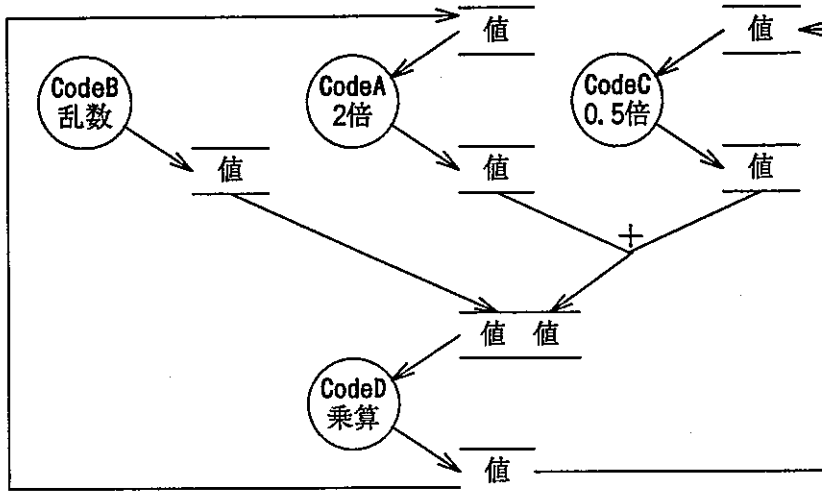


図3.1.2-7 サンプルPLANにおけるデータの流れ

最後に、PLAN実行エンジンでのPLAN実行結果の概要を示す。なお、初期値として、CodeAの入力値は1、CodeCの入力値は2とする。解析時間は、開始時間を0、終了時間を2、時間ステップを1とする。また、出力結果に添付しているPLANの図は以下の規則による。

PLANオブジェクト

- トリガフラッグ オフ    実行フラッグ オフ
- トリガフラッグ オン
- 実行フラッグ オン

データフロー

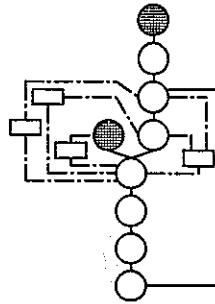
- アクティブフラッグ オフ    実行フラッグ オフ
- ▨ アクティブフラッグ オン
- 実行フラッグ オン

実行結果

===== Loop.1 =====

Name	Trigger	Execution
InitTime	On	Not
Print1	Off	Not
CodeA	Off	Not
CodeB	On	Not
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

・トリガフラッグ設定  
InitTime、CodeB トリガフラッグ オン



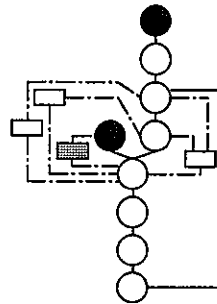
Name	Active	Execution
Filter1	Off	Not
Filter2	Off	Not
Filter3	Off	Not
Filter4	Off	Not

<InitTime>  
start = 0.00 end = 2.00 step = 1.00  
<CodeB>  
random = 0.3

InitTime 実行  
CodeB 実行

・実行フラッグ、アクティブフラッグ設定  
InitTime、CodeB 実行フラッグ オン  
Filter1 アクティブフラッグ オン

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Not
CodeA	Off	Not
CodeB	Off	Done
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

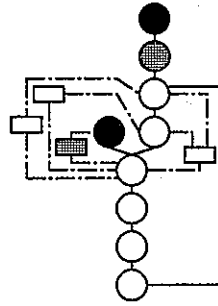


Name	Active	Execution
Filter1	On	Not
Filter2	Off	Not
Filter3	Off	Not
Filter4	Off	Not

===== Loop. 2 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	On	Not
CodeA	Off	Not
CodeB	Off	Done
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

・トリガフラッグ設定  
Print1 トリガフラッグ オン



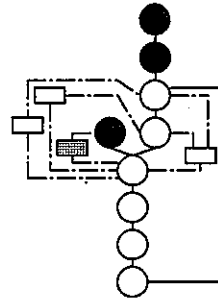
<Print>

Print 実行

Time : 0.00 -> 2.00 ( 1.00) 0.00			
CodeA	In = 1.00		
CodeB		Out = 0.30	
CodeC	In = 2.00		
CodeD			

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Not
CodeB	Off	Done
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

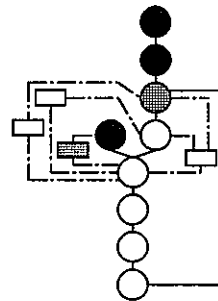
・実行フラッグ設定  
Print1 実行フラッグ オン



===== Loop. 3 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	On	Not
CodeB	Off	Done
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

・トリガフラッグ設定  
CodeA トリガフラッグ オン

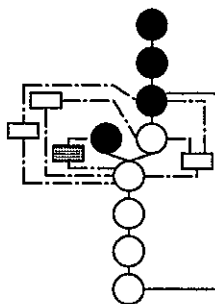


<CodeA>  
 $1.00 \times 2 = 2.00$

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

CodeA 実行

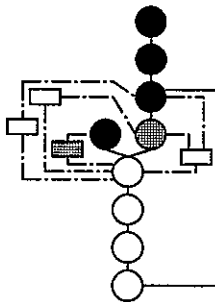
- ・実行フラッグ、アクティブフラッグ設定  
 CodeA 実行フラッグ オン



==== Loop. 4 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	On	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

- ・トリガフラッグ設定  
 CodeC トリガフラッグ オン

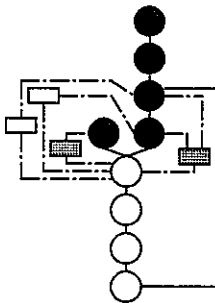


<CodeC>  
 $2.00 \times 0.5 = 1.00$

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

CodeC 実行

- ・実行フラッグ、アクティブフラッグ設定  
 CodeC 実行フラッグ オン  
 Filter2 アクティブフラッグオン

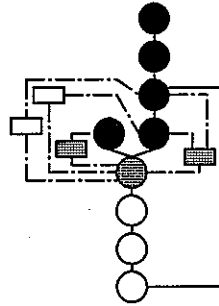


Name	Active	Execution
Filter1	On	Not
Filter2	On	Not
Filter3	Off	Not
Filter4	Off	Not

===== Loop. 5 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	On	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

・トリガフラッグ設定  
CodeD トリガフラッグ オン



<Filter1>

CodeB = 0.30 --> CodeD = 0.00->0.30 0.00

Filter1 実行

<Filter2>

CodeA = 2.00 CodeC = 1.00 -> CodeD = 0.30 0.00->3.00

Filter2 実行

<CodeD>

0.30 x 3.00 = 0.90

CodeD 実行

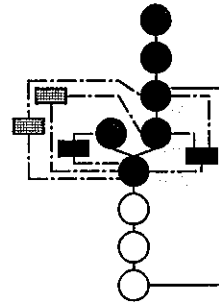
・実行フラッグ、アクティブフラッグ設定

CodeD 実行フラッグ オン

Filter1、Filter2 実行フラッグ オン

Filter3、Filter4 アクティブフラッグ オン

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Done
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

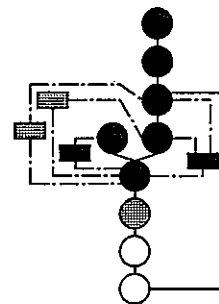


Name	Active	Execution
Filter1	On	Done
Filter2	On	Done
Filter3	On	Not
Filter4	On	Not

===== Loop. 6 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Done
Print2	On	Not
IncTime	Off	Not
Loop	Off	Not

・トリガフラッグ設定  
Print2 トリガフラッグ オン



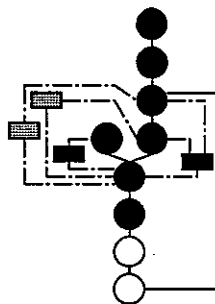
<Print>

Print 実行

Time : 0.00 -> 2.00 ( 1.00) 0.00			
CodeA	In = 1.00		Out = 2.00
CodeB			Out = 0.30
CodeC	In = 2.00		Out = 1.00
CodeD	In = 0.30 3.00		Out = 0.90

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Done
Print2	Off	Done
IncTime	Off	Not
Loop	Off	Not

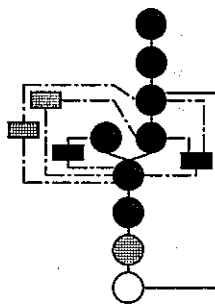
・実行フラッグ  
Print2 実行フラッグ オン



===== Loop. 7 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Done
Print2	Off	Done
IncTime	On	Not
Loop	Off	Not

・トリガフラッグ設定  
IncTime トリガフラッグ オン



<IncTime>

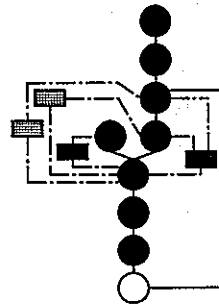
start = 0.00  
end = 2.00  
step = 1.00  
current = 0.00 -> 1.00

IncTime 実行



Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Done
Print2	Off	Done
IncTime	Off	Done
Loop	Off	Not

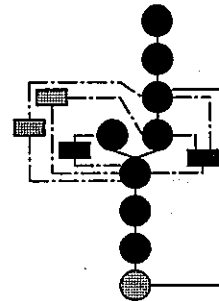
・実行フラッグ設定  
IncTime 実行フラッグ オン



==== Loop. 8 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Done
Print2	Off	Done
IncTime	Off	Done
Loop	On	Not

・トリガフラッグ設定  
Loop トリガフラッグ オン

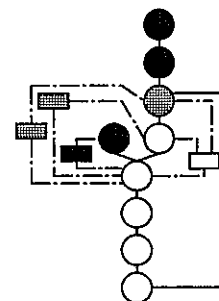


<Loop>

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	On	Not
CodeB	Off	Done
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

Loop 実行

・ループによる初期化  
ループに含まれるアイテムCodeA、CodeC、CodeD、Print、IncTime、Loop、Filter1、Filter2、Filter3、Filter4が初期化される。

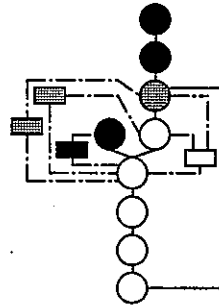


Name	Active	Execution
Filter1	On	Done
Filter2	Off	Not
Filter3	On	Not
Filter4	On	Not

==== Loop. 9 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	On	Not
CodeB	Off	Done
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

・トリガフラッグ設定  
CodeA トリガフラッグ オン



<Filter4>

CodeD = 0.90 --> CodeA = 1.00 -> 0.90

<CodeA>

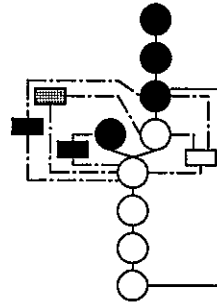
0.90 x 2 = 1.80

Filter4 実行

CodeA 実行

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

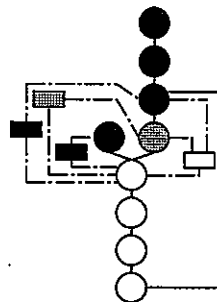
・実行フラッグ設定  
CodeA 実行フラッグ オン  
Filter4 実行フラッグ オン



==== Loop. 10 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	On	Not
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

・トリガフラッグ設定  
CodeC トリガフラッグ オン



<Filter3>

CodeD = 0.90 --> CodeC = 2.00 -> 0.90

<CodeC>

Filter3 実行

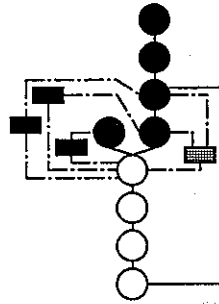
CodeC 実行

$$0.90 \times 0.5 = 0.45$$

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

Name	Active	Execution
Filter1	On	Done
Filter2	On	Not
Filter3	On	Done
Filter4	On	Done

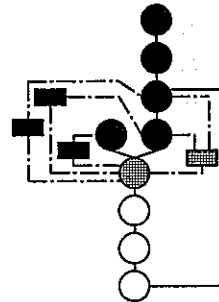
- ・実行フラッグ、アクティブフラッグ設定
- CodeC 実行フラッグ オン
- Filter2 アクティブフラッグ オン
- Filter3 実行フラッグ オン



===== Loop.11 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	On	Not
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

- ・トリガフラッグ設定
- CodeD トリガフラッグ オン



<Filter2>

$$\text{CodeA} = 1.80 \quad \text{CodeC} = 0.45 \rightarrow \text{CodeD} = 0.30 \quad 3.00 \rightarrow 2.25$$

<CodeD>

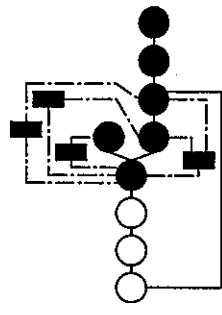
$$0.30 \times 2.25 = 0.67$$

Filter2 実行

CodeD 実行

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Done
Print2	Off	Not
IncTime	Off	Not
Loop	Off	Not

・実行フラッグ、アクティブフラッグ設定  
 CodeD 実行フラッグ オン  
 Filter2 実行フラッグオン

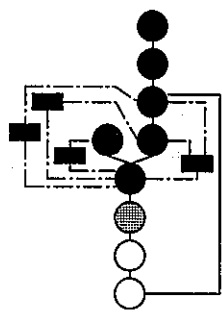


Name	Active	Execution
Filter1	On	Done
Filter2	On	Done
Filter3	On	Done
Filter4	On	Done

===== Loop. 12 =====

Name	Trigger	Execution
InitTime	Off	Done
Print1	Off	Done
CodeA	Off	Done
CodeB	Off	Done
CodeC	Off	Done
CodeD	Off	Done
Print2	On	Not
IncTime	Off	Not
Loop	Off	Not

・トリガフラッグ設定  
 Print2 トリガフラッグ オン



<Print>

Print 実行

Time : 0.00 -> 2.00 ( 1.00) 1.00			
CodeA	In =	0.90	Out = 1.80
CodeB			Out = 0.30
CodeC	In =	0.90	Out = 0.45
CodeD	In =	0.30 2.25	Out = 0.68

以下繰り返し

上記の動作例で示したように、本年度製作したPLAN実行エンジンのプロトタイプは、問題なく動作することが確認できた。

### 3. 1. 3 モジュールの設計

PLAN構築／実行モジュールの高度化仕様に基づき、プロトタイピングの成果を踏まえて、PLAN構築／実行モジュールの設計を行った。

3.1.1節で述べたように、PLANオブジェクトはコントロールフロー、データフローの2種類のフローで接続される。PLANの表示の際に、これらのフローやPLANオブジェクトを同時に表示することは、情報量が多くなり、PLAN全体の流れが理解しづらくなる。また、PLAN作成の際にも、これらのアイテムを同時に作成対象とすることは、操作が複雑になる。

このため、PLAN構築／実行モジュールは、操作性や理解力の向上のため、表示・編集対象となるアイテムに対応したレイヤーや操作モードを用意し、これをユーザが自由に切り替えることにより、PLANを構築、実行するものとした。

レイヤーは以下の3種類が用意される。

- (1)PLANオブジェクトレイヤー           PLANオブジェクトを対象するレイヤー。
- (2)コントロールフローレイヤー       コントロールフローを対象するレイヤー。
- (3)データフローレイヤー             データフローを対象するレイヤー。

また、レイヤー毎に以下の操作モードが用意される。

- (1)インアクティブモード    該当レイヤーが対象としているアイテムを非表示とするモード。
- (2)表示モード                該当レイヤーが対象としているアイテムの表示のみを行うモード。
- (3)編集モード                該当するレイヤーが対象としているアイテムを編集、表示するモード。

ユーザはこれらのレイヤー、操作モードを以下の制約下で操作することにより、PLANの表示、編集を行う。

- ・PLANオブジェクトレイヤーをインアクティブモードにすることはできない。
- ・各レイヤーのモードは以下のケースを除いて、独立に設定される。

編集モードは各レイヤーの中で1つのみが選択可能である。

例)

レイヤー	PLANオブジェクト	コントロールフロー	データフロー
操作モード	表示	インアクティブ	編集
操作モード	表示	編集	表示

ユーザが編集モードに変更  
 ↓  
 自動的に表示モードに変更

表3.1.3-1に操作モードの組み合わせパターンを示す。

表3.1.3-1 操作モード組み合わせパターン

PLANオブジェクト	コントロールフロー	データフロー
表示	インアクティブ	インアクティブ
表示	インアクティブ	表示
表示	インアクティブ	編集
表示	表示	インアクティブ
表示	表示	表示
表示	表示	編集
表示	編集	インアクティブ

・コントロールフローの両端のPLANオブジェクトが削除された場合は、コントロールフローは自動的に削除される。データフローについても同様である。

PLAN構築／実行モジュールのインターフェースについて述べる。平成5年度のシステムは、PLAN構築／実行モジュールは市販アプリケーションAVS上に構築されたため、各アイテムの表示形式や属性の入力方法等に多くの制約があった。今年度設計のシステムは、この制約のとらわれることなく、拡張性や操作性の向上の観点から、インタフェースの設計を行った。

PLANの構築方法について述べる。PLANの構築は最初にレイヤーを「PLANオブジェクト」、操作モードを「編集」として、図3.1.3-1に示すように、PLANオブジェクトエリアから対象となるPLANオブ

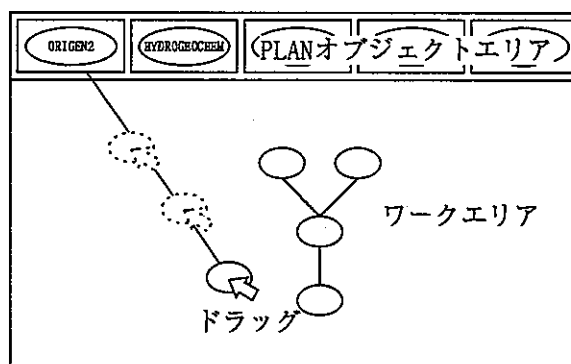


図3.1.3-1 PLAN構築イメージ

ジェクトをマウスでドラッグし、ワークエリアに配置する。なお、PLANオブジェクトは機能別に以下に示すように表示色が決定される。

- ・解析コードオブジェクト 赤色
- ・ユーティリティオブジェクト 青色

コントロールフローの編集は、レイヤーを「コントロールフロー」、操作モードを「編集」として、最初にフローの上流のPLANオブジェクトを選択し、次に下流のPLANオブジェクトを選択することにより行う。データフローの作成もコントロールフローと同様な方法で行う。

属性値や入力データ等の設定は編集対象となるアイテムを選択し、そのアイテム固有のインタフェースを表示する。そして、このインタフェースを通して入力作業を行う。なお、この入力作業はPLAN構築時だけでなく、PLAN実行時でも入力が可能である。ただし、実行が完了したPLANオブジェクトに関してはデータの表示のみが可能となる。

PLANの実行は、構築時と同じインタフェース上で行う。その際、以下の規則に従って、PLANオブジェクトの表示色が決定される。

- ・データ不完全 黄色

PLANオブジェクト実行のために必要なデータ（入力データ等）が入力されていない場合。データ不完全なオブジェクトにPLANの実行制御が移された場合は、実行が一時休止される。

- ・準備完了 青色

PLANオブジェクトの実行のために必要なデータが入力されている場合。制御が該当オブジェクトに移った際に実行中となる。

- ・実行中 赤色

実行中の場合。

- ・実行正常終了 緑色

実行が正常に終了した場合

- ・実行異常終了 紫色

実行が異常終了した場合

PLANの実行が完了すると、全てのPLANオブジェクトの色が実行前の状態に戻る。この状態で、PLANオブジェクトを選択することにより、入出力データや属性データを参照することができる。

### 3. 1. 4 まとめ

本年度は、より柔軟なPLAN表現を行なうためにデータフローの概念を取り入れPLAN構築/実行モジュールの高機能化に関する検討を行った。そして、その実現性、機能性を確認するため、PLAN実行エンジンのプロトタイピングを行ない、実装上、大きな問題がないことを確認した。

今後は、分散環境上に対応したPLAN実行エンジンの実現性や拡張性について、さらに以下の点に関して検討を進める必要がある。

- ・分散環境対応のため、実行キューに蓄えられたPLANオブジェクトを並列に実行する。
- ・拡張を容易にするため、「Loop」等のユーティリティオブジェクトをPLAN実行エンジンから独立した形態で実装する。（現状では、PLAN実行エンジン内で「Loop」オブジェクトの機能を実現している。）



### 3. 2 分散処理環境に関する検討

複数の現象解析コードを連成した大型の解析を効率的に行なう方法として、分散処理手法の導入が考えられる。本節では、連成解析システムのクライアント/サーバ環境、分散処理環境への拡張に関して検討を行った。

#### 3. 2. 1 クライアント/サーバ形式での利用

効率的に大規模な解析業務を行うには、図3.2.1-1に示すようなネットワークで接続された複数の計算機上で同時に複数の解析者が作業を行う必要がある。

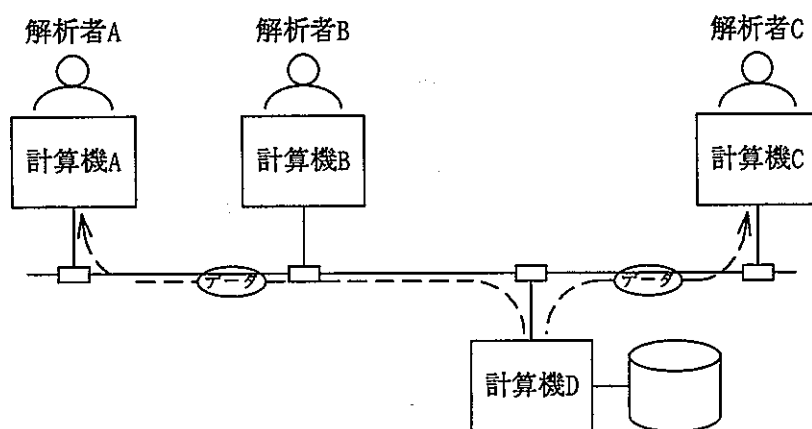


図3.2.1-1 グループワークを支援する計算機環境

このような環境を構築する際にはデータの共有化や計算機資源の共有化といった問題が生じる。本節では、特にデータの共有化に焦点をあて、その実現方法について述べる。

データの整合性の管理や計算機のデータの送受信を行うには、クライアント/サーバ対応のデータベース管理システム (DBMS) を利用する。クライアント/サーバ対応DBMSとはクライアントからの要求に従って、サーバが処理を行い、その結果をクライアントに再び返すことを基本動作にしたモデルである。検索処理を例にとり、クライアント、サーバでの処理について述べる。図3.2.1-2に示すように、クライアントは、ユーザにより入力された検索条件をサーバに転送する。サーバでは、この検索条件に従いデータを検索し、その結果をクライアントに転送する。そして、クライアントでは、この検索結果を表示する。

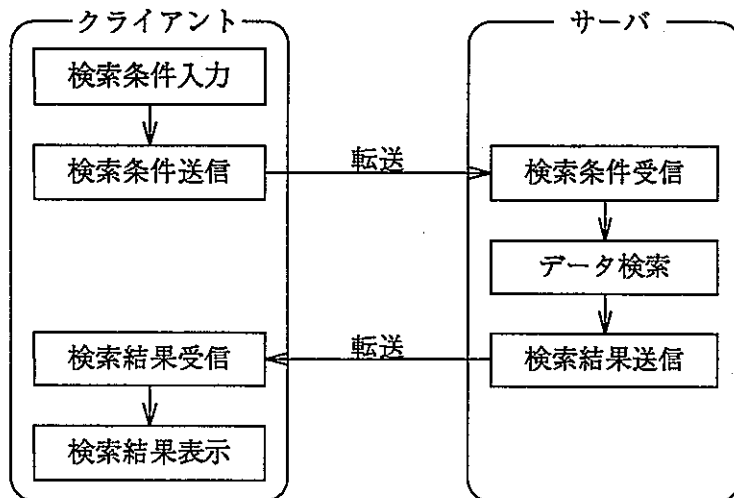


図3.2.1-2 クライアント/サーバ対応DBMS 検索フロー

クライアントとサーバの処理内容の違いから分かるように、検索条件の入力や検索結果の表示等のユーザとのコミュニケーションはクライアント側で行なわれ、サーバ側では、データの検索のみが行われる。よって、クライアント/サーバモデルを実現するには、クライアントは、比較的安価なグラフィック表示能力の優れたコンピュータを、サーバでは、データ検索に優れたコンピュータといったように、コストパフォーマンスの優れたシステムを構築することができる。また、クライアント/サーバ形式の特徴として、処理と通信が完全に分かれているため、サーバ側のコンピュータの機種に依存せずにクライアント側の機種が決定することができ、柔軟性に富んだシステムを構築することができる。

次にクライアントとサーバの必要とする機能について詳細に述べる。

#### クライアントの機能

クライアントは、ユーザとのコミュニケーションを行なうため、以下の機能が要求される。

##### (1)サーバへの要求データの送受信

サーバとの間でデータの送受信をユーザに意識されることなく実施する。

##### (2)サーバへの要求データの作成/処理結果の表示

解析条件の入力や結果の表示を行うため、ユーザにとって使いやすいインタフェースを提供する。

#### サーバの機能

サーバは、ユーザとの直接対話には用いられずに、データ管理のみが行なわれる。

##### (1)データの一元管理、排他制御

データの二重化防止や整合性を管理する。

## (2)データの送受信処理

要求データや処理結果の送受信処理を行う。

## (3)コミュニケーションサービス

ネットワーク上でのホスト環境の管理やルーティング機能等を行う。

## (4)セキュリティサービス

共有されるリソースへのアクセスを制御するために、ユーザIDとパスワードを管理する。

## (5)マネージメントサービス

ロギング機能により、リソースの使用状況を管理する。

これらの機能を実現するためには、以下に示した点を考慮し、システムを設計しなければならない。

### (1)データベース管理システム

クライアント側、サーバ側で必要とされる機能の大部分はDBMSが持つべき機能であるため、DBMSの選定が重要な要素となる。具体的には、DBMSがサポートする機種種の豊富さやユーザ管理の方法等。

