

# 統合プラットフォーム

アドバンスソフト(株)  
(