### (2)ネットワーク

データの送受信の量に見合ったネットワークの転送能力が必要となる。特にWANを利用する場合には、この点を考慮しないとクライアントとしての十分な機能を発揮できなくなる。

### (3)サーバ側コンピュータ

ディスクの容量、コンピュータ/ディスク間のスループット、演算能力、対障害性や障害時の復旧の容易さ等を考慮する必要がある。

### (4)クライアント側コンピュータ

ウインドウやグラフィックの表示能力。

以上の点を考慮したシステムの例を図3.2.1-3に示す。サーバはディスクI/O性能や拡張性に優れたSUN SPARCserverを用いる。ディスク装置は高速にデータをアクセスすることが可能なディスクアレイ装置を用いる。これは、比較的容量の少ないディスクを複数接続し、仮想的な大容量のディスクとしたものである。また、バックアップ装置としては、大容量のデータを高速に格納することができるDATを採用した。クライアントはグラフィック表示能力に優れたSUN SPARCstationやSGI Indyを選択した。また、グラフィック表示能力は前者に劣るが、コストパフォーマンスが優れているCompaq等のパーソナルコンピュータも限られた範囲では有効であると考えられる。

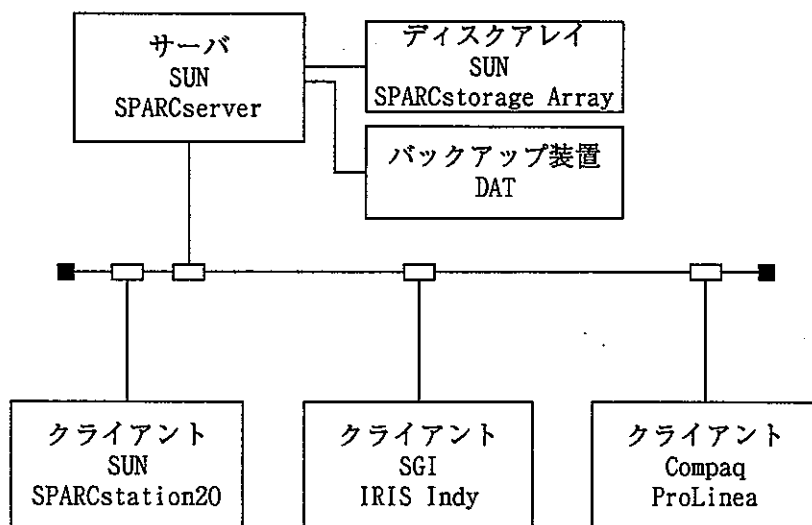


図3.2.1-3 クライアント/サーバ形式システム 構築例

### 3. 2. 2 マルチCPUによる分散処理環境

マルチCPUによる分散処理環境を検討するために、最初にマルチCPUシステムの定義を行う。マルチCPUシステムとは、独立に別々の演算を行うことができる複数のCPUを、バス、クロスバースイッチ等のインターコネクションネットワーク、あるいは、共有メモリを介して接続したものである。つまり、マルチCPUシステムとはユーザが自由に複数のCPUを利用でき、どのCPUもほぼ同様に汎用的にプログラムできるものであって、一つの仕事（目的）を互いに情報の交換を行いながら、実行できる計算機と定義できる。従って、イーサネットを利用して個々のシステムを接続し、二つのプロセスがデータを交換しながら処理が進むとき、これもマルチCPUシステムと呼ぶことができる。よって、CAPASAが対象とする分散処理環境は、図3.2.2-1に示すように、単体のCPUを持ったワークステーション、複数のCPUを持ったワークステーションと複数のCPUを持った並列計算機がイーサネット等のネットワークで接続された形態とする。

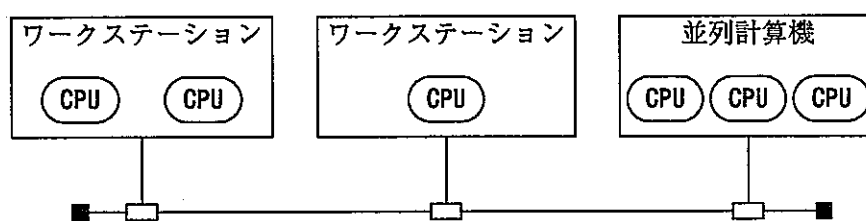


図3.2.2-1 分散処理ハードウェア環境

分散処理環境上にCAPASAを構築する際に必要となるハードウェア的な側面について述べる。CAPASAの機能は4種類に大別され、それぞれ必要とする計算機の能力が異なる。

#### (1) PLAN実行機能

PLANを実行する機能。浮動小数点演算が高速な計算機や並列計算機が適する。

#### (2) データベースのアクセス機能

データベースへのデータの格納、検索を行う機能。ディスクI/Oのスループットが優れた計算機が適する。通常、サーバーマシンと呼ばれるものがこれにあたる。

#### (3) 解析結果表示機能

解析結果を視覚化する機能。グラフィック性能が優れた計算機が適する。

#### (4) 入力データ作成機能

入力データ等を作成する機能。優れたデスクトップ環境を持つ計算機が適する。

これらの中で最もCPUの能力を必要とし、分散環境における長所が得られるPLAN実行機能について、詳しく述べる。PLAN実行機能は、個々のPLAN全体の実行とPLANオブジェクト（解析コード）の実行の2つの部分に分けられる。PLANの実行は、多くの部分で並列化できる要素を含んでおり、分散環境での実行に適している。例えば、図3.2.2-2に示したPLANでは、プロセスAとプロセスBは同時

に実行することができるため、これらのPLANオブジェクトの実行を、分散環境下で別々のCPUで実行することにより、全体のPLANの実行を単体の計算機で行うより短時間で実行することができる。

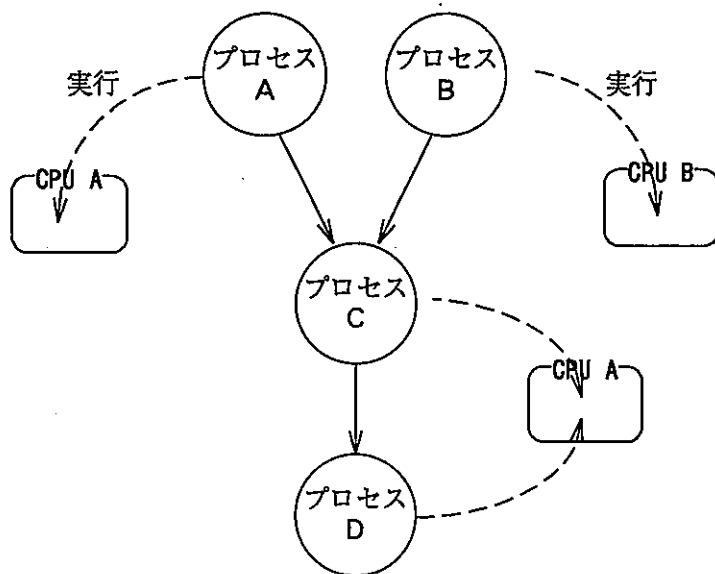


図3.2.2-2 分散環境におけるPLAN実行

上記のような単純なPLANでは、分散環境下での実行は容易であるが、複雑なPLANではPLANオブジェクト実行ためのスケジューリングが必要となってくる。例えば、図3.2.2-3に示すように、n個の並列可能（同時実行可能）なPLANオブジェクトとn個以下のCPUが存在する場合はこの例である。こ

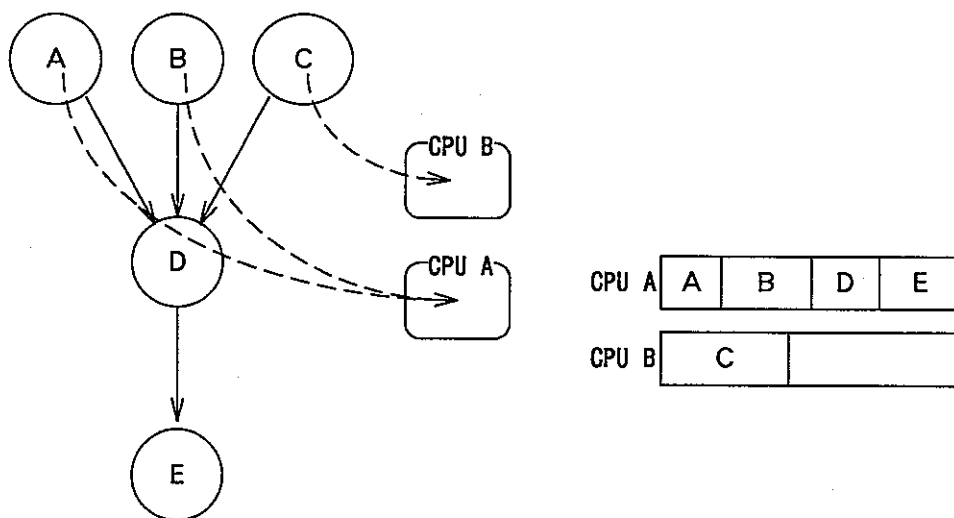


図3.2.2-3 スケジューリングが必要となるPLAN例

の場合スケジューリングを行うには、個々のPLANオブジェクトのコスト（実行時間）の算出が必要となる。コストの算出が可能であれば、ヒューリスティックスケジューリング等を行うことにより、最適解に近い解を求めることができる。コスト算出の方法としては、ユーザによる指定や、解析コードを実行するPLANオブジェクトの場合は過去の実行時間や計算時間に影響するパラメータから予測することが考えられるが、正確な実行時間を求めることは、困難である。

の解析コードの分散環境下での実行について述べる。単体の計算機で実行することを前提として作成された解析コードをソースプログラムの変更なしで、ネットワークで接続された複数の計算機で実行することは不可能である。よって、解析コードは並列計算機等の高速な単体の計算機で実行しなければならない。ここで、解析コード等のプログラムを並列に処理する際に必要となる概念にタスク粒度というものがある。このタスク粒度は概念的に3種類に分類される。

#### (1)粗粒度並列処理

サブルーチン等の比較的大きな処理ブロックを並列処理するもので、プロセッサ間通信や同期処理等の並列処理特有の無駄時間に比べ、タスクの大きさが十分に大きく、効率よく処理できる

#### (2)中間粒度並列処理

粗粒度よりももう少し細かく砕いたタスクを処理する方式で、DOループの繰返し間の並列性の利用等がある。

#### (3)細粒度並列処理

1命令単位、あるいはそれに近いごく細かい粒度の並列処理を指す。最も理想的な方法であるが、完全に実現することは困難である。簡単に実現できる例として、スーパースカラ等があげられる。

通常、並列化はユーザがソースプログラムを変更するか、または、並列計算機用のコンパイラを用いることにより実現される。コンパイラを利用した場合には、粒度が比較的細かい並列化が行われる。例えば、スーパースカラの技術を採用したSUN SPARCstation20上で解析コードをコンパイルすることにより、スーパースカラに適した実行プログラムが生成され、高速に実行することができる。また、並列計算機専用のコンパイラを用いることにより、DOループ等の並列化が自動的に行われる。

CAPASAでは、オリジナルの解析コードに変更を加えないことを原則としているため、解析コードを変更なしで高速に実行するには、解析コードを実行する並列計算機専用のコンパイラでコンパイルすることにより、高速化することが可能である。

### 3. 2. 3 PNC計算機環境での運用の検討

3.2.2節で述べたように、CAPASAの機能は4種類に大別され、それぞれ必要とする計算機の能力が異なる。このことを考慮しPNC計算機環境でのCAPASAの実装方法を検討した。なお、検討は以下の項目に重点をおいて行った。

- ・オープン性  
異機種、異オペレーティングシステムに対応する。
- ・柔軟性／拡張性  
機器構成の変更や拡張に対応する。
- ・スケーラビリティ  
機器構成の増強に従って、計算能力が向上する。
- ・ネットワーク負荷の軽減  
ネットワークの負荷を軽減することにより、WANに対応する。
- ・制約の軽減  
市販アプリケーションを使うことによる制約を軽減する。

図3.2.3-1にPNC計算機環境下でのCAPASAの実装例を示す。ただし、データベース管理システムを実装する計算機として、SUN SPARCserverを新規に導入することを想定した。

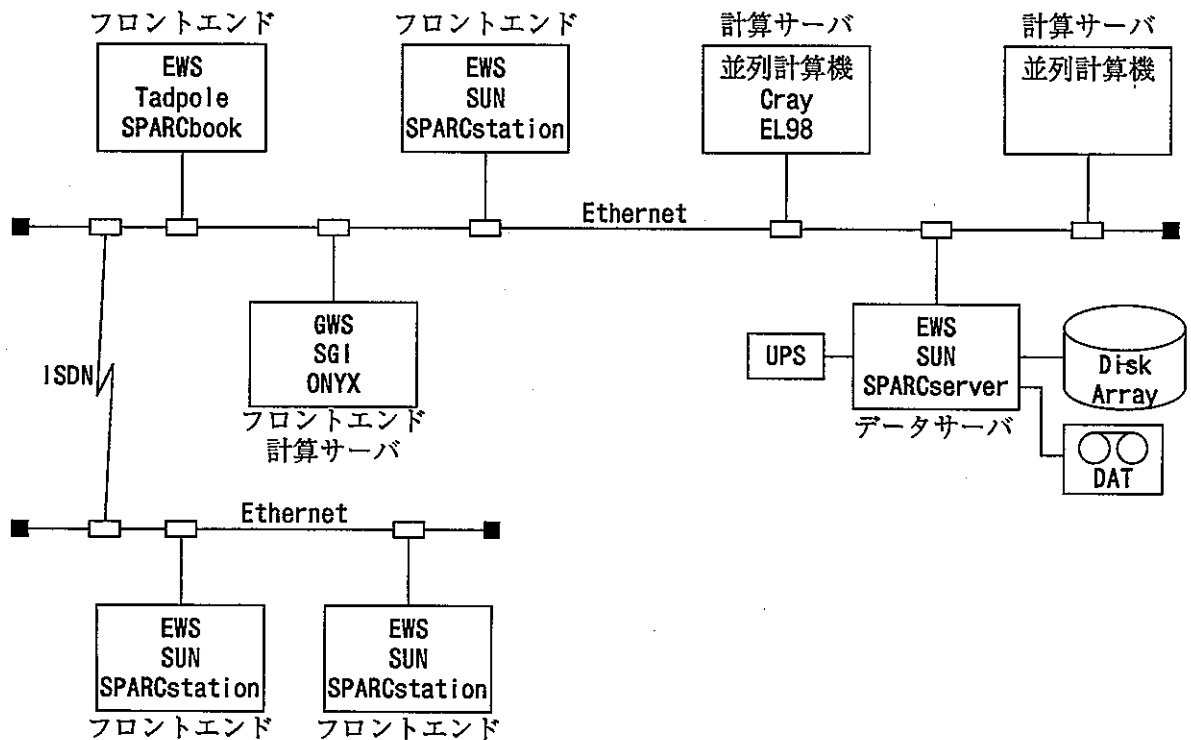


図3.2.3-1 PNC計算機環境下でのCAPASA



各計算機が受け持つ機能は以下の通りである。

### フロントエンド

計算サーバ、データサーバの制御を行う中心的な計算機である。ユーザは、PLANの作成や解析結果の参照等のすべての操作をこの計算機を通して行う。グラフィックの処理能力、及び、ユーザフレンドリなデスクトップ環境を持った計算機を用いる。SGI ONYXやSUN SPARCstationがこれにあたる。フロントエンドの台数を増やすことにより、同時にCAPASA/PCDEEを使用できる解析者の人数を増やすことができる。

### 計算サーバ

フロントエンドからの命令に従って、解析コードを実行する。並列計算機や高速なCPUを持った計算機を用いる。Cray EL98とSGI ONYXがこれにあたる。PLANの並列実行を行なう場合は、計算サーバの台数を増やすことにより、計算時間を短縮できる。

### データサーバ

フロントエンドからの命令に従って、解析コード実行のためのデータの準備や計算結果をデータベースに格納する。データサーバは、ディスクI/Oの優れた計算機を用い、ディスクアクセス時間を短縮するため、ディスクアレイシステム（500MBx10程度）を接続する。オブジェクト指向データベース管理システムVersantはこの計算機に実装される。SUN SPARCserverがこれにあたる。

図3.2.3-1に示したハードウェア構成を考えた場合、オペレーティングシステムはUNIX系、ウィンドウシステムはMotifであるため、スタンドアロンで構築したシステムからのGUI部分の移行は比較的容易であると思われる。また、Motifを採用しているワークステーションベンダーは多いため、機種選択の幅も広がる。しかし、データベース管理システムVERSANTに関しては、実行可能な計算機が限定されるため、現状ではフロントエンドとしてSUN以外の計算機を用いた場合、データアクセスのためにフロントエンド、データサーバ間で通信を行うアプリケーションを構築する必要がある。

図3.2.3-2に、この通信アプリケーションを用いたデータアクセス手順を示す。

フロントエンドからデータの要求が発生した場合、アプリケーションAを用いて、データサーバ上のアプリケーションBにデータ要求メッセージを送る。アプリケーションBでは、メッセージを受け取るとデータベース管理システムを通してデータを検索し、そのデータをアプリケーションAに送信する。この方法により、フロントエンドでVERSANTを実行する必要がなくなる。

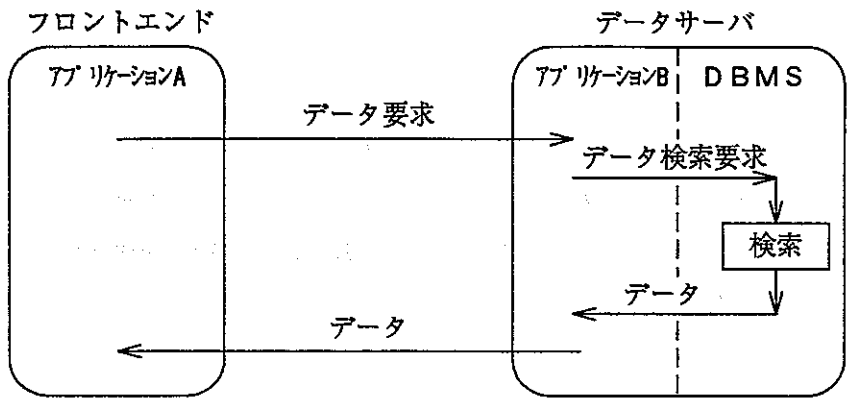


図3.2.3-2 PNC計算機環境でのデータアクセス手順

PLANの実行に関しては、PLAN実行を1台の計算サーバのみで行う場合には、問題はないが、3.2.2節で述べたPLANの並列実行を行うためには、ネットワーク上に接続された複数の計算サーバにPLANオブジェクトの実行を割り振る機能を持ったPLAN実行プログラムを構築する必要がある。

### 3. 2. 4 まとめ

本年度は、複数の現象解析コードを連成した大型の解析を効率的に実行するための方法として、CAPASAの分散処理環境への拡張に関して検討を行った。実際の実装段階では、さらに多くの検討、調査が必要ではあるが、現状では以下の点において分散処理環境が有効であることが分かった。

- ・グループワークを支援する環境として、クライアント/サーバモデルはネットワーク間でのデータ共有やコストパフォーマンスの観点から有効な手段である。
- ・CAPASAの機能を分類し、複数の異なった計算機にそれらの機能を割り当てることにより、解析を効率よく行うことができる。
- ・PLAN実行を複数の計算機上で並列に実行することは、PLAN実行時間を短縮する点では有効である。ただし、PLANオブジェクト実行のためのスケジューリングにはコスト算出方法等の多く課題を残している。

最後に、上記の分散処理環境のメリットを生かし、かつ、近い将来実現することが可能と思われるCAPASA/PCDEEの構成について示す。フロントエンド計算機としては、現状でVERSANTがサポートしている計算機であるSUN SPARCstationとPC-AT互換機(Compaq等)を利用する。ただし、PC-AT互換機を用いた場合は、SUN SPARCstationの一部の機能のみが提供されるものとする。計算サーバは、SGI ONYXを用い、PLANの並列実行は、コスト算出に多くの課題を残すため行わないものとする。ただし、ONYXは、高速なCPUを持っているため、SUN SPARCstation上で実行する場合よりも、高速にPLANを実行することができる。

#### 4. 連成解析システム上でのデータ等の管理機能向上のための環境整備

連成解析システムの基本システムでは、これまでに、オブジェクト指向データベースを導入し、解析結果、入力データ等の一部のデータの体系的管理を行ってきた。ここでは、基本システムに含まれる現象解析コードである ORIGEN2 と HYDROGEOCHEM を対象に、体系的管理の対象となるデータの範囲を拡張し、併せて出力の後処理及び入力、出力などのバージョン管理機能を拡張した。

##### 4. 1 機能仕様の検討

###### 4. 1. 1 連成解析システムでのデータの体系的管理方法について

ここでは、連成解析システムで扱う解析コードやデータを体系的に管理する方法について考察する。

まず、データベースに格納するデータを整理し、それぞれのデータベースでの格納に関する基本的考え方について検討する。次に、データのバージョン管理の考え方について検討する。なお、検討にあたっては基本システムの現在の仕様にとらわれることなく、できるだけ対象範囲を広げて検討した。

###### (1) データベースでのデータ格納の考え方

図4. 1. 1-1にCAPASAにおいてデータベースに格納されるべきデータとそれらの関係を図示す。これまでのシステム構成の議論からは、データベースで保持すべきデータは下記の6種類のデータが対象となると考えられる。

###### ① プロセス

CAPASAのプロセスライブラリに格納される解析コード、ユーティリティ・プログラムである。一般に、解析コードやユーティリティ・プログラムは、保存コストが少ないため、すべてのバージョンの保存を行う価値がある。もし、重要な解析結果を、CAPASAで再計算可能とすべきであるならば、解析プログラムのバージョン管理も含めた保存は不可避である。

プロセスの保存形態としては実行形式データをデータベース中に格納する方式が良い適切である。

なお、解析プログラムの変更点や改訂内容を確実にフォローできるようにするには、解析プログラムのドキュメンテーションや、さらにソース・プログラムを同時に管理する必要も生じよう。

###### ② 解析手順 (PLAN)

解析手順はPLANにより表現される。PLANはPLANライブラリに格納

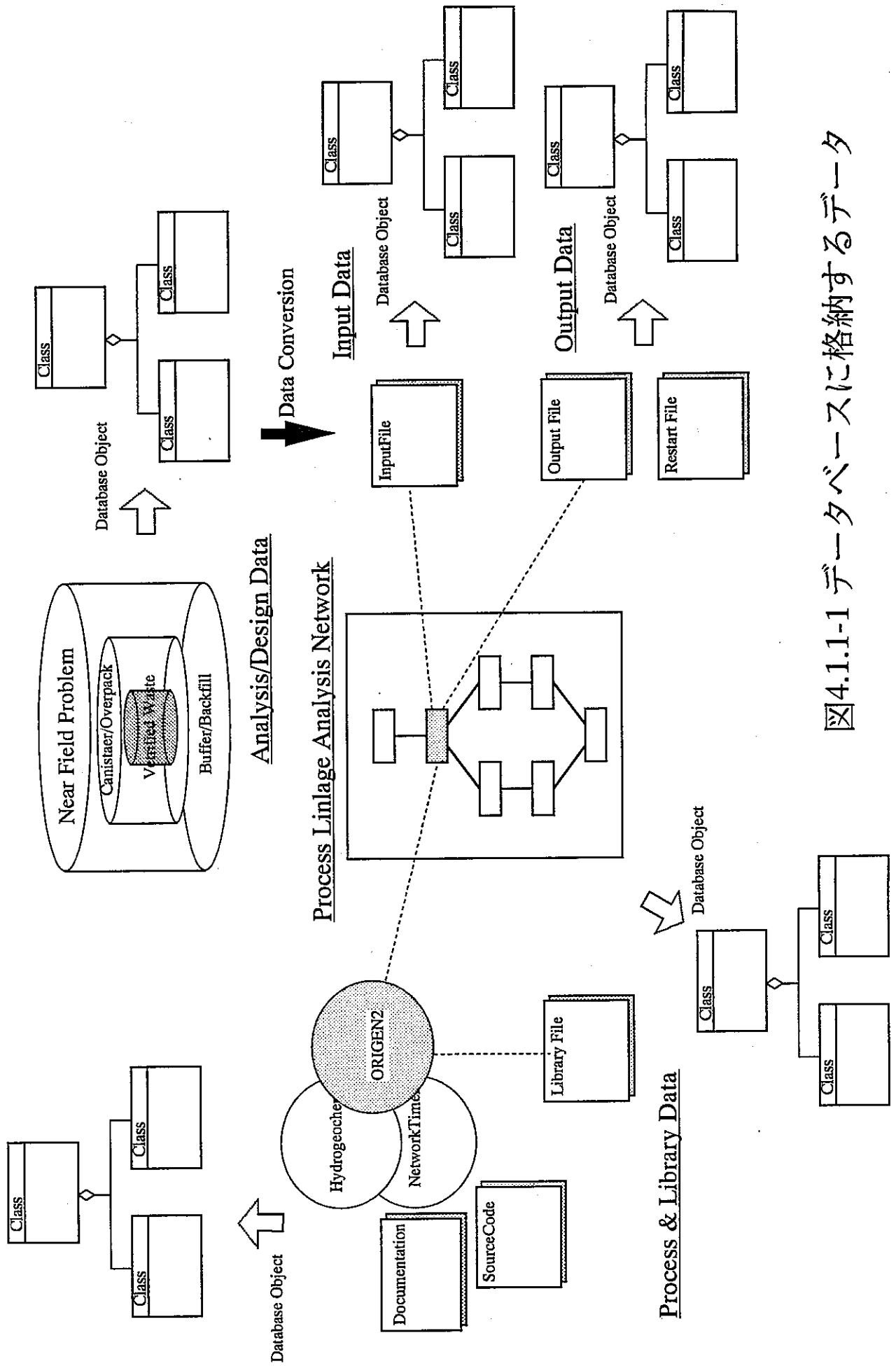


図4.1.1-1 データベースに格納するデータ

され、解析実行時のテンプレートとして使用されるものである。

現在の基本システムでは、解析手順は解析結果を保存するためのデータ構造としての役割を持っており、解析結果を保存するためには、解析手順も保存する仕掛となっている。

### ③入力データ（プロセスに対する直接の入力データ）

解析コードやユーティリティ・プログラムに固有の入力データは、結果の再計算のためにはかならず必要である。また、解析結果のみが必要な場合でも、解析条件を明確にするためにも、入力データの一部あるいは全部が必要となる。

入力データの保存方法としては、入力ファイル単位で保存する場合と、データベース中に構造化して保存する場合が考えられる。さらに、入力データファイルを作成する手順を保存する考えもある。これは、④の解析／設計情報データからの導出プロセスを保存するものである。この場合には、解析データの出典が明確になるというメリットがある。

### ④解析／設計情報データ（入力データ生成に用いられるデータ）

解析／設計情報データとは、ニアフィールド連成解析において各プロセスで共通に使用されるデータを意味しており、例えばガラス固化体、緩衝材、周辺岩盤などの形状、物性データ等である。基本システムでは、この種のデータの活用は図られていないが、多くの解析コードがシステムに取り込まれると、同一の解析対象を、解析プロセス毎にモデル化する必要が生じる。このような場合に、それぞれの入力データを整合性を持って作成しなければならないため、解析対象に関する単一のデータ・ソースから、自動的に各解析プログラム毎の入力データ作成のためのモデルを作成できることが望ましい。各解析プログラム毎で共通に使用可能とするためには、データベースにおけるデータ構造と、各解析プログラム用のモデル作成のためのアルゴリズムを検討し、実装しなければならない。

この種のデータは、地層処分システムの研究や設計の進展により更新されるはずのものである。このため、データベースとしても、これらの進展に対応できるような枠組みが望ましい。また、どの解析条件データが入力データの作成に用いられたかが重要となるため、入力データとの関連についてもデータベース中で保存できる枠組みが望ましい。

### ⑤ライブラリー・データ

解析コードに固有のライブラリー・データについては、解析プロセスとの関連をとりながら、データベースにて保持する。ライブラリーデータをファイル単位で持つか、データベースへ構造化して保存するかは、下記により判断すべきである。

- ・ライブラリーデータが特定の解析コードでの使用に限定されるのであればファイル単位で持つべきである。
- ・もし、データ変換により多くの解析プログラムにて共通に使用すべきものであれば、解析／設計情報データと同様に構造化して保持すべきである。

#### ⑥解析結果

解析により得られる結果は、通常、解析コードの出力ファイルやリスタートファイルとして作成される。このため、出力結果の保存方式は、ファイル単位で保存するか、あるいは構造化してデータベースに保存する方式が考えられる。

多くの場合、解析結果のうち必要な情報は出力ファイルの一部であり、出力ファイルが大きき場合には、ファイルすべてを保存することは賢明な方法ではない。

一方、解析結果の再計算が必要な場合には、リスタートファイルを活用することがあり、このような場合には、ファイル毎で結果を保存することが正当化されよう。

データを構造化して保存することの意義は、次のようにまとめられる。

- a. 解析結果のデータ形式が安定、不変であり、将来、どのような解析プログラムからでも、容易に活用可能となる。このため、ポスト処理機能とのデータ受渡しが容易になる。ファイル単位で持った場合には、必要なデータにアクセスするための手続きを記述するための労力が必要である（ファイル構造の変更、出力オプション等を考慮した手続きが必要となる）。
- b. 連成計算にて、データ受渡しを考慮する際に、データベースを介した受渡しが可能な場合に、ファイル構造に悩まされる事なく、データ変換手続きを作成できる。

これらによるメリットが大きく、かつファイル単位で保存することが、合理的でなければ、データベースで構造化して保存すべきである。

#### (2) 解析結果、入力データの保存レベル

解析結果ならびに入力データの保存には表4. 1-1に示すように、3つのレベルが考えられる。どのレベルでデータを保存するかは、データの活用形態による。

最も簡単なデータ保持方式は、UNIXファイルとして保存するもので、ファイル保存の信頼性は低いが、データ保存のための労力（システム開発に要する）は最も少ない。一方、ファイルのまま保存するため、データ量が大きくなる可能

## 表4.1.1-1 入力データ、解析結果の保存の考え方

レベル1	UNIXファイルとして保存	信頼性：× コスト：低
		データ重要度が低く、長期保存を想定しない場合に可能
レベル2	データベースへファイル単位で保存	信頼性：○ コスト：中
		ファイル単位でのデータの使用に意味がある場合（例えば再計算が重要な解析のリストファイルなど）
レベル3	データベースへ構造化して保存	信頼性：○ コスト：高
		CAPASA上で連成計算やデータ編集等により再利用されるデータ



性がある。例えば、再計算が容易である場合のようにデータの重要度が低く、かつ長期間保存する価値のない場合に適した形態である。

次のレベルは、データベースにファイル単位で格納する方式である。この場合には、UNIXファイルとして保存する場合と比べて、ファイルを長期間にわたり高い信頼度で保存可能である。データベースに格納するためには、そのためのデータベース設計を行わねばならず、また、データ量も大きくなるため、特にファイル単位での使用が有効であるもの（例えば、入力ファイルやリスタート・ファイル等）に限るべきであろう。

最もコストの高い方式は、データベースへ構造化して保存する方式である。この場合には、あらかじめ、データベースにクラス構造を定めておく必要がある。また、一度データを格納した後は、データ構造（スキーマ）の変更は、困難な場合もあり、あらかじめ十分な配慮が必要である。

### (3) バージョン管理について

バージョン管理は、対象とするオブジェクトの改訂の履歴が重要となる場合に必要となる。例えば、解析コードの機能拡張を考えてみると、プログラムを少しずつ改訂しながら、解析コードが進展するのが通例である。その途中段階にても解析を実行するとすれば、どの段階の解析コードを使用したかにより、解析アルゴリズムに違いが生じることが考えられる。また、解析コードが使用するライブラリ・データも同様であり、ライブラリの改訂があれば、どの時点でのライブラリを使用したかは、解析結果の解釈に大きな意味を持つてくる。

このため、本来その内容が変わりにくいオブジェクトではあるが、そのわずかな変更が大きな意味を持つ場合にバージョン管理の必要性が生じることになる。

そのような観点からバージョン管理が大きな意味を持つのは次のデータである。

- ① プロセス（解析コード、ユーティリティ・プログラム）
- ② ライブラリー・データ
- ③ 解析／設計データ

一方、解析を実行する際には、パラメータを系統的に振った解析が行われるが、入力データの部分的な変更を行いながら、解析を進めることになる。解析を実施する立場からは、入力データの素性を明らかにする上で、そのバージョン管理を行う必要が生じよう。しかしながら、一般に入力データのバージョン管理へのニーズは短期的要求であることが多く、データベースでの管理が必要なものは系統的な解析を実施した結果の一部となろう。

CAPASA/PCDEEでは、入力データ、解析コード、解析結果の関連を

示すネットワーク構造を解析 (Analysis)として管理する。このため入力データ毎のバージョン管理を行うのではなく、解析毎にバージョン管理を実施することで十分と考えられる。

表4. 1. 1-2にCAPASA/PCDEEにおけるデータとバージョン管理の考え方を示す。

なお、今年度製作したORIGEN2/Hydrogeochem Workbenchでの扱いは表中の表現と対応している。

表4. 1. 1-2 CAPASA/PCDEEにおけるバージョン管理の考え方

管理対象とするデータ	バージョン管理の階層
解析 (PLAN) 入力データ、解析結果、手順	3 階層 [解析名称]-[Major Version]-[Minor Version]
プロセス (解析コード)	2 階層 [コード名称]-[Version]
ライブラリーデータ	1 階層 [ライブラリー名称]
解析/設計データ	未定

#### 4. 1. 2 ユーザ支援環境

本節では、核燃料の燃焼や中性子照射による放射性物質の蓄積・崩壊等に関する計算を行うORIGEN2、及び、地下水中成分の輸送及び化学反応の計算を行うHydrogeochemを用いた解析を効率よく行うことができる環境を提供するORIGEN2/Hydrogeochem Workbench（以下、Workbench）について述べる。

通常、解析業務は入力データの作成、解析コードの実行、解析結果の参照といった作業を繰り返し行う。この際、入出力データ・解析コードの管理、データの表示・編集等を効率よく行うことが解析業務を円滑に行う上で重要なポイントとなる。このため、本システムは以下の特徴を持つ。

- ・ 入出力データや解析条件等を表わした情報をデータベースに集中管理することによる再利用性や品質保証の向上
- ・ グラフィカルユーザインタフェースを用いた操作性の優れた環境の提供
- ・ データ編集機能による外部ツールとの連携

Workbenchは図4.1.2-1に示すようにORIGEN2とHydrogeochemの2種類の解析コードと以下の機能を持つ複数のToolから構成されている。

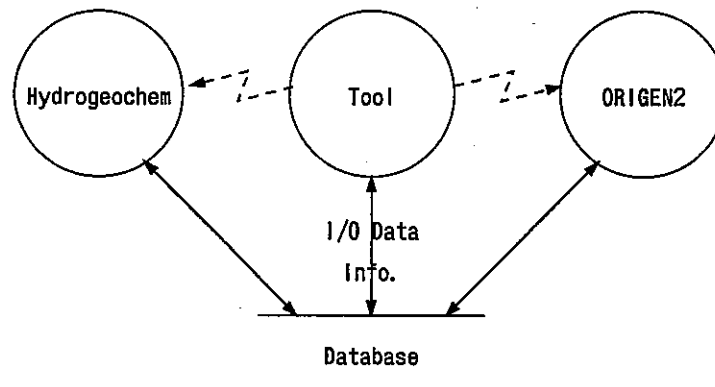


図4.1.2-1 Workbench 概略図

- ・ データ管理機能      データ全体の管理を行う。
- ・ 全体制御機能      解析コードの実行管理やシステム全体の管理を行う。
- ・ 入力データ作成機能      解析コードの入力データや解析条件等を表わした補助情報を作成しデータベースに保存する。
- ・ 入出力データ保存機能      解析コードの入出力データをクラス構造に変換しデータベースに保存する。
- ・ データ参照機能      解析コードの入出力データをデータベースから検索し、表示する。
- ・ データ編集機能      解析コードの出力データをデータベースから検索し、特定のフォーマットでファイルに出力する。

### (1)データ管理機能

データ管理機能は解析コード、解析コードの入出力データ、解析の条件を表わした補助情報を一括管理し、データベースに保存する。これらのデータは解析名称をキーデータ（データを一意に定めることができるデータ）として、各種データをこれに関連付けることにより構成されている。

図4.1.2-2はデータ概念図を示しており、1つの解析のデータセットは中心となるAnalysisクラス、

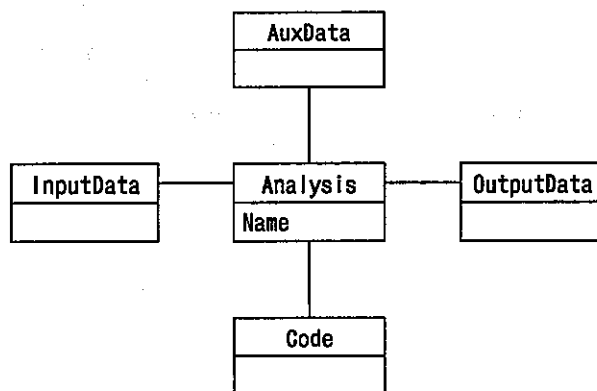


図4.1.2-2 データ概念図

補助情報を持ったAuxDataクラス、入力データを表わしたInputDataクラス、出力データを表わしたOutputDataクラス、解析コードの情報を持ったCodeクラスから構成されている。

また、これらのデータセットはバージョン付けがなされており、データサーベイや管理を容易に行うことができる。例えば、既存のAnalysisクラスのインスタンスを利用して解析を行うと、最初にAnalysisクラス、InputDataクラス、AuxDataクラスのインスタンスが複製される。ただし、その際にAnalysisクラスのバージョン属性のみが変更（バージョンアップ）される。次に必要であれば、InputDataクラス、AuxDataクラスのインスタンスの編集を行う。そして、解析が実行されるとOutputDataクラスがバージョンアップされたAnalysisクラスのインスタンスに関連付けられ、データセット全体がバージョンアップされる。（図4.1.2-3参照）

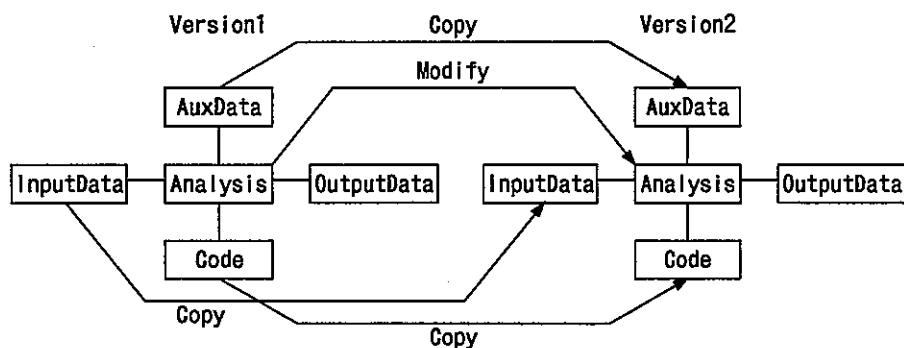


図4.1.2-3 データのバージョン管理

データベースへの保存は図4.1.2-4に示すように2段階で管理される。解析対象となっているデータは最初は個々の計算機毎に用意された一時的なデータベース（パーソナルデータベース）に格納され、ユーザは解析の実行、データ表示等の処理をこのデータベースに対して行う。そして、ユーザがこのデータを長期に渡って保存したい場合にはサーバーとなっている計算機上に用意されている永続的なデータベース（グループデータベース）に複製する。

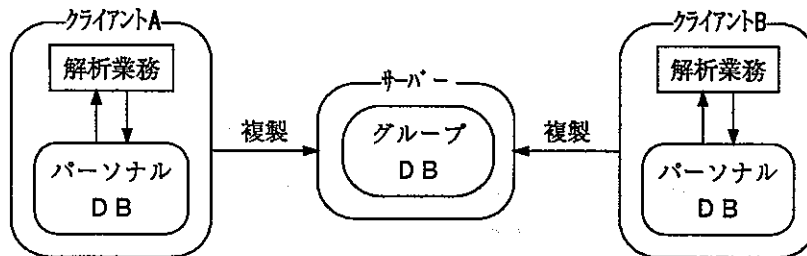


図4.1.2-4 データベース構成

この機能により、データ管理や解析実行による計算機の負荷を分散することができ、複数の解析者が同時に効率よく解析業務を行うことが可能となる。

## (2) 全体制御機能

解析業務を行うには、解析名称の設定、補助情報の入力、入力データの作成、解析コードの実行、出力データの保存、データの参照を行う。全体制御機能は図4.1.2-5に示すようにこれらの作業内容に適したアプリケーションの起動や複数のアプリケーションに共通なデータの設定、データベースの管理を行う。

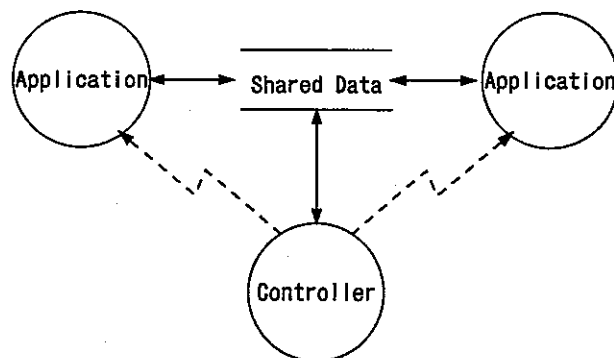


図4.1.2-5 全体制御機能

### (3)入力データ作成機能

解析コードの入力データや解析条件等を表わした補助情報を作成し、データベースに保存する。入力データは解析コードが持つ特定の形式（解析コードの入力ファイルと同等の形式）とその主要な部分をクラス構造に変換した形式でデータベースに保存される。

補助情報は以下に示すような解析コード間で共通な日付等の情報と解析コード固有の情報を持つ。

- ・ 共通情報
 

作成日	作成年月日
作成者	作成者
コメント	注釈
- ・ ORIGEN2用情報
 

イベント	Charge、Discharge、Cooling BR、Reprocess、Cooling BV、Vetrified、Storageの各イベントの期間
------	---

ORIGEN2の入力データには、他の機能との連携や入出力データのクラス構造への変換を確実にを行うため、以下の制約を与えるものとする。（Hydrogechemに関しては制約はなし）

#### ORIGEN2の制約

- ・ 1回の実行で出力できるテーブルは表4.1.2-1中から1種類のみとする。

表4.1.2-1 出力テーブル種別

Table No.	Description	Unit
1	Isotopic Composition of each element	atom fraction
2	Isotopic Composition of each element	weight fraction
3	Composition	gram-atoms
4	Composition	atom fraction
5	Composition	grams
6	Composition	weight fraction
7	Radioactivity(total)	Gi
8	Radioactivity(total)	fraction
9	Thermal Power	watts
10	Thermal Power	fraction
13	Radioactive Inhalation hazard	m3 air
14	Radioactive Inhalation hazard	fraction
15	Radioactive Ingestion hazard	m3 water
16	Radioactive Ingestion hazard	fraction
17	Chemical ingestion hazard	m3 water
18	Chemical ingestion hazard	fraction
19	Neutron absorption rate	neutrons/sec
20	Neutron absorption rate	fraction
21	Neutron induced fission rate	fissions/sec
22	Neutron induced fission rate	fraction
23	Radioactivity(alpha)	Gi
24	Radioactivity(alpha)	fraction

- ・出力テーブルのタイプは核種の出力を含むものに限定する。(表4.1.2-2の1、2、3、5のみ)

表4.1.2-2 出力テーブルタイプ

NOPTOL(1) NOPTA(1) NOPTF(1)	Table type printed		
	Nuclide	Summary	Element
1	Yes	Yes	Yes
2	Yes	No	Yes
3	Yes	Yes	No
4	No	Yes	Yes
5	Yes	No	No
6	No	No	Yes
7	No	Yes	No
8	No	No	No

- ・時系列データの前に出力されるデータのタイトルは「BEFORE TIME-SERIES DATA (固化体のタイプ名)」とする。
- ・時系列データのタイトルは「TIME-SERIES DATA(\*Y-\*Y)」とする。

入力データの主要な部分は、データ表示の際の検索を効率よく行うためにクラス構造に変換される。変換対象となるデータは以下の通りである。

#### ORIGEN2の対象データ

- ・出力テーブル種別

#### Hydrogeochemの対象データ

- ・ノードの位置情報
- ・コンポーネント、化学種の名称

#### (4)入出力データ保存機能

解析コードの出力ファイルをクラス構造に変換し、データベースに保存する。クラス構造の変換は、データベースの容量を考慮して出力ファイルの全データを変換するのではなく、主要なデータのみを変換しデータベースに保存する。変換対象となるデータは以下の通りである。

#### ORIGEN2の対象データ

- ・イベント終了時点毎の核種のデータ
- ・核種の時系列データ

#### Hydrogeochemの対象データ

- ・コンポーネント毎の全ノードでの時系列の濃度データ
- ・化学種毎の指定ノード(データセット5で設定されたノード)での時系列の濃度データ

#### (5)データ参照機能

解析コードの入出力データや補助情報をデータベースから検索し、テキスト形式、表形式、グラ



フ形式のいずれかで表示する。解析コード毎の表示形式は以下の通りである。

#### ORIGEN2の表示形式

- ・ イベント終了時点での核種のデータを表形式で表示する。
- ・ 核種、または、元素の時系列データを横軸を時間、縦軸をパラメータとしてグラフ表示する。
- ・ 入出力データ、Photonライブラリ、Decayライブラリをテキスト形式で表示する。
- ・ 補助情報を表形式で表示する。

#### Hydrogeochemの表示形式

- ・ 指定ノードのコンポーネント、または、化学種の濃度を横軸を時間、縦軸を濃度としてグラフ表示する。ただし、化学種の濃度はデータセット5で指定されたノードのみとする。
- ・ 入出力データをテキスト形式で表示する。
- ・ 補助情報を表形式で表示する。

#### (6)データ編集機能

特定の解析内容に対応するクラス構造に変換して保存された出力データをデータベースから検索し、以下に示した特定のフォーマットでファイルに出力する。

#### ORIGEN2のフォーマット

- ・ データベースに保存されている出力データを外部アプリケーション（PNC所有）で利用できるファイル形式で出力する。フォーマットは以下の通りである。

固化体の名称

テーブル番号

"ACTINIDES+DAUGHTERS"

"EVENT TABLE"

イベント リスト

イベント 期間

核種データ

"NUCLIDE TABLE"

核種時系列データ

"FISION PRODUCTS"

"EVENT TABLE"

イベント リスト

イベント 期間

核種データ

"NUCLIDE TABLE"

核種時系列データ

### Hydrogeochemのフォーマット

- ・コンポーネントまたは化学種の指定ノードでの濃度の時系列データを以下のフォーマットで出力する。

”コンポーネント名称”, ”タイプ”、または、” 化学種のタイプ”、” 化学種名称”

ノード番号

時間、値

時間、値

...

例) コンポーネント名称「SODIUM」、タイプ「Dissol」、ノード番号 1が設定された時

”SODIUM”, ”Dissol”

1

0.0000E+00, 1.0000E-01

5.0000E-01, 1.0000E-01

1.0000E+00, 1.0000E-01

1.5000E+00, 1.0000E-01

## 4. 2 アプリケーションの設計・製作

### 4. 2. 1 データベース仕様

4.1節の機能仕様の検討に基づき、オブジェクト指向データベース管理システムVERSANTのデータベーススキーマを設計、製作した。図4.2.1-1から図4.2.1-4にスキーマのオブジェクトダイアグラムを示す。また、図4.2.1-5から図4.2.1-6にORIGEN2、Hydrogeochem関連のインスタンス図を示す。

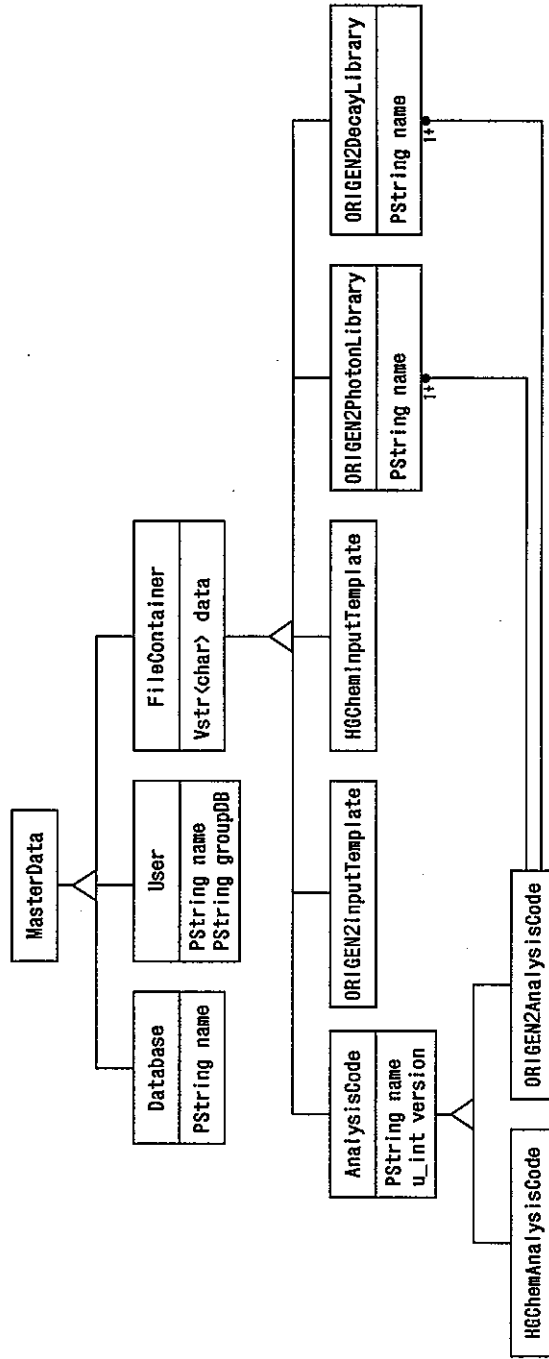


図 4.2.1-1 データベーススキーマ マスターデータ

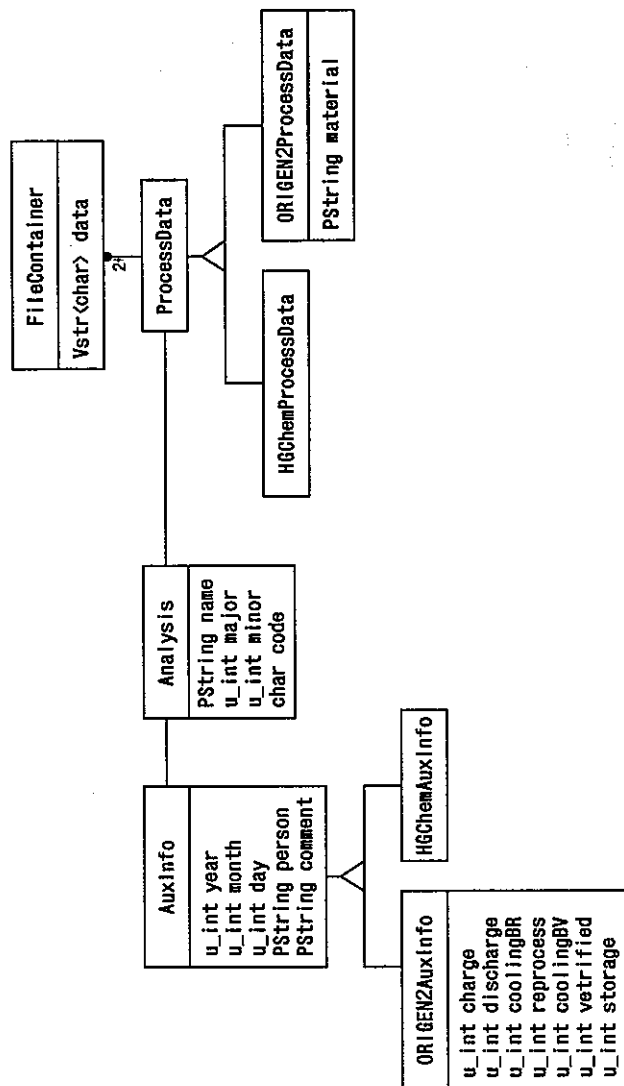


図4.2.1-2 データベーススキーマ 解析関連

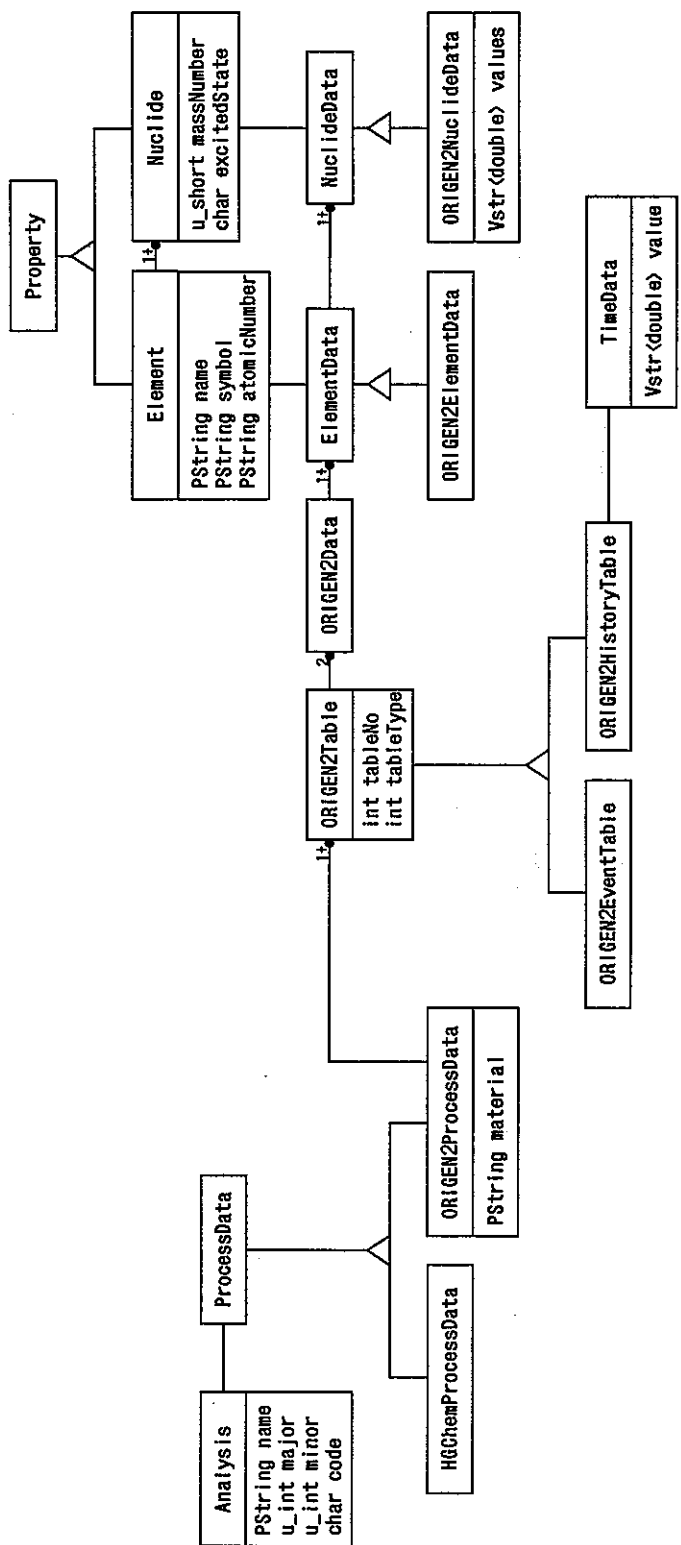


図4.2.1-3 データベーススキーマ ORIGIN2関連

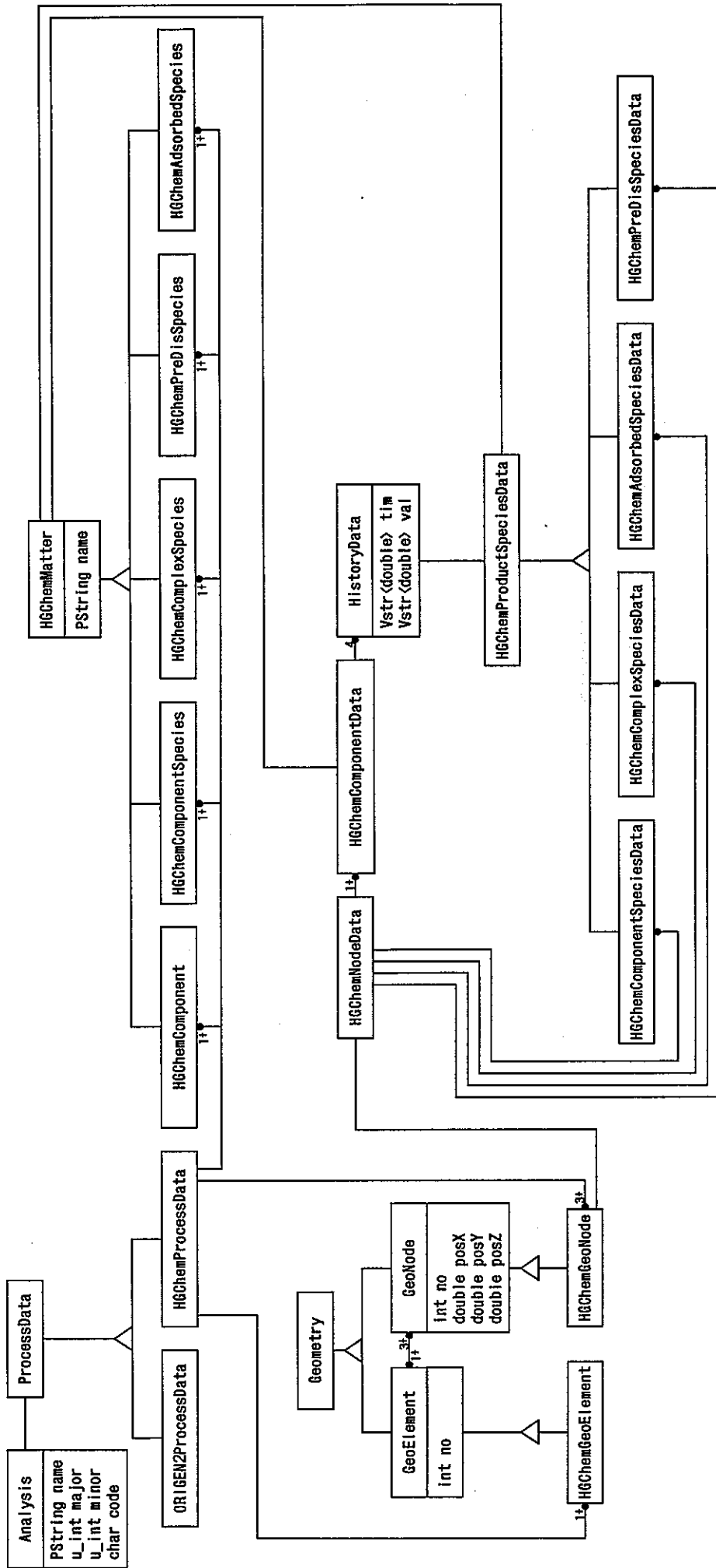


図4.2.1-4 データベーススキーマ Hydrogeochem関連

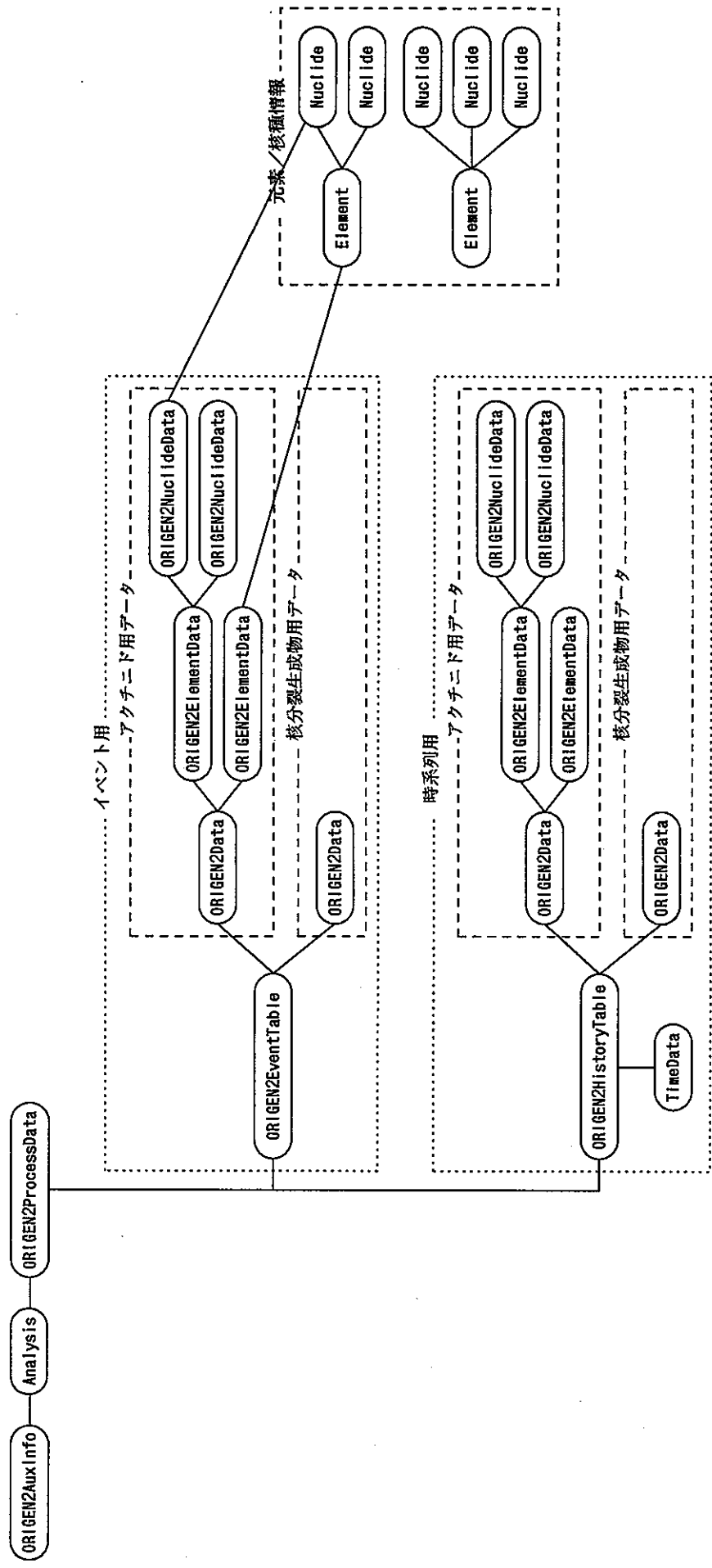


図4.2.1-5 インスタンス図 ORIGEN2関連



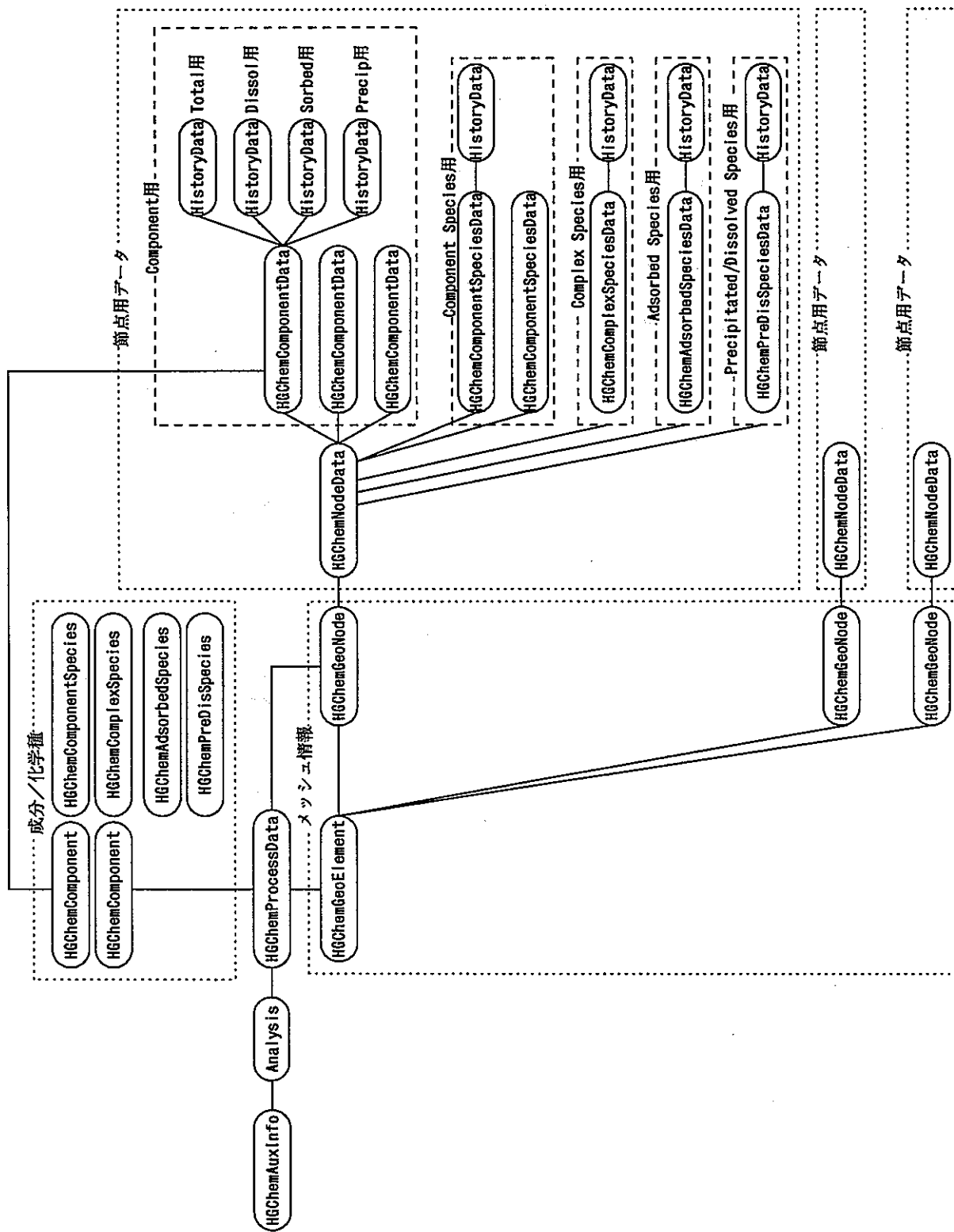


図4.2.1-6 インスタンス図 Hydrogeochem 関連

#### 4. 2. 2 ソフトウェア仕様

本システムは図4.2.2-1に示すように7種類のモジュール、2種類の解析コード、及び、5種類のデータストアから構成されている。

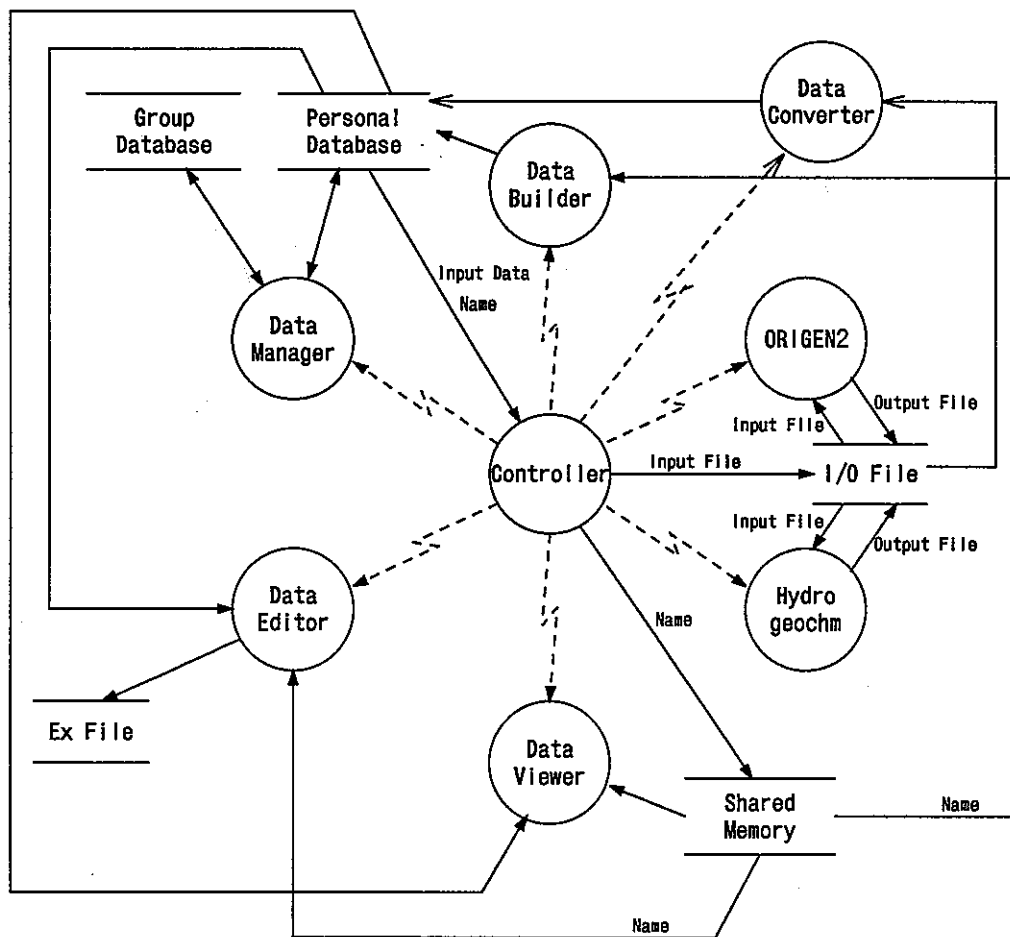


図4.2.2-1 モジュール全体構成

#### モジュール

- ・全体制御モジュール (Controller)  
各モジュールの制御やモジュール間で共通で用いられるデータの管理を行う。
- ・入力データ作成モジュール (DataBuilder)  
補助情報の作成や解析コードの入力データを作成する。
- ・データ変換モジュール (DataConverter)  
解析コードの入出力データをクラス構造に変換し、データベースに保存する。
- ・データ表示モジュール (DataViewer)  
解析コードの入出力データを表形式、グラフ形式、テキスト形式で表示する。
- ・データ編集モジュール (DataEditor)

データベースに保存された出力データを特定のフォーマットでファイルに出力する。

- ・データ管理モジュール (DataManager)

パーソナルデータベースとグループデータベースの管理やデータベース内のデータの削除を行う。

### 解析コード

- ・ORIGEN2

核燃料の燃焼や中性子照射による放射性物質の蓄積、崩壊を計算する。

- ・Hydrogeochem

地下水中成分の輸送と化学反応を計算する。

### データベース

- ・パーソナルデータベース (Personal Database)

一時的なワーク領域用のデータベース。解析対象となっているデータのみが保存される。

- ・グループデータベース (Group Database)

永続的なデータベース。パーソナルデータベースのデータが複製されることにより保存される。

- ・I/Oファイル (I/O File)

解析データの入出力ファイルが一時的に格納される。

- ・外部ファイル (Ex File)

外部アプリケーションが用いるファイルを格納する。

- ・共有メモリ (Shared Memory)

解析名称等のモジュール間の共通な情報が格納される。

本システムが推奨するハードウェア構成、及び、ソフトウェアは以下の通りである。

### ハードウェア

- ・サンマイクロシステム SPARCstation20

主メモリ 32MB

ハードディスク 1GB

カラーモニタ

### ソフトウェア

- ・オペレーティングシステム 日本語 Solaris 2.3

- ・ウィンドウシステム 日本語 Motif 1.2.2

- ・オブジェクト指向データベース管理システム VERSANT

#### 4. 2. 3 動作確認

仕様に基づきアプリケーションを製作し、以下に示した解析の例に従い、問題なく動作することを確認した。

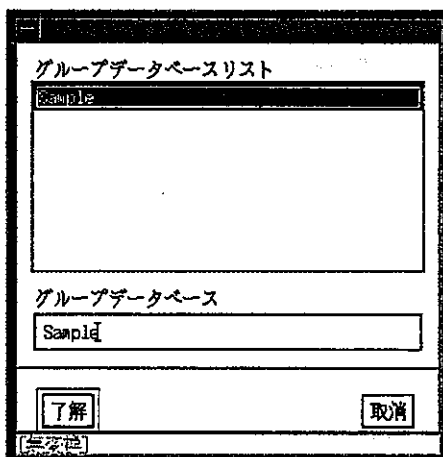
全体制御モジュール起動

全体制御モジュールを起動することができる。



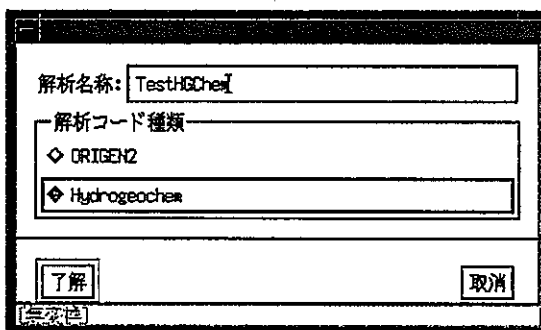
グループデータベース選択

グループデータベースの一欄が表示され、接続するグループデータベースを選択することができる。



Hydrogeochem新規解析作成

Hydrogeochemの新規解析を設定することができる。



続き

入力データ作成

Hydrogeochem用の入力データを作成することができる。

```
モジュール データ
3 TEST OF HYDROGEOCHEM: EQUILIBRIUM MODULE - ADSORPTION 000
C ***** DATA SET 2: BASIC INTEGER PARAMETERS
6 2 1 0 3 1 8 -1 1 0 1 0 50 1 150 0 0 0 NNP NEL NMT NCM NTI KSS NMPM KVI
C ***** DATA SET 3: BASIC REAL PARAMETERS
0.5000 0.000 0.50000 15.000 1.000 1.000 1.000 1.000 1.0000 1.0E-5 1.000
C ***** DATA SET 4: PRINT AND AUXILIARY STORAGE CONTROL
5555
0000
1.0D38
C ***** DATA SET 5: CHEMICAL OUTPUT AND CHEMICAL PROPERTY TYPE INDICATOR
3
1 3 6 NOSEP
C ***** DATA SET 6: MATERIAL PROPERTIES
2.0D-3 0.0 0.0 1.2 0.0 1.0D-6 0.2D-6 1.37D5
C ***** DATA SET 7: NODE COORDINATES
1 2 2 0.0 0.0 0.0
2 2 2 0.025D0 0.000 0.000
0 0 0 0.0 0.0 0.0 END OF X-COORD
1 2 2 0.000 0.025D0 0.000
2 2 2 0.000 0.025D0 0.0
0 0 0 0.000 0.000 0.000 END OF Z-COORD
C ***** DATA SET 8: ELEMENT CONNECTIVITY
1 1 2 4 3 1 1 2 IE
C ***** DATA SET 10: CHEMICAL COMPONENT INFORMATION
7
SODIUM 1 3
```

補助情報作成

補助情報を作成することができる。

解析コード情報

名称	バージョン
gmic	ver.1

補助情報

作成日: 1995 年 5 月 19 日

作成者: ifusaeda

コメント: これはサンプル用のデータです。I

了解

[ひら]

続き

↓  
解析実行

Hydrogeochemを実行することができる。

モジュール	
経過時間:	00:00.25
状況:	終了

↓  
補助情報参照

入力データ作成モジュールで作成された補助情報を参照することができる。

モジュール	
解析コード	SRIIC Version.1
作成日	1995年 5月 19日
作成者	fusaeda
コメント	これはサンプル用のデータです。

↓  
入力データ参照

入力データ作成モジュールで作成された入力データを参照することができる。

モジュール	
3 TEST OF HYDROGEOCHEM: EQUILIBRIUM MIXTURE - ADSORPTION 000 5	
C ***** DATA SET 2: BASIC INTEGER PARAMETERS	
6 2 1 0 3 1 8 -1 1 0 1 0 50 1 150 0 0 0 NIP NEL NMIT NOM NFI NSS NIPPM KVI	
C ***** DATA SET 3: BASIC REAL PARAMETERS	
0.5000 0.000 0.50000 15.000 1.000 1.000 1.000 1.000 1.00-5 1.000	
C ***** DATA SET 4: PRINT AND AUXILIARY STORAGE CONTROL	
SESS	
0000	
1.0000	
3	
C ***** DATA SET 5: CHEMICAL OUTPUT AND CHEMICAL PROPERTY TYPE INDICATOR	
1 3 6	
C ***** DATA SET 6: MATERIAL PROPERTIES	
2.00-3 0.0 0.0 1.2 0.0 1.00-6 0.20-6 1.3705	
C ***** DATA SET 7: MISC COORDINATES	
1 2 2 0.0 0.0 0.0	
2 2 2 0.02500 0.000 0.000	
0 0 0 0.0 0.0 0.0	
1 2 2 0.000 0.02500 0.000	
2 2 2 0.000 0.02500 0.000	
0 0 0 0.000 0.000 0.000	
C ***** DATA SET 8: ELEMENT CONNECTIVITY	
1 1 2 4 3 1 1 2	
C ***** DATA SET 10: CHEMICAL COMPONENT INFORMATION	
7	
SODIUM	1 3
LEAD	2 3
HYDROGEN	3 3
CHLORIDE	4 3
SIL	5 2

↓  
入力データ参照

グループデータベースへチェックインすることができる。

モジュール	
解析名称:	TestHGChem
バージョン:	11 . 11
[無変換]	

ORIGEN2読み込み

既存のORIGEN2の解析をグループデータベースからチェックアウトすることができる。

解析コード種類

- ◆ ORIGEN2
  - ◆ Hydrogeochem

既存解析

名称	バージョン
TestORIGEN2	Ver. 1 . 1

解析

名称: TestORIGEN2

バージョン: 1 . 1

了解 取消

イベントデータ参照

イベントのデータを参照することができる。

モジュール 表示

アクチニド+線核種

イベント 期間	Change	Discharge	Cooling BR	Reprocess	Cooling BV	Vitrified	Storage
	0 Year	0 Year	3 Year	0 Year	1 Year	0 Year	30 Year
He 4	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
Tl206	0.000e+00	1.564e-08	1.564e-08	1.564e-08	1.564e-08	1.564e-08	1.564e-08
Tl207	0.000e+00	3.124e+03	1.137e+04	1.137e+04	1.453e+04	1.453e+04	8.403e+04
Tl208	0.000e+00	1.636e+07	1.537e+08	1.537e+08	1.075e+08	1.075e+08	1.406e+06
Tl208	0.000e+00	1.896e+02	4.601e+01	4.601e+01	4.607e+01	4.607e+01	1.140e+02
Pb206	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
Pb207	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
Pb208	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
Pb209	0.000e+00	8.783e+03	2.130e+03	2.130e+03	2.133e+03	2.133e+03	5.276e+03
Pb210	0.000e+00	4.144e+01	4.309e+01	4.309e+01	4.271e+01	4.271e+01	1.970e+02
Pb211	0.000e+00	3.132e+03	1.140e+04	1.140e+04	1.457e+04	1.457e+04	8.427e+04
Pb212	0.000e+00	4.720e+07	4.278e+08	4.278e+08	2.992e+08	2.992e+08	3.913e+06
Pb214	0.000e+00	2.357e+00	2.417e+01	2.417e+01	3.824e+01	3.824e+01	5.069e+02
Bi208	0.000e+00	2.385e-08	2.385e-08	2.385e-08	2.385e-08	2.385e-08	2.385e-08
Bi209	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
Bi210M	0.000e+00	1.570e-08	1.570e-08	1.570e-08	1.570e-08	1.570e-08	1.570e-08
Bi210	0.000e+00	4.184e+01	4.312e+01	4.312e+01	4.274e+01	4.274e+01	1.970e+02
Bi211	0.000e+00	3.132e+03	1.140e+04	1.140e+04	1.457e+04	1.457e+04	8.427e+04
Bi212	0.000e+00	4.720e+07	4.278e+08	4.278e+08	2.992e+08	2.992e+08	3.913e+06
Bi213	0.000e+00	8.775e+03	2.130e+03	2.130e+03	2.133e+03	2.133e+03	5.276e+03
Bi214	0.000e+00	2.357e+00	2.417e+01	2.417e+01	3.824e+01	3.824e+01	5.069e+02
Po210	0.000e+00	2.185e+01	4.289e+01	4.289e+01	4.232e+01	4.232e+01	1.970e+02
Po211M	0.000e+00	4.914e-06	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
Po211	0.000e+00	8.771e+00	3.191e+01	3.191e+01	4.081e+01	4.081e+01	2.350e+02

続き

↓  
グラフ表示

指定された核種の時系列データをグラフ表示する。

モジュール グラフ データ

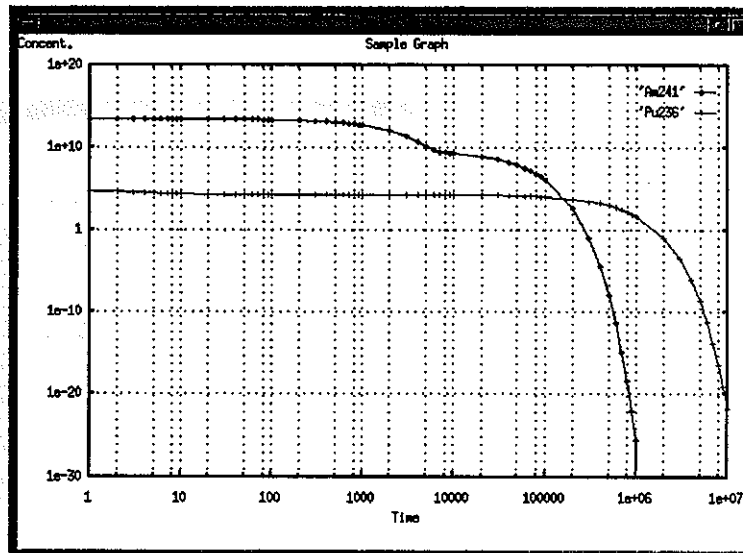
元素	核種	グラフ
Am	Total	Am241
Cm	237	Pu238
Bk	238	
Cf	239	

タイトル: Sample Graph

X軸  
最大: [ ] 最小: [ ] ステップ: [ ]  
ラベル: Time

Y軸  
最大: [ ] 最小: [ ] ステップ: [ ]  
ラベル: Concent

[ 変換 ]



↓  
データ編集

ORIGEN2の出力結果を指定フォーマットでファイル出力できる。

モジュール

出力タイプ

- ◇ 1行
- ◇ 複数行



#### 4. 2. 4 拡張性について

WorkBenchでのデータベーススキーマ構築に当たっては、今後の新規解析コードの取り込みの際の作業量を軽減するため、以下の点を考慮した。

##### ・抽象クラスの定義

解析コードに依存したクラスのみを定義するのではなく、最初に抽象クラスを作成し、その抽象クラスから、解析コードに適応したクラスを導出する方法を採用した。これにより、新規解析コード取り込みの際は、同様に抽象クラスから新規クラスを導出することにより、比較的容易にクラスを作成することができる。(例 図4.2.4-1を参照)

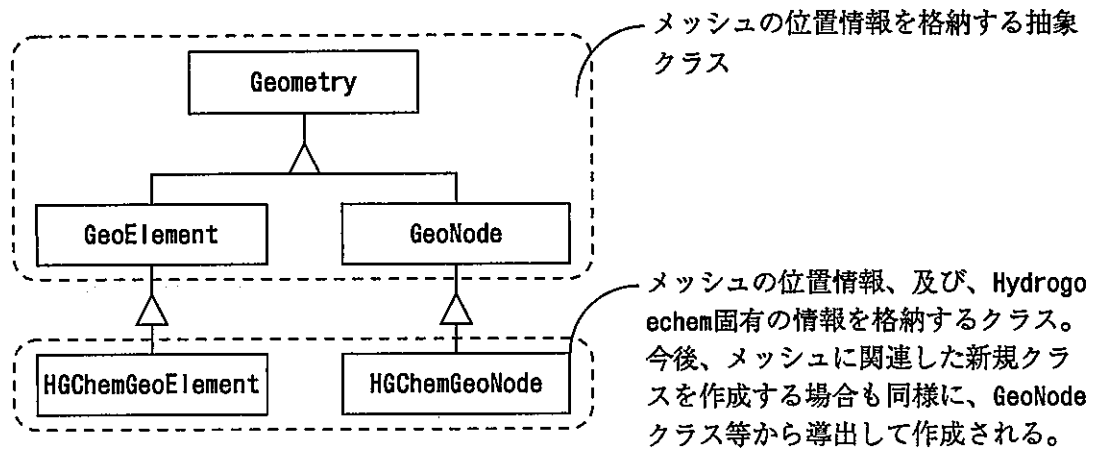


図4.2.4-1 抽象クラスの適用例

##### ・物性値の保存

元素、核種の情報を解析コードに依存することなく保存することにより、今後の再利用性を向上する。(Propertyクラス、Elementクラス、Nuclideクラスに情報を保存)

また、ファイルとデータベースのクラス間のデータ変換を行なうデータ変換モジュールでは、Hydrogeochemの入力ファイルに関しては、図4.2.4-2に示すように入力データ全体をデータセット毎に用意された計算機のメモリ内のクラスに格納し、この中から必要な情報をデータベースのクラスに割り付ける方法とした。これにより、データベースへの保存項目の追加は、ファイルに直接アクセスするのではなく、メモリ内のクラスにアクセスすることでできるため、容易に行なうことができる。

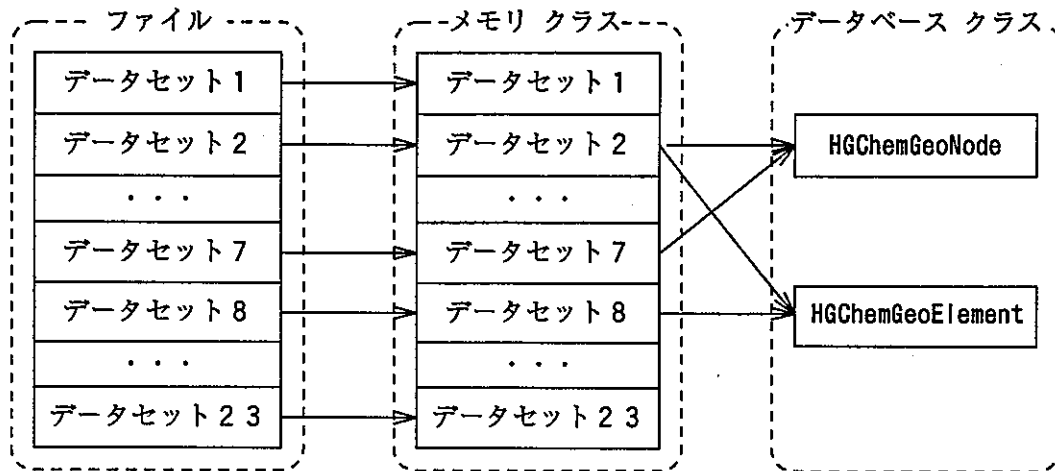


図4.2.4-2 Hydrogeochemにおけるデータ変換方法

本年度の製作したWorkBenchは、解析コードとしてORIGEN2とHydrogeochemを対象したが、データベーススキーマやデータ変換方法を汎用的なものとするにより、今後の拡張や新規解析コードの導入を意識したものとなっている。

## 5. あとがき

本研究では、ニアフィールドの複合現象を柔軟に解析できる連成解析用計算支援システム（CAPASA）の拡張と高度化を目的として、下記の項目を実施した。

- ①現象解析コード連成における作業内容・基本的手順の知識化
- ②連成解析システムの基本フレーム機能の拡充に関する検討
- ③連成解析システム上でのデータ等の管理機能向上のための環境整備

現象解析コード連成における作業内容・基本的手順の知識化では、現象解析コードを連成解析システム上で連成させるための基本的手順を、これまでの研究で得られた知見を基に、システム分析から連成実現にいたるステップ毎の手順としてまとめた。今後の連成解析システムの運用を考えた場合には、連成実現のための手順をより具体的に定めていくことが重要であり、今後とも、連成実現の経験を積み重ねながら、知識の蓄積と手法の整備を進めて行く必要がある。

連成解析システムの基本フレーム機能の充実に関する検討では、下記の項目について検討した。

- ・PLAN構築、PLAN実行モジュールの高機能化に関する検討・設計
- ・分散処理に関する検討

PLAN構築、実行モジュールの高機能化に関する検討では、今後の連成解析領域の拡張における作業効率と解析業務の効率向上のために、PLAN構築モジュールとPLAN実行モジュールの高機能化をプロトタイピングを含めて検討した。この結果、PLANにおけるデータフローならびに制御フローの表現について見直し、PLAN表現能力を拡張した高度化仕様を定めることができた。

また、分散処理に関する検討では、複数の現象解析コードを連成した大型の解析を効率的に行うために、分散処理手法の導入について検討した。分散処理環境においては、PLANオブジェクト単位での並列処理により、解析の効率化が期待できるが、そのためにはオブジェクトの実行を制御するためのコントローラの実現が課題となる。

高度化の検討結果をシステムに反映するのは、次年度以降の課題であるが、PLAN構築、実行モジュールについては、本年度に行った設計に従い、システムの実装が可能な段階となった。また、分散処理についても、現行のPNC計算機環境を対象とした具体的な検討を開始すべきと思われる。

連成解析システム上でのデータ等の管理機能向上のための環境整備では、ORIGEN2 と

HYDORGEOCHEMを対象とし、解析結果の登録、入出力データのバージョン管理等の機能を拡張し、基本システムにおけるデータ管理機能の向上を図った。但し、データ管理機能については、連成解析に含まれる現象解析コード固有の特徴をうまく取り込んで行くためにも、今後の連成領域拡張に伴い機能拡張を段階的に進める必要がある。

添付資料1

連成解析システム上のデータの体系的管理のための環境整備

**ORIGEN2/Hydrogeochem Workbench**

仕様書

平成7年3月

三菱重工業株式会社

## 目次

1. 概要	-----	1
2. 機能構成	-----	2
2.1 データ管理機能	-----	2
2.2 全体制御機能	-----	4
2.3 入力データ作成機能	-----	5
2.4 出力保存機能	-----	7
2.5 データ参照機能	-----	8
2.6 データ編集機能	-----	9
3. システム構成	-----	10
3.1 全体構成	-----	10
3.2 全体制御モジュール	-----	12
3.2.1 機能	-----	12
3.2.2 ユーザインタフェース	-----	13
3.3 入力データ作成モジュール	-----	18
3.3.1 機能	-----	18
3.3.2 ユーザインタフェース	-----	19
3.4 データ変換モジュール	-----	24
3.4.1 機能	-----	24
3.5 データ表示モジュール	-----	26
3.5.1 機能	-----	26
3.5.2 データ表示制御サブモジュール	-----	27
3.5.2.1 機能	-----	27
3.5.2.2 ユーザインタフェース	-----	27
3.5.3 補助情報表示サブモジュール	-----	29
3.5.3.1 機能	-----	29
3.5.3.2 ユーザインタフェース	-----	29
3.5.4 テキスト表示サブモジュール	-----	31
3.5.4.1 機能	-----	31
3.5.4.2 ユーザインタフェース	-----	31
3.5.5 ORIGEN2イベント表示サブモジュール	-----	32
3.5.5.1 機能	-----	32
3.5.5.2 ユーザインタフェース	-----	32
3.5.6 ORIGEN2核種/元素表示サブモジュール	-----	34
3.5.6.1 機能	-----	34
3.5.6.2 ユーザインタフェース	-----	34
3.5.7 Hydrogeochem濃度表示サブモジュール	-----	37

3.5.7.1 機能	-----	37
3.5.7.2 ユーザインタフェース	-----	37
3.6 データ編集モジュール	-----	41
3.6.1 機能	-----	41
3.6.2 ユーザインタフェース	-----	43
3.7 データ管理モジュール	-----	46
3.7.1 機能	-----	46
3.7.2 ユーザインタフェース	-----	46
4. データベース	-----	50
4.1 クラス構造	-----	50

## 図リスト

図1-1 WorkBench 概略図

図2.1-1 データ概念図

図2.1-2 データのバージョン管理

図2.1-3 データベース構成

図2.2-1 全体制御機能

図3.1-1 モジュール全体構成

図3.2-1 全体制御モジュール ウィンドウ構成

図3.3-1 入力データ作成モジュール (ORIGEN2設定時) ウィンドウ構成

図3.3-2 入力データ作成モジュール (Hydrogeochem設定時) ウィンドウ構成

図3.4-1 入出力データ概念図

図3.5-1 データ表示モジュール全体構成

図3.5-2 データ表示制御サブモジュール ウィンドウ構成

図3.5-3 補助情報表示モジュール ウィンドウ構成

図3.5-4 テキスト表示サブモジュール ウィンドウ構成

図3.5-5 ORIGEN2イベント表示サブモジュール ウィンドウ構成

図3.5-6 ORIGEN2元素/核種表示サブモジュール ウィンドウ構成

図3.5-7 Hydrogeochem濃度表示サブモジュール ウィンドウ構成

図3.6-2 データ編集モジュール ウィンドウ構成

図3.7-1 データ管理モジュール ウィンドウ構成

図4.1-1 データベーススキーマ マスターデータ

図4.1-2 データベーススキーマ 解析関連

図4.1-3 データベーススキーマ ORIGEN2関連

図4.1-4 データベーススキーマ Hydrogeochem関連



## 表リスト

表2.3-1 出力テーブル種別

表2.3-2 出力テーブルタイプ

## 1. 概要

ORIGEN2/Hydrogeochem Workbench（以下、Workbench）は、核燃料の燃焼や中性子照射による放射性物質の蓄積・崩壊等に関する計算を行うORIGEN2、及び、地下水中成分の輸送の計算を行うHydrogeochemを用いた解析を効率よく行うことができる環境を提供するものである。

通常、解析業務は入力データの作成、解析コードの実行、解析結果の参照といった作業を繰り返し行う。この際、入出力データ、解析コードの管理、データの表示・編集等を効率よく行うことが解析業務を円滑に行う上で重要なポイントとなる。このため、本システムは以下の特徴を持つものとする。

- ・ 入出力データや解析条件等を表わした情報をデータベースに集中管理することによる再利用性や品質保証の向上
- ・ グラフィカルユーザインタフェースを用いた操作性の優れた環境の提供
- ・ データ編集機能による外部ツールとの連携

Workbenchは図1-1に示すようにORIGEN2とHydrogeochemの2種類の解析コードと以下の機能を持つ複数のToolから構成されている。

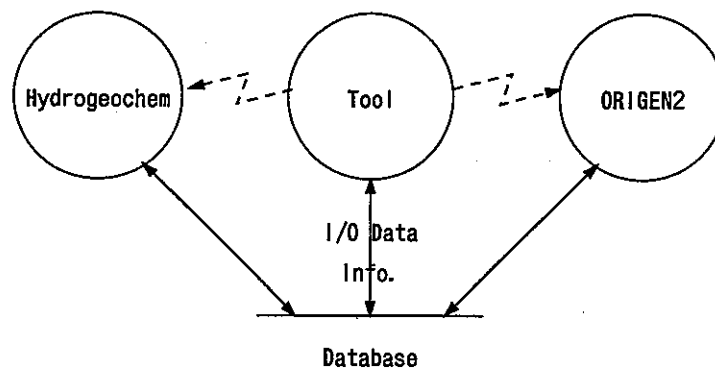


図1-1 Workbench 概略図

- |             |  |
|-------------|--|
| ・ データ管理機能   | データ全体の管理を行う。                                 |
| ・ 全体制御機能    | 解析コードの実行管理やシステム全体の管理を行う。                     |
| ・ 入力データ作成機能 | 解析コードの入力データや解析条件等を表わした補助情報を作成しデータベースに保存する。   |
| ・ 出力データ保存機能 | 解析コードの出力データをデータベースに保存する。                     |
| ・ データ参照機能   | 解析コードの入出力データをデータベースから検索し、表示する。               |
| ・ データ編集機能   | 解析コードの出力データをデータベースから検索し、特定のフォーマットでファイルに出力する。 |

## 2. 機能構成

### 2. 1 データ管理機能

データ管理機能は解析コード、解析コードの入出力データ、解析の条件を表わした補助情報を一括管理し、データベースに保存する。これらのデータは解析名称をキーデータ（データを一意に定めることができるデータ）として、各種データをこれに関連付けることにより構成されている。 図2.1-1はデータ概

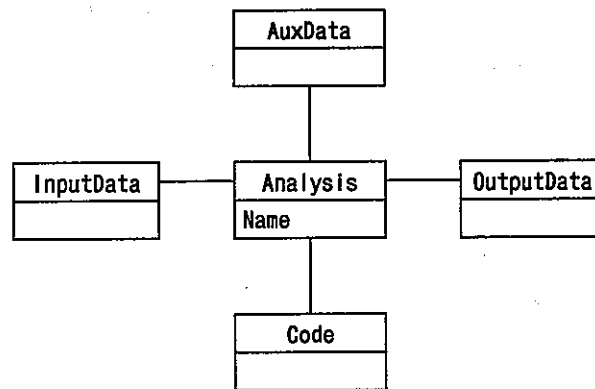


図2.1-1 データ概念図

念図を示しており、1つの解析のデータセットは中心となるAnalysisクラス、補助情報を持ったAuxDataクラス、入力データを表わしたInputDataクラス、出力データを表わしたOutputDataクラス、解析コードの情報を持ったCodeクラスから構成されている。

また、これらのデータセットはバージョン付けがなされており、データサーベイや管理を容易に行うことができる。例えば、既存のAnalysisクラスのインスタンスを利用して解析を行うと、最初にAnalysisクラス、InputDataクラス、AuxDataクラスのインスタンスが複製される。ただし、その際にAnalysisクラスのバージョン属性のみが変更（バージョンアップ）される。次に必要であれば、InputDataクラス、

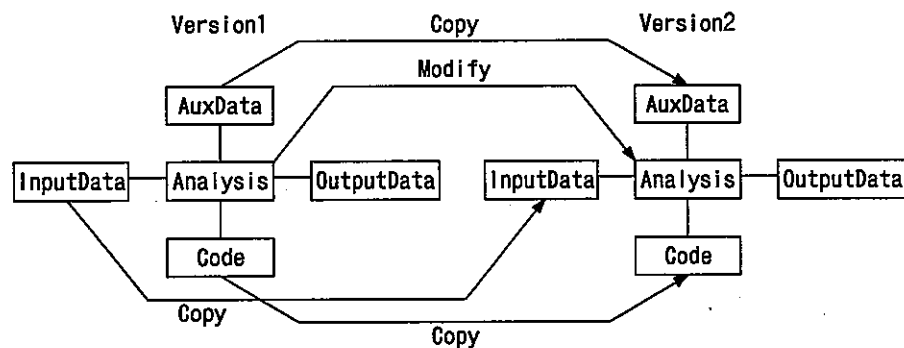


図2.1-2 データのバージョン管理

AuxDataクラスのインスタンスの編集を行う。そして、解析が実行されるとOutputDataクラスがバージョンアップされたAnalysisクラスのインスタンスに関連付けられ、データセット全体がバージョンアップされる。（図2.1-2参照）

データベースへの保存は2段階で管理される。解析対象となっているデータは最初は個々の計算機毎に用意された一時的なデータベース(パーソナルデータベース)に格納され、ユーザは解析の実行、データ表示等の処理をこのデータベースに対して行う。そして、ユーザがこのデータを長期に渡って保存したい場合にはサーバーとなっている計算機上に用意されている永続的なデータベース(グループデータベース)に複製する。

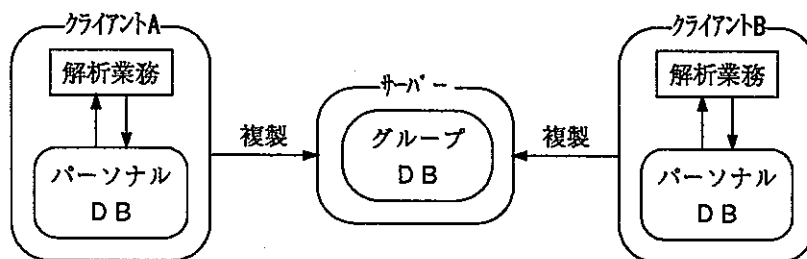


図2.1-3 データベース構成

この機能により、データ管理や解析実行による計算機の負荷を分散することができ、複数の解析者が同時に効率よく解析業務を行うことが可能となる。

## 2. 2 全体制御機能

解析業務を行うには、解析名称の設定、補助情報の入力、入力データの作成、解析コードの実行、出力データの保存、データの参照を行う。全体制御機能はこれらの作業内容に適したアプリケーションの起動や複数のアプリケーションに共通なデータの設定、データベースの管理を行う。

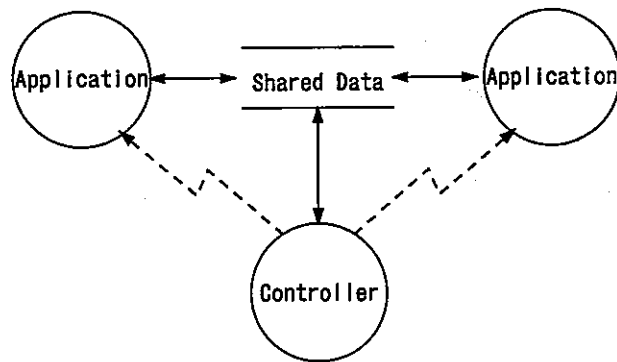


図2.2-1 全体制御機能

## 2. 3 入力データ作成機能

解析コードの入力データや解析条件等を表わした補助情報を作成し、データベースに保存する。入力データは解析コードが持つ特定の形式（解析コードの入力ファイルと同等の形式）とその主要な部分をクラス構造に変換した形式でデータベースに保存される。

補助情報は以下に示すような解析コード間で共通な日付等の情報と解析コード固有の情報を持つ。

- ・ 共通情報
 

作成日	作成年月日
作成者	作成者
コメント	注釈
- ・ ORIGEN2用情報
 

固化体	固化体の名称
イベント	Charge、Discharge、Cooling BR、Reprocess、Cooling BV、Vetrified、Storageの各イベントの期間

入力データには、他の機能との連携や入出力データのクラス構造への変換を確実にを行うため、以下の制約を与えるものとする。

### ORIGEN2の制約

- ・ 1回の実行で出力できるテーブルは表2.3-1中から1種類のみとする。

表2.3-1 出力テーブル種別

Table No.	Description	Unit
1	Isotopic Composition of each element	atom fraction
2	Isotopic Composition of each element	weight fraction
3	Composition	gram-atoms
4	Composition	atom fraction
5	Composition	grams
6	Composition	weight fraction
7	Radioactivity(total)	Gi
8	Radioactivity(total)	fraction
9	Thermal Power	watts
10	Thermal Power	fraction
13	Radioactive Inhalation hazard	m3 air
14	Radioactive Inhalation hazard	fraction
15	Radioactive Ingestion hazard	m3 water
16	Radioactive Ingestion hazard	fraction
17	Chemical ingestion hazard	m3 water
18	Chemical ingestion hazard	fraction
19	Neutron absorption rate	neutrons/sec
20	Neutron absorption rate	fraction
21	Neutron induced fission rate	fissions/sec
22	Neutron induced fission rate	fraction
23	Radioactivity(alpha)	Gi
24	Radioactivity(alpha)	fraction

- ・出力テーブルのタイプは核種の出力を含む。(表2.3-2の1、2、3、5のみ)

表2.3-2 出力テーブルタイプ

NOPTOL(1) NOPTA(1) NOPTF(1)	Table type printed		
	Nuclide	Summary	Element
1	Yes	Yes	Yes
2	Yes	No	Yes
3	Yes	Yes	No
4	No	Yes	Yes
5	Yes	No	No
6	No	No	Yes
7	No	Yes	No
8	No	No	No

- ・時系列データの前に出力されるデータのタイトルは「BEFORE TIME-SERIES DATA (固化体のタイプ名)」とする。
- ・時系列データのタイトルは「TIME-SERIES DATA(\*Y-\*Y)」とする。

入力データの主要な部分は、データ表示の際の検索を効率よく行うためにクラス構造に変換される。変換対象となるデータは以下の通りである。

#### ORIGEN2の対象データ

- ・出力テーブル種別
- ・固化体の名称

#### Hydrogeochemの対象データ

- ・ノードの位置情報

## 2.4 出力保存機能

解析コードの出力ファイルをクラス構造に変換し、データベースに保存する。クラス構造の変換は、データベースの容量を考慮して出力ファイルの全データを変換するのではなく、主要なデータのみを変換しデータベースに保存する。変換対象となるデータは以下の通りである。

### ORIGEN2の対象データ

- ・ イベント終了時点毎の核種のデータ
- ・ 核種の時系列データ

### Hydrogeochemの対象データ

- ・ コンポーネント毎の全ノードの時系列の濃度データ
- ・ 化学種毎の指定ノードの時系列の濃度データ



## 2. 5 データ参照機能

解析コードの入出力データや補助情報をデータベースから検索し、テキスト形式、表形式、グラフ形式のいずれかで表示する。解析コード毎の表示形式は以下の通りである。

### ORIGEN2の表示形式

- ・ イベント終了時点での核種のデータを表形式で表示する。
- ・ 核種、または、元素のデータを横軸を時間、縦軸をパラメータとしてグラフ表示する。
- ・ 入出力データ、Photonライブラリ、Decayライブラリをテキスト形式で表示する。
- ・ 補助情報を表形式で表示する。

### Hydrogeochemの表示形式

- ・ 指定ノードのコンポーネント、または、化学種の濃度を横軸を時間、縦軸をパラメータとしてグラフ表示する。
- ・ 入出力データをテキスト形式で表示する。
- ・ 補助情報を表形式で表示する。

## 2. 6 データ編集機能

解析コードの出力データをデータベースから検索し、以下に示した特定のフォーマットでファイルに出力する。

### ORIGEN2のフォーマット

- ・データ抽出を行う外部アプリケーション (PNC所有) が利用するファイルを出力する。フォーマットは以下の通りである。

固化体の名称

テーブル番号

"ACTINIDES+DAUGHTERS"

"EVENT TABLE"

イベント リスト

イベント 期間

核種データ

"NUCLIDE TABLE"

核種時系列データ

"FISION PRODUCTS"

"EVENT TABLE"

イベント リスト

イベント 期間

核種データ

"NUCLIDE TABLE"

核種時系列データ

### Hydrogeochemのフォーマット

- ・コンポーネントまたは化学種の指定ノードの濃度の時系列データをCSVフォーマットで出力する。フォーマットは以下の通りである。

コンポーネントまたは化学種の名称

ノード番号

時系列データ

### 3. システム構成

#### 3. 1 全体構成

本システムは以下に示す6種類のモジュール、2種類の解析コード、及び、5種類のデータストアから構成されている。

##### モジュール

- ・全体制御モジュール (Controller)  
各モジュールの制御やモジュール間で共通で用いられるデータの管理を行う。
- ・入力データ作成モジュール (DataBuilder)  
補助情報の作成や解析コードの入力データを作成する。
- ・データ変換モジュール (DataConverter)  
解析コードの入出力データをクラス構造に変換し、データベースに保存する。
- ・データ表示モジュール (DataViewer)  
解析コードの入出力データを表形式、グラフ形式、テキスト形式で表示する。
- ・データ編集モジュール (DataEditor)  
解析結果を特定のフォーマットでファイルに出力する。
- ・データ管理モジュール (DataManager)  
パーソナルデータベースとグループデータベースの管理やデータベース内のデータの削除を行う。

##### 解析コード

- ・ORIGEN2  
核燃料の燃焼や中性子照射による放射性物質の蓄積、崩壊を計算する。
- ・Hydrogeochem  
地下水中成分の輸送を計算する。

##### データベース

- ・パーソナルデータベース (Personal Database)  
一時的なワーク領域用のデータベース。解析対象となっているデータのみが保存される。
- ・グループデータベース (Group Database)  
永続的なデータベース。パーソナルデータベースのデータが複製されることにより保存される。
- ・I/Oファイル (I/O File)  
解析データの入出力ファイルが一時的に格納される。
- ・外部ファイル (Ex File)  
外部アプリケーションが用いるファイルを格納する。
- ・共有メモリ (Shared Memory)  
解析名称等のモジュール間の共通な情報が格納される。

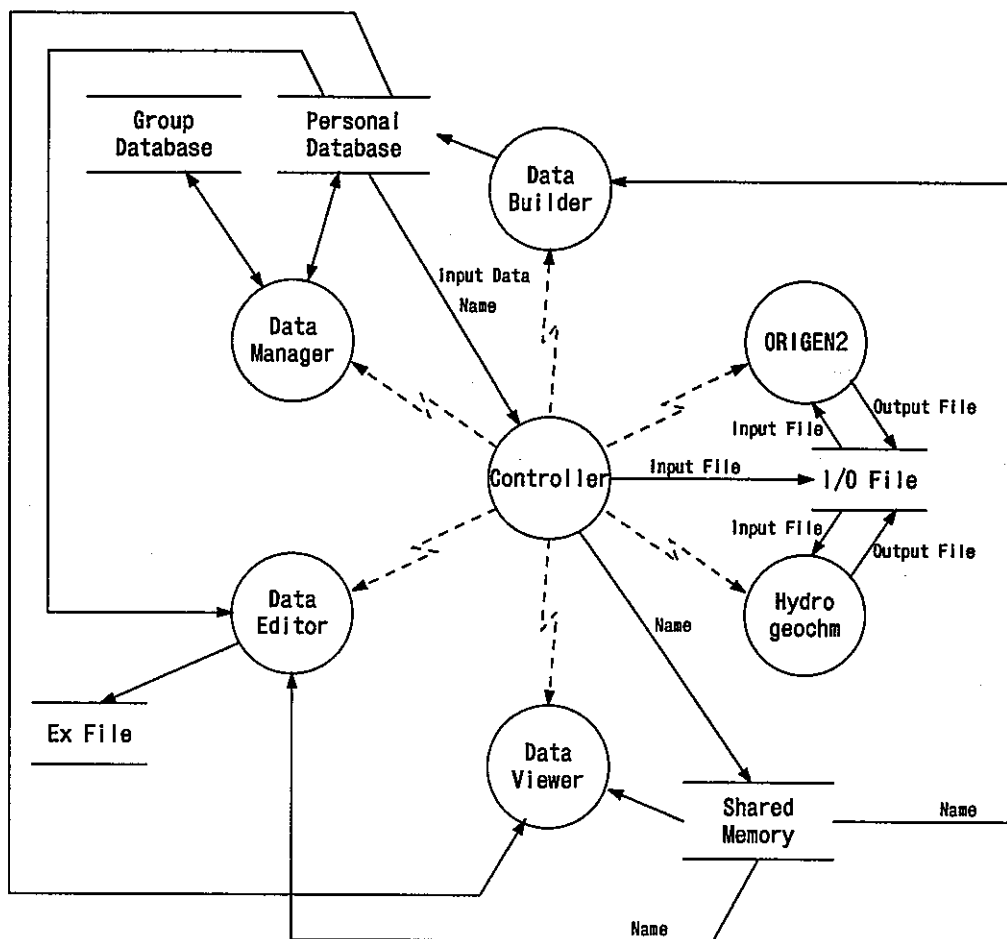


図3.1-1 モジュール全体構成

本システムが推奨するハードウェア構成、及び、ソフトウェアは以下の通りである。

#### ハードウェア

- ・サンマイクロシステム SPARCstation20
- 主メモリ       64MB
- ハードディスク 1GB
- カラーモニタ

#### ソフトウェア

- ・オペレーティングシステム 日本語 Solaris 2.3
- ・ウインドウシステム 日本語 Motif 1.2.2
- ・オブジェクト指向データベース管理システム Versant

## 3. 2 全体制御モジュール (Controller)

### 3. 2. 1 機能

全体制御モジュールは、対象となるグループデータベースの設定、対象となる解析の設定、モジュール／解析コードの制御、及び、モジュール間で共通に用いられる情報の設定を行う。

#### (1) グループデータベース設定

対象となるグループデータベースの設定を行う。設定の際は、あらかじめ作成されたグループデータベースから接続するデータベースを選択する。ただし、パーソナルデータベースにデータが存在する場合は、システム起動時に自動的に前回作業していたグループデータベースが接続される。

#### (2) 解析対象設定

以下の項目を設定する。

- ・ 解析名称  
解析名称を設定する。
- ・ バージョン  
メジャーバージョンとマイナーバージョンを設定する。
- ・ 解析コードの種類  
ORIGEN2またはHydrogeochemとそのバージョンを選択する。

既存の解析を指定した場合は、対象となるデータがグループデータベースからパーソナルデータベースに複製され、各項目が自動的に設定される。ただし、本モジュール起動時にパーソナルデータベース内にデータがある場合は、そのデータが解析対象となり、前回の解析作業を続けて行うことができる。

#### (3) モジュール／解析コードの制御

各モジュールを以下の規則に従って管理する。

- ・ データ作成モジュールは常に起動可能である。ただし、解析コード実行中にデータを保存することはできない。
- ・ データ編集モジュールとデータ管理モジュールの一部機能は一度も解析コードを実行していない場合（解析結果が存在しない場合）は起動できない。また、これらのモジュールと解析コードの実行は排他的である。
- ・ データ表示モジュールの解析結果の参照部分と解析コードの実行は排他的である。

解析コード実行は以下の手順で行われる。

- (a) パーソナルデータベース内に格納されている入力データが入力ファイルに変換される。
- (b) (a)で作成されたファイルを入力として解析コードが起動される。
- (c) 解析コード終了後、出力ファイル内のデータがクラス構造に変換され、パーソナルデータベースに保存される。
- (d) (a)で作成されたファイルが削除される。

#### (4) 共通情報の設定

複数のモジュールで共通に用いられ、かつ、頻繁にアクセスされる情報を共有メモリに格納する。全体

制御モジュールは、この共有メモリの管理（割り付け、解放）を行う。共有メモリに格納される情報は以下の通りである。

- ・解析名称
- ・バージョン
- ・モジュールのプロセス情報（モジュールが稼働中か否か等）

### 3. 2. 2 ユーザインタフェース

#### (1) ウィンドウ構成

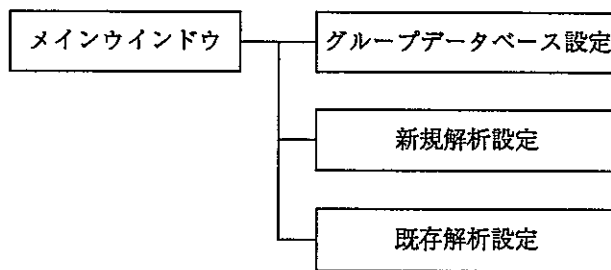


図3.2-1 全体制御モジュール ウィンドウ構成

#### (2) メインウインドウ

本システムを中心となるウィンドウ。解析名称の表示やモジュールの起動を行う。

Hydrogeochem/ORIGEN2 ワークベンチ				
モジュール	解析設定			
データベース:				
解析名称:				
データ作成	解析実行	データ表示	データ編集	データ管理

モジュール

グループデータベース設定...
終了

解析設定

新規解析...
既存解析...

- ・モジュール グループデータベース設定メニュー  
グループデータベース設定ウインドウを表示する。
- ・モジュール 終了メニュー  
本システムを終了する。稼働中の全モジュールは全て強制終了される。
- ・解析設定 新規解析メニュー  
新規解析設定ウインドウを表示する。
- ・解析設定 既存解析メニュー  
既存解析設定ウインドウを表示する。

- ・データ作成ボタン  
入力データ作成モジュールを起動する。
- ・解析実行ボタン  
解析を実行する。
- ・データ表示ボタン  
データ表示モジュールを起動する。
- ・データ編集ボタン  
データ編集モジュールを起動する。
- ・データ管理ボタン  
データ管理モジュールを起動する。
- ・データベースフィールド  
対象となっているグループデータベースの名称が表示される。
- ・解析名称フィールド  
解析名称、バージョン、解析コード名が表示される。

### (3) グループデータベース設定ウインドウ

あらかじめ作成されているグループデータベースの中から対象となるデータベースを選択する。

- ・グループデータベースリストボックス  
グループデータベースの名称の一覧が表示される。  
処理対象とするデータベースを選択する。選択されたデータベースがグループデータベースフィールドに設定される。
- ・グループデータベースフィールド  
処理対象とするグループデータベース名を設定する。
- ・了解ボタン  
グループデータベースフィールドで設定されているデータベースを処理対象とする。パーソナルデータベース内にデータが存在する場合はすべて削除される。
- ・取消ボタン  
ウインドウを消去する。

#### (4) 新規解析設定ウインドウ

解析名称の設定、及び、解析コードの種別を設定する。なお、バージョンは自動的に「1.1」に設定される。

The screenshot shows a dialog box titled "新規解析設定" (New Analysis Settings). It has a text input field for "解析名称:" (Analysis Name). Below it is a section for "解析コード種類" (Analysis Code Type) with two radio button options: "ORIGEN2" and "Hydrogeochem". At the bottom, there are two buttons: "了解" (OK) and "取消" (Cancel).

- ・解析名称フィールド  
解析名称を設定する。
- ・解析コード種類ラジオボタン  
解析コードの種類を選択する。メインウインドウにて各モジュールを選択した場合はここで設定された解析コードに適したモジュールが自動的に起動される。
- ・了解ボタン  
解析名称と解析コードの種類を設定する。設定内容は共有メモリに格納される。
- ・取消ボタン  
ウインドウを消去する。

#### (5) 既存解析設定ウインドウ

既存解析を読み込み、バージョンを設定する。既存解析の読み込みとは、グループデータベースからデータセットを読み込み、パーソナルデータベースに複製することを意味する。



既存解析設定

解析コード種類

ORIGEN2

Hydrogeochem

既存解析

名称	バージョン	

解析

名称:

バージョン:  .

- ・解析コード種類ラジオボタン  
解析コードの種類を選択する。
- ・既存解析 名称リストボックス  
解析コード種類ラジオボタンで選択された解析コードに関する解析の名称の一覧が表示される。  
読み込む解析の名称を選択する。
- ・既存解析 バージョンリストボックス(メジャー)  
既存解析名称リストボックスで選択されている解析のメジャーバージョンの一覧が表示される。  
読み込むメジャーバージョンを選択する。
- ・既存解析 バージョンリストボックス(マイナー)  
既存解析バージョンリストボックス(メジャー)で選択されているメジャーバージョンが持つマイナーバージョンの一覧が表示される。  
読み込むマイナーバージョンを選択する。
- ・解析 名称フィールド  
解析名称を設定する。デフォルトでは、既存解析名称リストボックスで選択されている名称とする。
- ・解析 バージョンフィールド(メジャー)  
メジャーバージョンを設定する。デフォルトでは、既存解析メジャーバージョンリストボックスで選択されているメジャーバージョンとする。
- ・解析 バージョンフィールド(マイナー)  
マイナーバージョンを設定する。デフォルトでは、既存解析マイナーバージョンリストボックスで選択されているマイナーバージョンとする。
- ・了解ボタン  
解析名称リストボックス、バージョンリストボックス(メジャー)、バージョンリストボックス(マイナ

一)で選択されている解析のデータをグループデータベースから読み込み、パーソナルデータベースに複製する。ただし、その際、名称とバージョンは解析名称フィールド、バージョンフィールド(メジャー)、バージョンフィールド(マイナー)に従う。

・取消ボタン

ウインドウを消去する。

### 3. 3 入力データ作成モジュール (DataBuilder)

#### 3. 3. 1 機能

入力データ作成モジュールは、解析コードに関する情報、補助情報、及び、入力データを作成する。入力する項目はメインモジュールで設定された解析コードの種類に依存する。

##### (1) 解析コードに関する情報

解析に用いる解析コードのバージョン、及び、解析コード固有の以下の情報を設定する。

###### ORIGEN2

解析に用いるPhotonライブラリとDecayライブラリを設定する。

なお、解析コードやライブラリに関する情報はあらかじめデータベースに保存しておく必要がある。

##### (2) 補助情報

解析条件等を表わした情報を設定する。設定する項目は解析コード間で共通な項目と固有な項目がある。

###### 共通項目

- ・ 作成者           データの作成者を設定する。デフォルトはログイン名。
- ・ 作成日           データの作成年月日を設定する。デフォルトは現在日。
- ・ コメント        注釈を設定する。

###### ORIGEN2

- ・ 固化体名        固化体の名称を設定する。
- ・ イベント        各イベントの期間を設定する。イベントは以下の通りとする。
  - Charge
  - Discharge
  - Cooling BR
  - Reprocess
  - Cooling BV
  - Vetrified
  - Storage

##### (3) 入力データ作成

解析コードの入力データを作成する。新規作成を容易に行うためテンプレートを用意する。

### 3. 3. 2 ユーザインタフェース

#### ORIGEN2設定時

##### (1) ウィンドウ構成

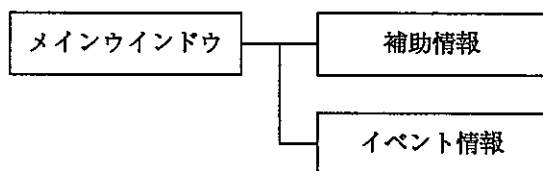
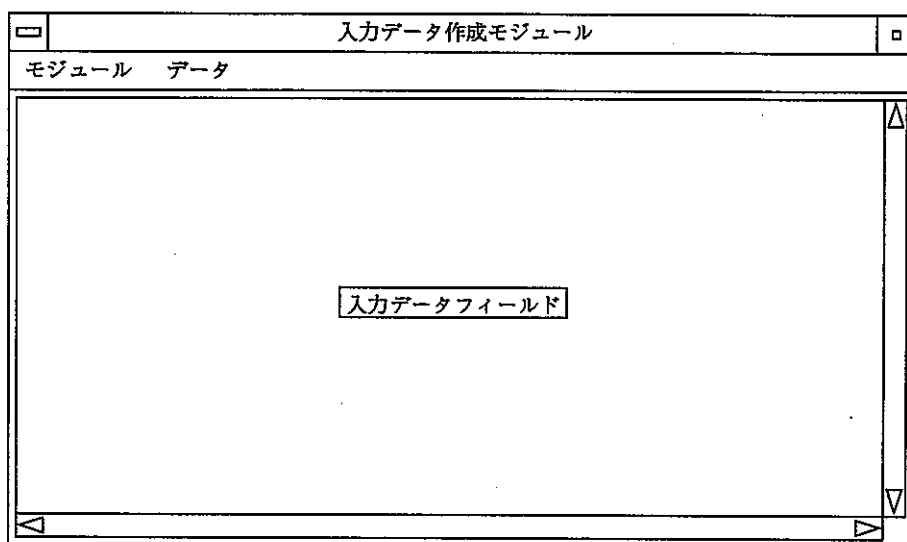


図3. 3-1 入力データ作成モジュール (ORIGEN2設定時) ウィンドウ構成

##### (2) メインウインドウ

解析コードの入力データを設定する。



モジュール	データ
保存	補助情報..
終了	イベント情報..
	テンプレート

- ・モジュール 保存メニュー  
設定内容を保存する。
- ・モジュール 終了メニュー  
データ作成モジュールを終了する。
- ・データ 補助情報メニュー  
補助情報ウィンドウを表示する。
- ・データ イベント情報メニュー

イベント情報ウインドウを表示する。

・テンプレートボタン

入力データフィールドにテンプレートとして用意されたデータを読み込む。

(3) 補助情報ウインドウ

解析コードの情報、補助情報を設定する。

補助情報

解析コード情報

名称 バージョン Photonライブラリ Decayライブラリ

補助情報

作成日  年  月  日

作成者

コメント

了解

・名称リストボックス

データベースに登録されているORIGEN2の固化体の名称が表示される。使用する固化体の名称を選択する。

・バージョンリストボックス

名称リストボックスで選択されている名称に対応したバージョンの一覧が表示される。使用するバージョンを選択する。

・Photonライブラリリストボックス

バージョンリストボックスで選択されているバージョンに対応したPhotonライブラリの一覧が表示される。使用するPhotonライブラリを選択する。

・Decayライブラリリストボックス

バージョンリストボックスで選択されているバージョンに対応したDecayライブラリの一覧が表示される。使用するDecayライブラリを選択する。

・作成日フィールド

データの作成日を設定する。デフォルトは現在日とする。

・作成者フィールド

データの作成者を設定する。デフォルトはログイン名とする。

・コメントフィールド

入力データに関するコメントを設定する。

・了解ボタン

ウインドウを消去する。

#### (4) イベント情報ウインドウ

各イベント毎の終了時の相対日を設定する。

イベント情報	
イベント	
Charge :	<input type="text"/>
Discharge :	<input type="text"/>
Cooling BR :	<input type="text"/>
Reprocess :	<input type="text"/>
Cooling BV :	<input type="text"/>
Vetrified :	<input type="text"/>
Storage :	<input type="text"/>
<input type="button" value="了解"/>	

・イベント Chargeフィールド

Chargeイベントの終了日の相対日を設定する。

・イベント Dischargeフィールド

Dischargeイベントの終了日の相対日を設定する。

・イベント Cooling BRフィールド

Cooling BRイベントの終了日の相対日を設定する。

・イベント Reprocessフィールド

Reprocessイベントの終了日の相対日を設定する。

・イベント Cooling BVフィールド

Cooling BVイベントの終了日の相対日を設定する。

・イベント Vetrifiedフィールド

Vetrifiedイベントの終了日の相対日を設定する。

・イベント Storageフィールド

Storageイベントの終了日の相対日を設定する。

・了解ボタン

ウインドウを消去する。

## Hydrogeochem設定時

### (1) ウィンドウ構成

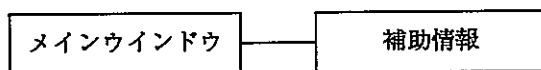
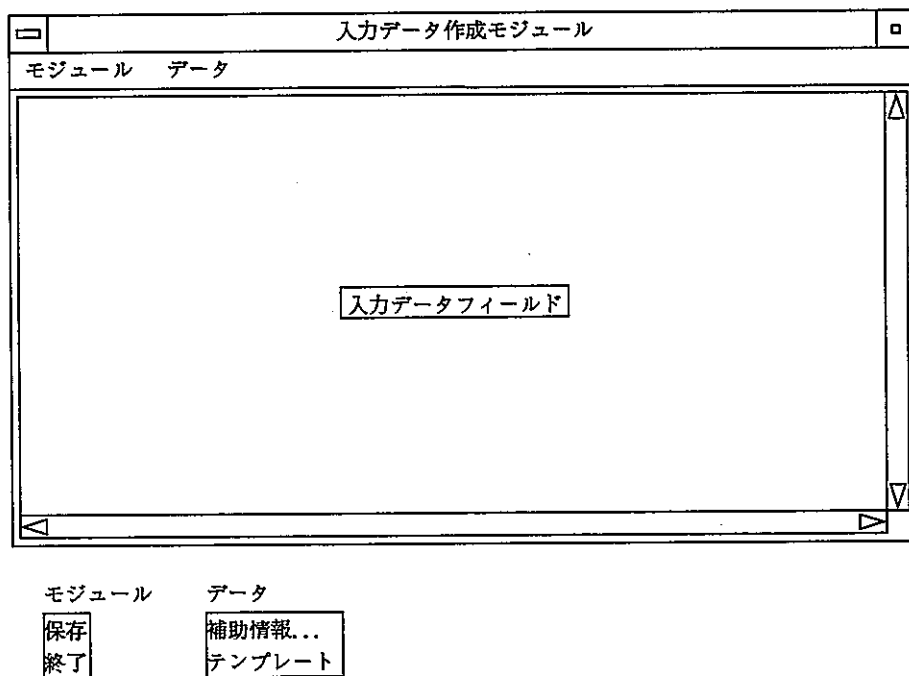


図3.3-2 入力データ作成モジュール (Hydrogeochem設定時) ウィンドウ構成

### (2) メインウインドウ

解析コードの入力データを設定する。



- ・モジュール 保存メニュー  
設定内容を保存する。
- ・モジュール 終了メニュー  
データ作成モジュールを終了する。
- ・データ 補助情報メニュー  
補助情報ウインドウを表示する。
- ・テンプレートボタン  
入力データフィールドにテンプレートとして用意されたデータを読み込む。

### (3) 補助情報ウインドウ

解析コードの情報、補助情報を設定する。

補助情報

解析コード情報

名称                      バージョン

補助情報

作成日  年  月  日

作成者

コメント

了解

- ・名称フィールド  
データベースに登録されているHydrogeochemの名称が表示される。  
使用する名称を選択する。
- ・バージョンフィールド  
名称リストボックスで選択されている名称に対応したバージョンの一覧が表示される。  
使用するバージョンを選択する。
- ・作成日フィールド  
データの作成日を設定する。デフォルトは現在日とする。
- ・作成者フィールド  
データの作成者を設定する。デフォルトはログイン名とする。
- ・コメントフィールド  
入力データに関するコメントを設定する。
- ・了解ボタン  
ウインドウを消去する。



### 3. 4 データ変換モジュール (DataConverter)

#### 3. 4. 1 機能

データ変換モジュールは、解析コードの入力データ/出力ファイルのクラス構造への変換/データベースへの保存やデータベース内のインスタンスから解析コード用入力ファイルの作成を行う。図3.4-1は入出力データのクラス構造の概念を示したものである。入力データの一部はクラス構造に変換され、InputClassクラスに格納される。また、入力データはそのままの形態でInputFileクラスにも格納される。出力データの一部はクラス構造に変換され、OutputClassクラスに格納される。

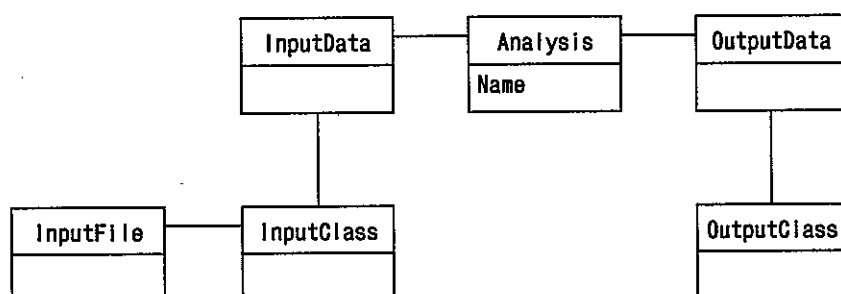


図3.4-1 入出力データ概念図

#### (1) 入力データのクラス構造変換

入力データ作成モジュールで作成された入力データの一部を変換し、InputClassクラスに保存する。変換対象となる内容は以下の通りである。

##### ORIGEN2

- ・ 固化体の名称
- ・ 出力テーブル種別 (内容、単位)

##### Hydrogeochem

- ・ ノードの情報 (位置、タイプ)

#### (2) 入力データの格納

入力データをそのままの形態でInputFileクラスに格納する。

#### (3) 入力ファイル作成

InputFileクラスに格納されたデータをファイルに変換する。解析コード実行直前に利用される。

#### (4) 出力データのクラス構造変換

解析コードの出力ファイルの一部を変換し、OutputClassクラスに格納する。変換対象となる内容は以下の通りである。

##### ORIGEN2

- ・ 各イベントの終了時点毎の核種のデータ

- ・核種の時系列データ

#### Hydrogeochem

- ・コンポーネント毎の全ノードの時系列濃度データ
- ・化学種毎の指定ノードの時系列濃度データ

### 3. 5 データ表示モジュール (DataViewer)

#### 3. 5. 1 機能

データ表示モジュールは、解析コードの入出力データをデータの形態に応じてテキスト、表、グラフのいずれかで表示する。

表示対象となるデータは以下の通りである。

##### (1) テキスト形式表示

###### ORIGEN2

- ・入力データ
- ・ユニット6への出力データ
- ・Photonライブラリ
- ・Decayライブラリ

###### Hydrogeochem

- ・入力データ
- ・出力データ

##### (2) 表形式表示

###### ORIGEN2

- ・各イベントの終了時毎の核種のデータ
- ・補助情報

###### Hydrogeochem

- ・補助情報

##### (3) グラフ形式表示

###### ORIGEN2

- ・核種または元素の時系列データ

###### Hydrogeochem

- ・コンポーネントまたは化学種の濃度の時系列データ

データ表示モジュールは以下のサブモジュールより構成されている。

- ・データ表示制御サブモジュール (DataViewerController)  
サブモジュールの制御を行う。
- ・補助情報表示サブモジュール (AuxInfoViewer)  
補助情報を表形式で表示する。
- ・テキスト表示サブモジュール (TextViewer)  
テキストデータを表示する。
- ・ORIGEN2イベント表示サブモジュール (ORIGENEventViewer)  
ORIGEN2の各イベント終了時の核種のデータを表形式で表示する。
- ・ORIGEN2核種/元素表示サブモジュール (ORIGENTrendViewer)  
ORIGEN2の核種または元素の時系列データをグラフ形式で表示する。
- ・Hydrogeochem濃度表示サブモジュール (HGChemTrendViewer)  
Hydrogeochemのコンポーネントまたは化学種の濃度の時系列データをグラフ形式で表示する。

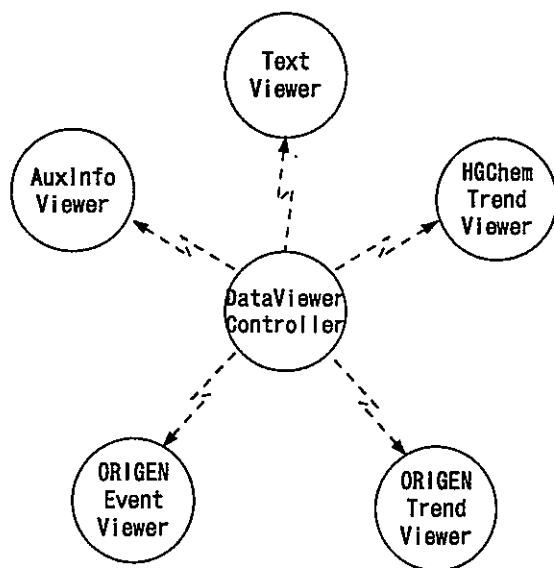


図3.5-1 データ表示モジュール全体構成

### 3.5.2 データ表示制御サブモジュール

#### 3.5.2.1 機能

データ表示制御サブモジュールは補助情報表示サブモジュール、テキストデータサブモジュール、ORIGEN2イベント表示サブモジュール、ORIGEN2核種/元素表示サブモジュール、Hydrogeochem濃度表示サブモジュールの起動、終了を制御する。

#### 3.5.2.2 ユーザインタフェース

##### (1) ウィンドウ構成

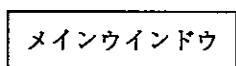
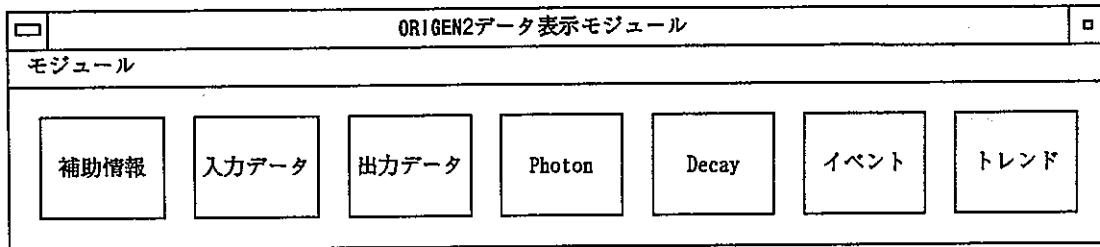


図3.5-2 データ表示制御サブモジュール ウィンドウ構成

##### (2) メインウインドウ

データ表示モジュールの中心となるウィンドウ。表示する項目を選択する。表示項目はメインモジュールで設定された解析コード種別に依存するため、解析コード固有のグラフィカルインタフェースが用意される。

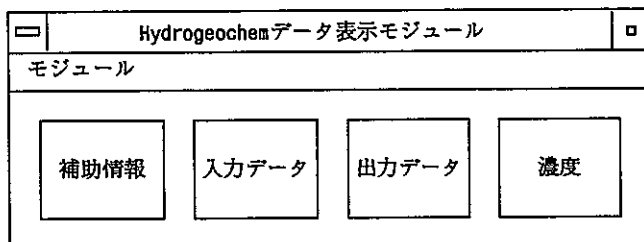
#### ORIGEN2設定時



モジュール  
終了

- ・モジュール 終了メニュー  
データ表示モジュールを終了する。
- ・補助情報ボタン  
補助情報表示サブモジュールを起動する。
- ・入力データボタン  
テキスト表示サブモジュールを起動し、入力データをテキスト形式で表示する。
- ・出力データボタン  
テキストデータ表示サブモジュールを起動し、出力データをテキスト形式で表示する。
- ・Photonボタン  
テキスト表示サブモジュールを起動し、Photonライブラリをテキスト形式で表示する。
- ・Decayボタン  
テキスト表示サブモジュールを起動し、Decayライブラリをテキスト形式で表示する。
- ・イベントボタン  
ORIGEN2核種イベント表示サブモジュールを起動する。
- ・トレンドボタン  
ORIGEN2核種/元素表示サブモジュールを起動する。

#### Hydrogeochem設定時



モジュール  
終了

- ・モジュール 終了メニュー  
データ表示モジュールを終了する。
- ・補助情報ボタン  
補助情報表示サブモジュールを起動する。
- ・入力データボタン  
テキストデータ表示サブモジュールし、入力データをテキスト形式で表示を起動する。

・出力データボタン

テキストデータ表示サブモジュールし、出力データをテキスト形式で表示を起動する。

・濃度ボタン

Hydrogeochem濃度表示サブモジュールを起動する。

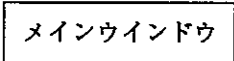
### 3. 5. 3 補助情報表示サブモジュール

#### 3. 5. 3. 1 機能

補助情報表示サブモジュールは入力データ作成モジュールで設定された補助情報を表形式です。補助情報の内容は解析コードの種類に依存する。

#### 3. 5. 3. 2 ユーザインタフェース

##### (1) ウィンドウ構成



メインウインドウ

図3.5-3 補助情報表示サブモジュール ウィンドウ構成

##### (2) メインウインドウ

補助情報が表形式で表示される。解析コード固有のグラフィカルインタフェースが用意される。

**ORIGEN2設定時**

ORIGEN2補助情報表示モジュール							
モジュール							
解析コード							
Photonライブラリ							
Decayライブラリ							
作成日							
作成者							
コメント							
イベント	Charge	Discharge	Cooling BR	Reprocess	Cooling BV	Vetrified	Storage
期間							

モジュール

- ・モジュール 終了メニュー  
補助情報表示サブモジュールを終了する。
- ・解析コードフィールド  
解析コード名称とバージョンが表示される。
- ・Photonライブラリ  
Photonライブラリの名称が表示される。
- ・Decayライブラリ  
Decayライブラリの名称が表示される。
- ・作成日フィールド  
入力データの作成日が表示される。
- ・作成者フィールド  
入力データの作成者が表示される。
- ・コメントフィールド  
コメントが表示される。
- ・イベント表  
各イベント終了時点の相対日付が表示される。

## Hydrogeochem設定時

Hydrogeochem補助情報表示モジュール	
モジュール	
解析コード	
作成日	
作成者	
コメント	

モジュール

終了

- ・モジュール 終了メニュー  
補助情報表示サブモジュールを終了する。
- ・解析コードフィールド  
解析コード名称とバージョンが表示される。
- ・作成日フィールド  
入力データの作成日が表示される。
- ・作成者フィールド  
入力データの作成者が表示される。
- ・コメントフィールド  
コメントが表示される。

### 3. 5. 4 テキスト表示サブモジュール

#### 3. 5. 4. 1 機能

テキストデータを表示する。

#### 3. 5. 4. 2 ユーザインタフェース

##### (1) ウィンドウ構成

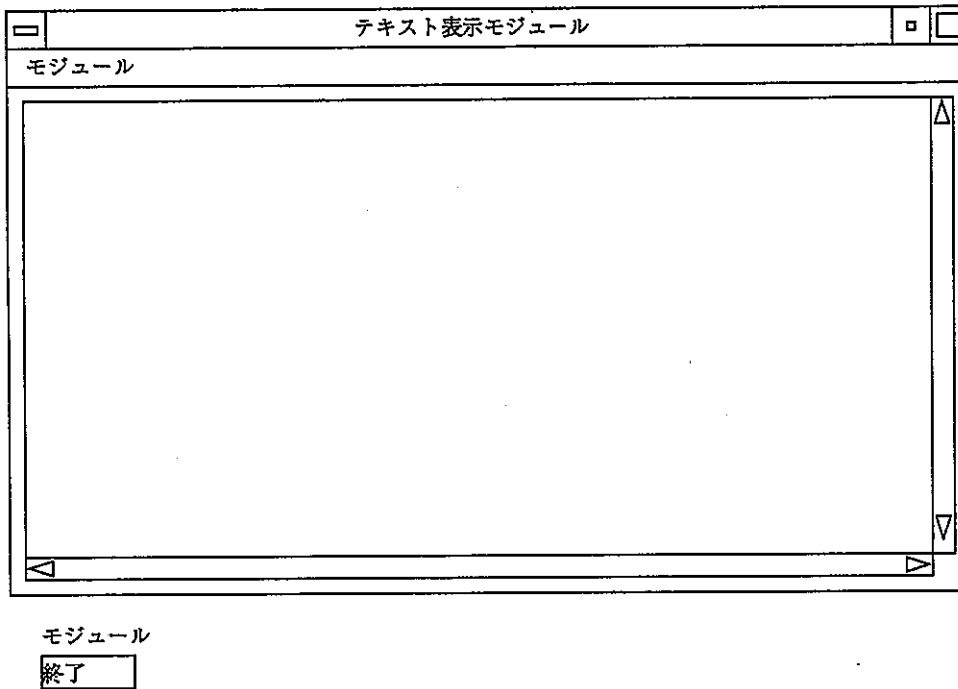
メインウィンドウ

図3.5-4 テキスト表示サブモジュール ウィンドウ構成



## (2) メインウインドウ

テキストデータの名称、及び、テキストデータが表示される。



- ・モジュール 終了メニュー  
テキスト表示サブモジュールを終了する。
- ・データフィールド  
テキストデータが表示される。

### 3. 5. 5 ORIGIN2イベント表示サブモジュール

#### 3. 5. 5. 1 機能

ORIGIN2の各イベント終了時の核種毎のデータをアクチニドとその娘核種または核分裂生成物別に表形式で表示する。表示されるデータの種類はORIGIN2の入力データの出力テーブル番号に依存する。

#### 3. 5. 5. 2 ユーザインタフェース

##### (1) ウィンドウ構成

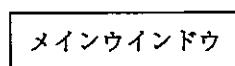


図3.5-5 ORIGIN2イベント表示サブモジュール ウィンドウ構成

(2) メインウインドウ

各イベント終了時の核種のデータの表が表示される。

ORIGEN2イベント表示モジュール							
モジュール 表示							
タイトル表示部							
イベント	Charge	Discharge	Cooling BR	Reprocess	Cooling BV	Vetrified	Storage
期間							
核種名称							
データ表示部							

モジュール

終了

表示

アクチニド+娘核種  
核分裂生成物

- ・モジュール 終了メニュー  
ORIGEN2イベント表示サブモジュールを終了する。
- ・表示 アクチニド+娘核種メニュー  
表示対象の核種をアクチニドとその娘核種とする。
- ・表示 核分裂生成物メニュー  
表示対象の核種を核分裂生成物とする。
- ・タイトル表示部  
データの種類が表示される。表示されるデータの種類は入力データの出力テーブル番号に依存する。
- ・期間表示部  
イベントの期間（入力データ作成モジュールで入力された値）が表示される。
- ・核種名称表示部  
表示メニューに応じて核種の名称の一覧が表示される。
- ・データ表示部  
各イベント毎の核種のデータが表示される。

### 3. 5. 6 ORIGEN2核種/元素表示サブモジュール

#### 3. 5. 6. 1 機能

ORIGEN2の核種または元素の時系列データをフリーソフトの「gnuplot」を用いてグラフ形式で表示する。元素のデータは核種のデータの総和により求められる。表示されるデータの種類はORIGEN2の入力データの出力テーブル番号に依存する。

#### 3. 5. 6. 2 ユーザインタフェース

##### (1) ウィンドウ構成

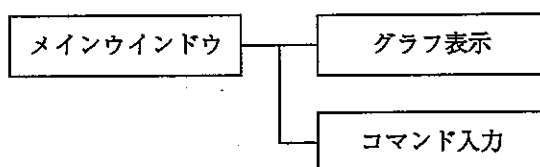


図3.5-6 ORIGEN2元素/核種表示サブモジュール ウィンドウ構成

##### (2) メインウインドウ

グラフ表示する元素/核種を選択や、グラフのスタイル、軸設定などの設定を行う。設定が終了するとグラフ表示ウインドウにグラフが表示される。

ORIGEN2元素/核種表示モジュール							
モジュール	データ						
<table border="1"> <tr> <td>元素</td> <td>核種</td> <td>グラフ</td> </tr> <tr> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> </table>	元素	核種	グラフ	<input type="text"/>	<input type="text"/>	<input type="text"/>	
元素	核種	グラフ					
<input type="text"/>	<input type="text"/>	<input type="text"/>					
タイトル: <input type="text"/>							
<b>X軸</b>							
最大: <input type="text"/>	最小: <input type="text"/> ステップ: <input type="text"/>						
ラベル: <input type="text"/>							
<b>Y軸</b>							
最大: <input type="text"/>	最小: <input type="text"/> ステップ: <input type="text"/>						
ラベル: <input type="text"/>							

モジュール

表示...  
終了

データ

削除  
すべて削除

・モジュール 表示メニュー

グラフリストボックスの項目をグラフ表示したウインドウを表示する。

・モジュール 終了メニュー

ORIGEN2元素/核種表示モジュールを終了する。

・データ 削除メニュー

グラフ表示リストボックスで選択されている項目をリストボックスから削除する。

・データ すべて削除メニュー

グラフ表示リストボックスのすべての項目をリストボックスから削除する。

・元素リストボックス

元素の一覧が表示される。グラフ表示する元素を選択する。

・核種リストボックス

元素リストボックスで選択されている元素が持つ核種の一覧、及び、「Total」が表示される。

グラフ表示する核種を選択する。ただし、「Total」が選択された場合は元素のデータをグラフ表示する。選択された項目がグラフリストボックスに追加される。

・グラフリストボックス

グラフ表示する元素、または、核種が表示される。

・タイトルフィールド

グラフのタイトルを設定する。

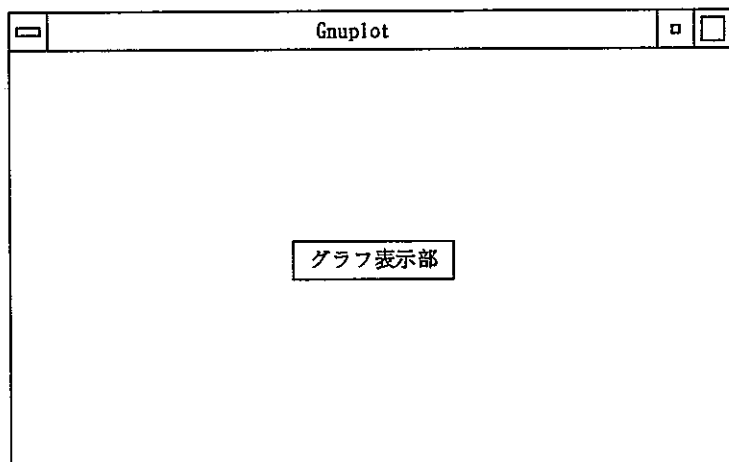
・X軸 最大フィールド

X軸の最大値を設定する。

- ・ X軸 最小フィールド  
X軸の最小値を設定する。
- ・ X軸 ステップフィールド  
X軸のステップを設定する。最小、ステップのいずれかのフィールドが空白の場合自動スケールになる。
- ・ X軸 ラベルフィールド  
X軸のラベルを設定する。
- ・ Y軸 最大フィールド  
Y軸の最大値を設定する。
- ・ Y軸 最小フィールド  
Y軸の最小値を設定する。
- ・ Y軸 ステップフィールド  
Y軸のステップを設定する。最小、ステップのいずれかのフィールドが空白の場合自動スケールになる。
- ・ Y軸 ラベルフィールド  
Y軸のラベルを設定する。

## (2) グラフ表示ウインドウ

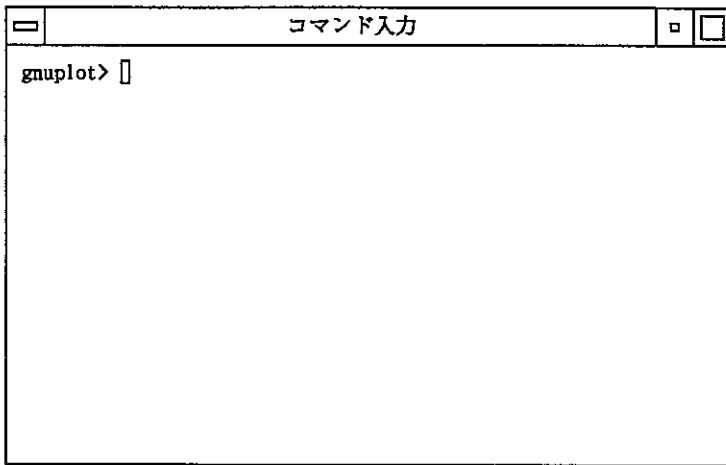
メインウインドウのグラフィストボックスに設定されている元素の持つ核種のデータがグラフ表示される。



- ・ グラフ表示部  
元素または核種のグラフが表示される。

## (3) コマンド入力ウインドウ

グラフの表示スタイル等を変更するため、Gnuplotのコマンドを入力する。



### 3. 5. 7 Hydrogeochem濃度表示サブモジュール

#### 3. 5. 7. 1 機能

Hydrogeochemのコンポーネントまたは化学種の濃度の時系列データを節点毎にフリーソフトの「gnuplot」を用いてグラフ形式で表示する。

#### 3. 5. 7. 2 ユーザインタフェース

##### (1) ウィンドウ構成

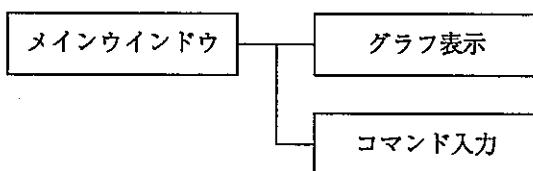


図3.5-7 Hydrogeochem濃度表示サブモジュール ウィンドウ構成

##### (2) メインウインドウ

グラフ表示する節点とコンポーネント／化学種の選択や、グラフのスタイル、軸設定などの設定を行う。設定が終了するとグラフ表示ウインドウにグラフが表示される。

Hydrogeochem濃度表示モジュール		
モジュール	データ	
節点番号: <input type="text"/>	コンポーネント/タイプ	タイプ/化学種
表示	<input type="list"/>	<input type="list"/>
<input type="radio"/> コンポーネント		
<input type="radio"/> 化学種		
グラフ <input type="list"/>		
タイトル: <input type="text"/>		
X軸 最大: <input type="text"/> 最小: <input type="text"/> ステップ: <input type="text"/>		
ラベル: <input type="text"/>		
Y軸 最大: <input type="text"/> 最小: <input type="text"/> ステップ: <input type="text"/>		
ラベル: <input type="text"/>		

モジュール

表示...  
終了

データ

削除  
すべて削除

・モジュール 表示メニュー

グラフリストボックスの項目をグラフ表示したウインドウを表示する。

・モジュール 終了メニュー

Hydrogeochem濃度表示モジュールを終了する。

・データ 削除メニュー

グラフ表示リストボックスで選択されている項目をリストボックスから削除する。

・データ すべて削除メニュー

グラフ表示リストボックスのすべての項目をリストボックスから削除する。

・節点フィールド

グラフ表示する節点を設定する。

・表示ラジオボタン

表示対象となるデータを選択する。

・コンポーネント/タイプリストボックス

表示ラジオボタンで選択されている項目に対応したデータ（コンポーネントが選択されている場合コンポーネント、化学種が選択されている場合化学種のデータタイプ）の一覧が表示される。

グラフ表示するデータを選択する。

・タイプ/化学種リストボックス

コンポーネント/タイプリストボックスで選択されている項目に対応したデータ（コンポーネントのデータタイプまたは化学種）の一覧が表示される。

グラフ表示するデータを選択する。選択されたデータがグラフリストボックスに表示される。

・グラフリストボックス

グラフ表示するコンポーネントまたは化学種が表示される。

・タイトルフィールド

グラフのタイトルを設定する。

・X軸 最大フィールド

X軸の最大値を設定する。

・X軸 最小フィールド

X軸の最小値を設定する。

・X軸 ステップフィールド

X軸のステップを設定する。最小、ステップのいずれかのフィールドが空白の場合自動スケールになる。

・X軸 ラベルフィールド

X軸のラベルを設定する。

・Y軸 最大フィールド

Y軸の最大値を設定する。

・Y軸 最小フィールド

Y軸の最小値を設定する。

・Y軸 ステップフィールド

Y軸のステップを設定する。最小、ステップのいずれかのフィールドが空白の場合自動スケールになる。

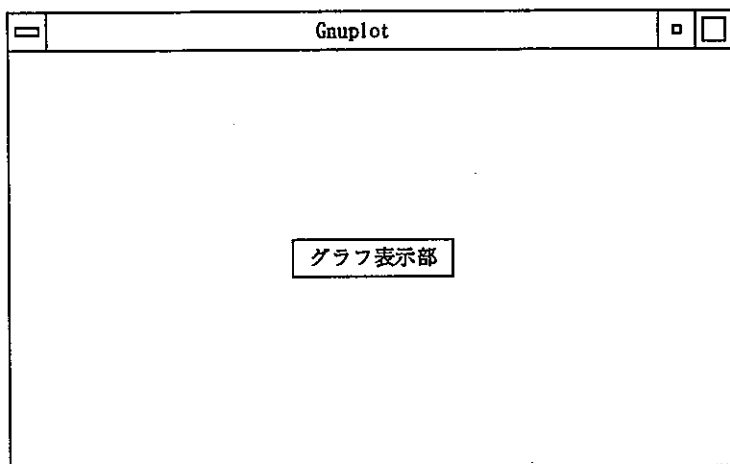
・Y軸 ラベルフィールド

Y軸のラベルを設定する。

## (2) グラフ表示ウインドウ

メインウインドウのグラフリストボックスに設定されているコンポーネントまたは化学種の濃度データがグラフ表示される。



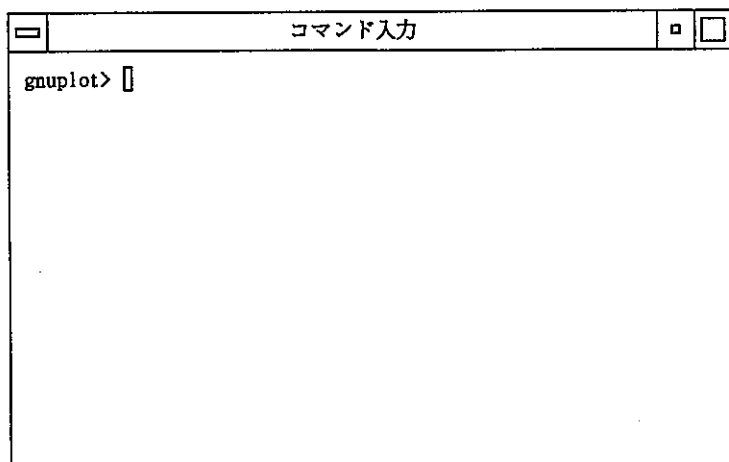


・グラフ表示部

コンポーネントまたは化学種のグラフが表示される。

(3) コマンド入力ウインドウ

グラフの表示スタイル等を変更するため、Gnuplotのコマンドを入力する。



### 3. 6 データ編集モジュール (DataEditor)

#### 3. 6. 1 機能

データ編集モジュールは解析コードの出力データを指定したファイル名で以下に示した特定のフォーマットでファイルに出力する。

#### ORIGEN2 パターン 1

番号	名称	内容
1	固化体名称	補助情報の固化体名称 (入力データ作成モジュールで設定)
2	テーブル番号	出力テーブル番号 (入力データに設定)
3	ACTINIDES+DAUGHTERS	文字列
4	EVENT TABLE	文字列
5	イベントリスト	Charge、Discharge... のイベントのリスト
6	イベント期間	各イベント終了時の相対時間
7	核種データ	各イベント終了時の核種のデータのリスト
8	NUCLIDE TABLE	文字列
9	時間	時間データ
10	核種データ	核種の時系列データのリスト
11	FISSION PRODUCTS	文字列
12	EVENT TABLE	文字列
13	イベントリスト	Charge、Discharge... のイベントのリスト
14	イベント期間	各イベント終了時の相対時間
15	核種データ	各イベント終了時の核種のデータのリスト
16	NUCLIDE TABLE	文字列
17	時間	時間データ
18	核種データ	核種の時系列データのリスト

3番から10まではアクチニドとその娘核種のデータ

11番から18番までは核生成物のデータ

ORIGEN2 パターン 2

番号	名称	内容
1	固化体名称	補助情報の固化体名称 (入力データ作成モジュールで設定)
2	テーブル番号	出力テーブル番号 (入力データに設定)
3	ACTINIDES+DAUGHTERS	文字列
4	EVENT TABLE	文字列
5	イベントリスト	Charge、Discharge...のイベントのリスト
6	イベント期間	各イベント終了時の相対時間
7	核種データ	各イベント終了時の核種のデータのリスト
8	NUCLIDE TABLE	文字列
9	時間	時間データ 最大で9列 (9タイムステップ分)
10	核種データ	核種の時系列データのリスト 最大で9列 (9タイムステップ分)
11	FISSION PRODUCTS	文字列
12	EVENT TABLE	文字列
13	イベントリスト	Charge、Discharge...のイベントのリスト
14	イベント期間	各イベント終了時の相対時間
15	核種データ	各イベント終了時の核種のデータのリスト
16	NUCLIDE TABLE	文字列
17	時間	時間データ 最大で9列 (9タイムステップ分)
18	核種データ	核種の時系列データのリスト 最大で9列 (9タイムステップ分)

3番から10まではアクチニドとその娘核種のデータ

11番から18番までは核生成物のデータ

9番、10番、17番、18番は最大で9列とし、それを超える場合は8番、9番、10番または16番、17番、18番の繰り返しとなる。

Hydrogeochem

番号	名称	内容
1	名称	コンポーネントまたは化学種の名称
2	ノード番号	ノードの番号
3	濃度時系列データ	時間、濃度データの組の時系列データのリスト

名称、ノード番号、濃度時系列データはCSVフォーマットとする。

### 3. 6. 2 ユーザインタフェース

#### (1) ウィンドウ構成

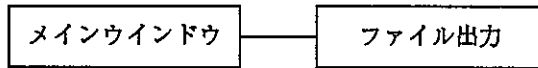


図3.6-2 データ編集モジュール ウィンドウ構成

#### (2) メインウインドウ

ファイルに出力するデータ項目、及び、ファイル名称を設定する。解析コードの種類により出力できるデータ項目は異なるため、グラフィカルユーザインタフェースは解析コードの種類固有となる。

#### ORIGEN2設定時

☐	データ編集モジュール	☐
モジュール		
出力タイプ		
◇	1行	
◇	複数行	

モジュール

出力...
終了

・モジュール 出力メニュー

ファイル出力ウインドウを表示する。

・モジュール 終了メニュー

データ編集モジュールを編集する。

・出力タイプラジオボタン

出力するフォーマットの種類を選択する。

1行 核種時系列データを1行で出力する。(パターン1に対応)

複数行 核種時系列データを10列づつ区切って出力する。(パターン2に対応)

#### Hydrogeochem設定時

データ編集モジュール

---

モジュール

節点番号:

出力

コンポーネント

化学種

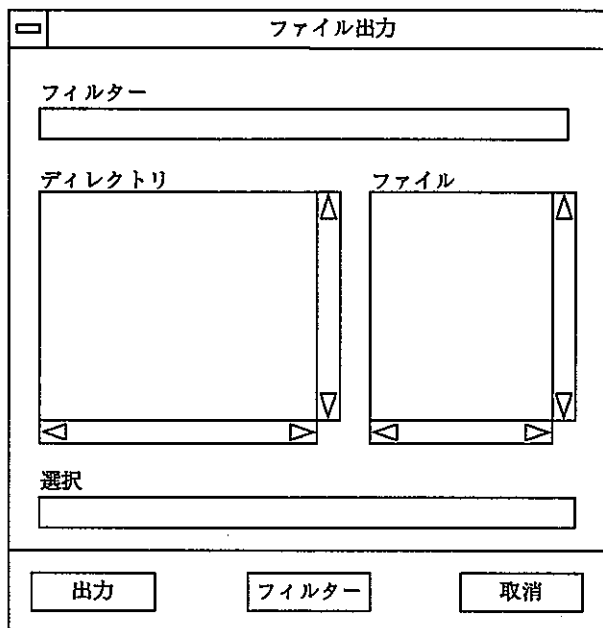
コンポーネント/タイプ

タイプ/化学種

モジュール

- ・モジュール 出力メニュー  
ファイル出力ウインドウを表示する。
- ・モジュール 終了メニュー  
データ編集モジュールを編集する。
- ・節点番号フィールド  
出力する節点番号を設定する。
- ・出力ラジオボタン  
出力するデータの種類を選択する。
- ・コンポーネント/タイプリストボックス  
表示ラジオボタンで選択されている項目に対応したデータ（コンポーネントが選択されている場合コンポーネント、化学種が選択されている場合化学種のデータタイプ）の一覧が表示される。  
出力するデータを選択する。
- ・タイプ/化学種リストボックス  
コンポーネント/タイプリストボックスで選択されている項目に対応したデータ（コンポーネントのデータタイプまたは化学種）の一覧が表示される。  
出力するデータを選択する。

## (2) ファイル出力ウインドウ



- ・フィルターフィールド  
ファイルを検索するディレクトリを設定する。
- ・ディレクトリリストボックス  
ディレクトリの一覧が表示される。  
フィルターとして設定するディレクトリを選択する。
- ・ファイルリストボックス  
フィルターフィールドで設定されているディレクトリ内のファイルの一覧が表示される。  
ファイル名を選択する。選択されたファイル名が選択フィールドに設定される。
- ・選択フィールド  
出力するファイル名を設定する。
- ・出力ボタン  
メインウィンドウに設定されている条件に従って、選択フィールドで設定されている名称のファイルに出力する。
- ・フィルターボタン  
フィルターフィールドに設定されているディレクトリ内のファイルを検索する。
- ・取消ボタン  
ウィンドウを消去する。

### 3. 7 データ管理モジュール

#### 3. 7. 1 機能

パーソナルデータベース、グループデータベースの管理を行う。

##### (1) パーソナルデータベース／グループデータベースの複製

パーソナルデータベース内のデータをグループデータベースに複製する。複製の際には解析名称、バージョンは以下の規則に従ってデフォルトが設定される。

###### ・新規解析時

名称は全体制御モジュールで設定された値となる。

バージョンは設定された名称の解析が存在しない場合は「1.1」、存在する場合は「最新メジャーバージョン.1」となる。

###### ・既存解析時

名称、バージョン共、全体制御モジュールで設定された値となる。

##### (2) 解析表示

グループデータベース内の解析の名称、バージョンの一覧、及び、補助情報の一部を表示する。

##### (3) 解析削除

グループデータベース内の解析を削除する。解析の削除とは解析名称、補助情報、入出力データ等の解析に関する全てのデータセットを削除することを意味する。

##### (4) パーソナルデータベース削除

パーソナルデータベース内のデータを全て削除する。

#### 3. 7. 2 ユーザインタフェース

##### (1) ウィンドウ構成

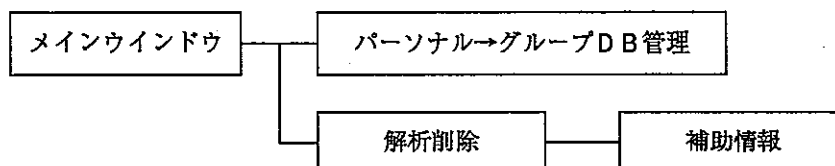


図3.7-1 データ管理モジュール ウィンドウ構成

## (2) メインウインドウ

本モジュールの中心となるウインドウ。

データ管理モジュール		
モジュール		
パーソナル →グループ DB管理	解析削除	パーソナル

モジュール

終了

- ・モジュール 終了メニュー  
データ管理モジュールを終了する。
- ・パーソナル→グループDB管理ボタン  
データベース管理ウインドウを表示する。
- ・解析削除ボタン  
解析削除ウインドウを表示する。
- ・パーソナルボタン  
パーソナルデータベースの内容を削除する。

## (2) パーソナル→グループDB管理ウインドウ

パーソナルデータベースの内容をグループデータベースに複製する。

データベース管理モジュール	
モジュール	
解析名称:	<input type="text"/>
バージョン:	<input type="text"/> . <input type="text"/>

モジュール

保存  
終了

- ・モジュール 保存メニュー  
解析名称フィールド、バージョンフィールドに従ってパーソナルデータベースの内容をグループデータベースに複製する。
- ・モジュール 終了メニュー  
データ管理モジュールを終了する。
- ・解析名称フィールド  
解析名称を設定する。
- ・バージョンフィールド  
メジャーバージョンとマイナーバージョンを設定する。



### (3) 解析削除ウインドウ

グループデータベース内の解析のデータセットを削除する。

- ・モジュール 削除ボタン  
解析名称リストボックス、バージョンリストボックスに従ってデータセットをグループデータベースから削除する。
- ・モジュール 補助情報ボタン  
補助情報ウインドウを表示する。
- ・モジュール 終了ボタン  
解析削除モジュールを終了する。
- ・解析コード種類ラジオボタン  
解析コードの種類を選択する。
- ・解析名称リストボックス  
解析コード種類ラジオボタンで設定された解析コードの解析名称の一覧が表示される。  
削除する解析名称を選択する。
- ・バージョンリストボックス (メジャー)  
解析名称リストボックスで選択されている解析のメジャーバージョンの一覧、および「全て」が表示される。  
削除するメジャーバージョンまたは「全て」を選択する。「全て」を選択した場合は指定した解析名称を持つデータセットが全て削除対象となる。
- ・バージョンリストボックス (マイナー)  
解析名称リストボックスで選択されている解析でバージョンリストボックス (メジャー) で選択されているメジャーバージョンが持つマイナーバージョンの一覧、および「全て」が表示される。「全て」を選択した場合は指定した解析名称、メジャーバージョンを持つデータセットが全て削除対象となる。  
削除するマイナーバージョンまたは「全て」を選択する。

#### (4) 補助情報ウインドウ

解析コード間で共通に設定されている補助情報を表示する。

データ管理モジュール 補助情報

補助情報

作成日: 年 月 日

作成者:

コメント:

了解

- ・了解ボタン  
ウインドウを消去する。
- ・作成日フィールド  
データの作成日が表示される。
- ・作成者  
データの作成者が表示される。
- ・コメント  
データのコメントが表示される。

## 4. データベース

### 4. 1 クラス構造

オブジェクト指向データベース管理システムVERSANTに実装されているクラス構造を示したオブジェクトダイアグラムを以下に示す。

図4.1-1は、解析コードやライブラリ等のマスター系のデータを格納するためのクラス構造を示す。

図4.1-2に示すオブジェクトダイアグラムは、解析の名称や補助情報等の解析関連のデータの中心となるクラスである。

図4.1-3に示すオブジェクトダイアグラムはORIGEN2に関連したデータを格納するクラスである。

図4.1-4に示すオブジェクトダイアグラムはHydrogeochemに関連したデータを格納するクラスである。

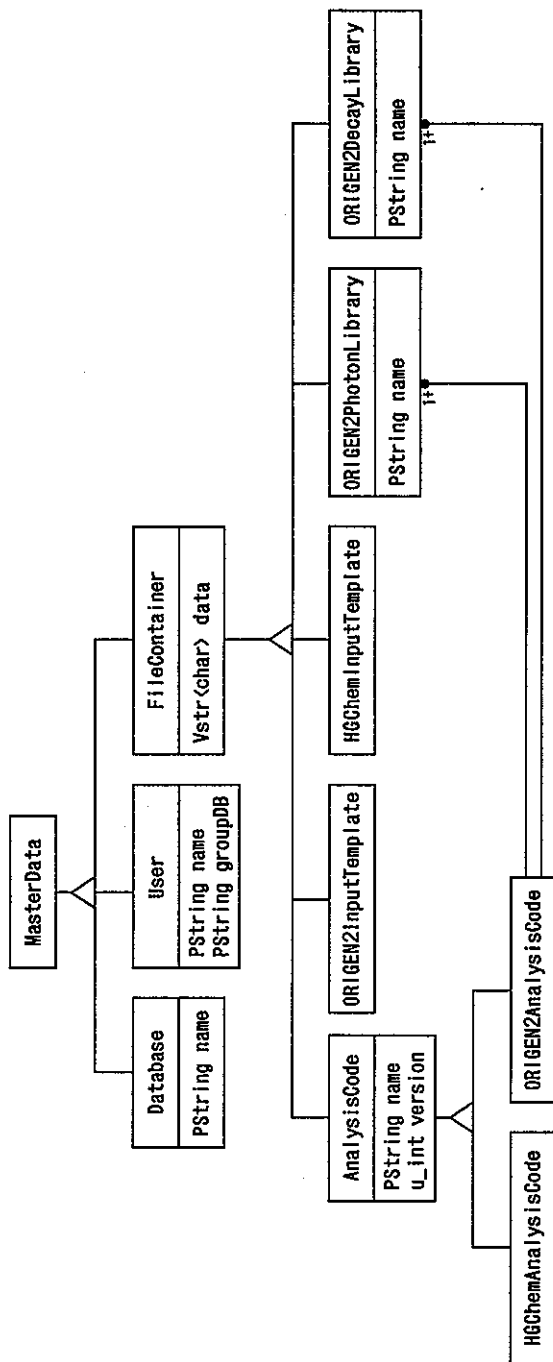


図4.1-1 データベーススキーマ マスタデータ

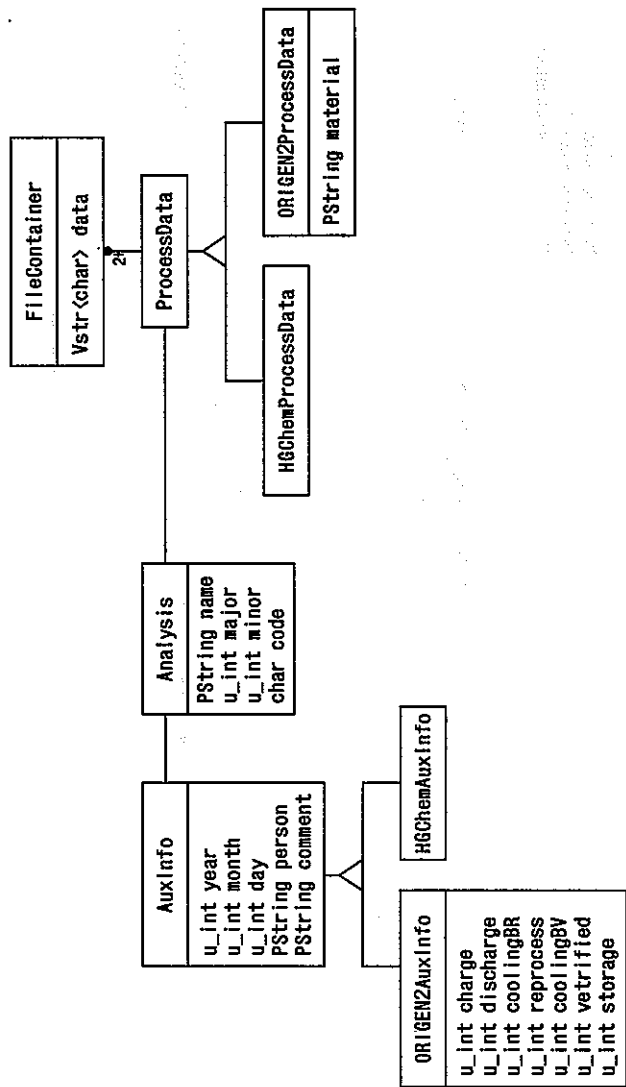


図4.1-2 データベーススキーマ 解析関連

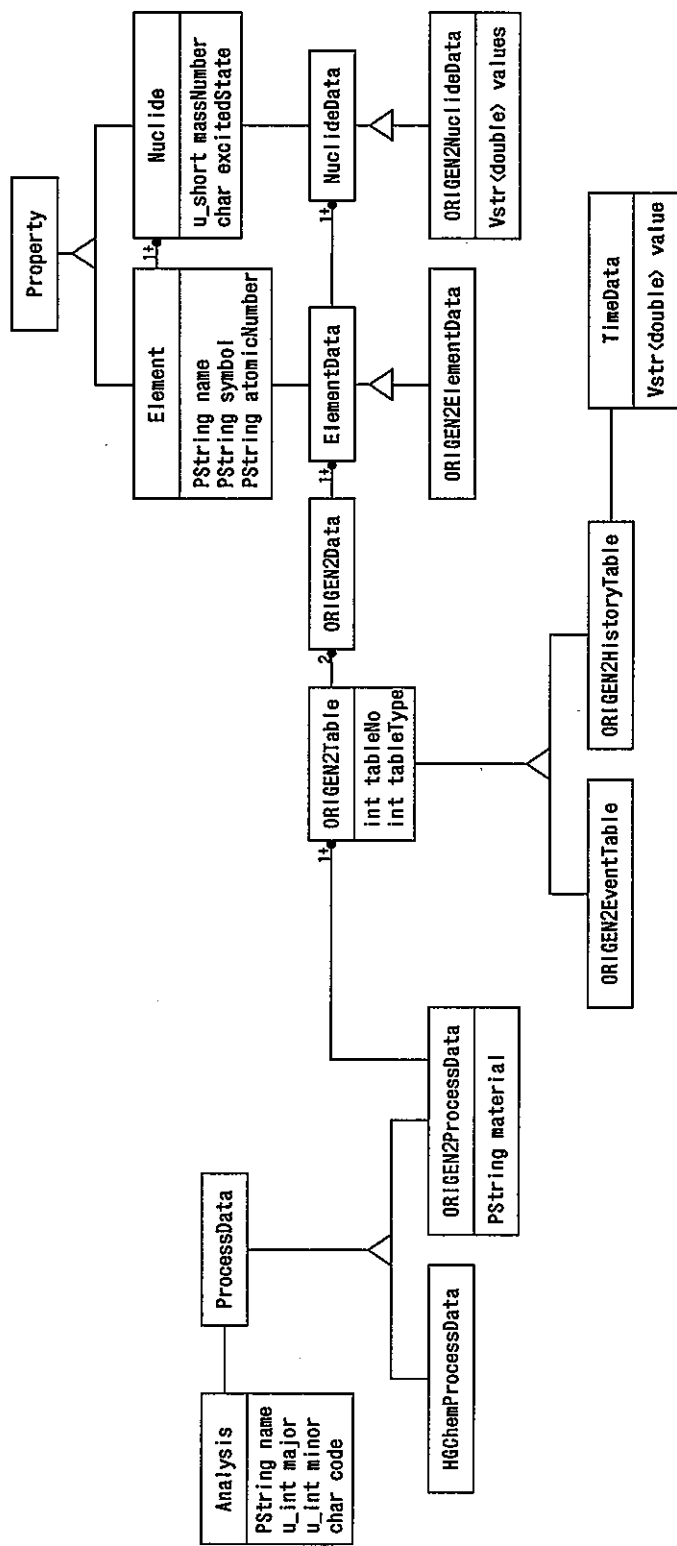


図4.1-3 データベーススキーマ ORIGEN2関連

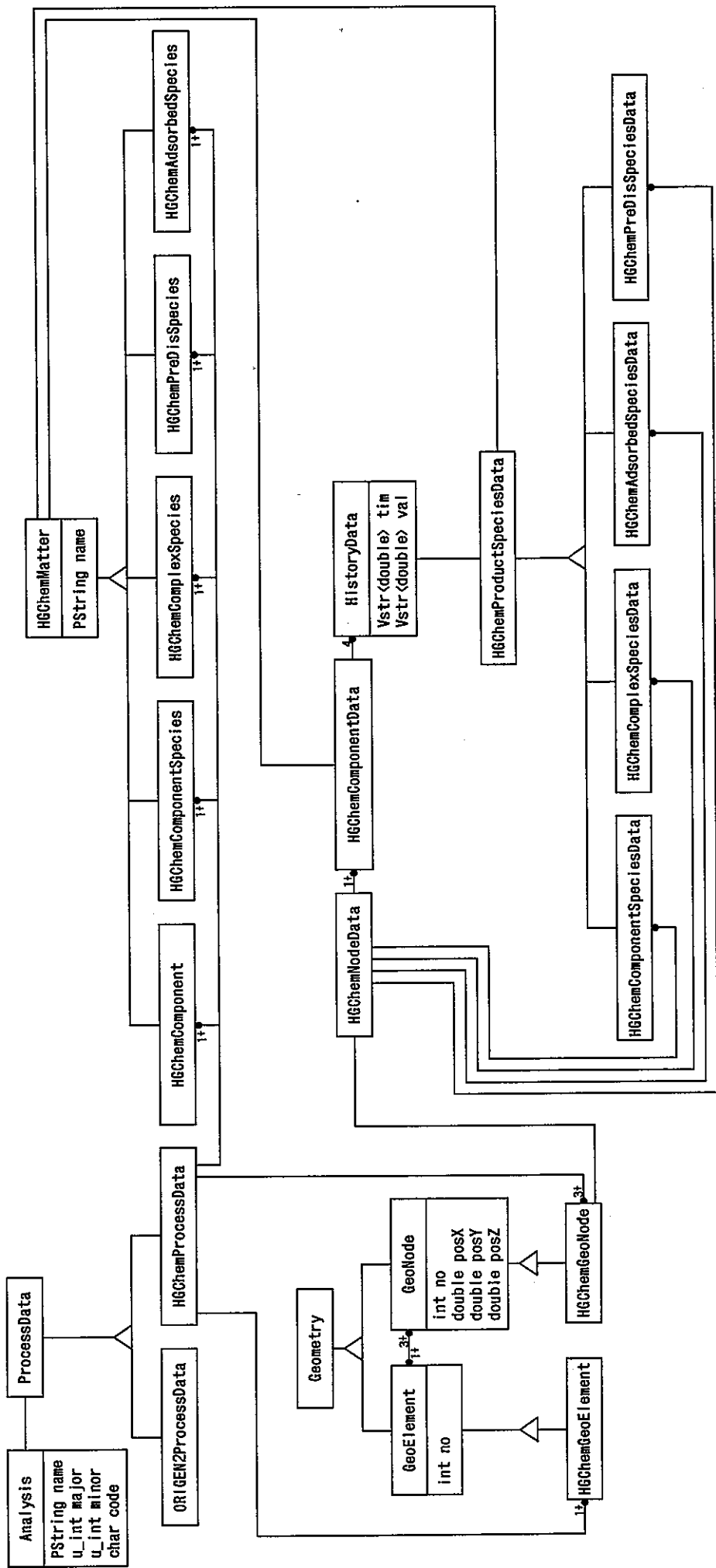


図4.1-4 データベーススキーマ Hydrogeochem関連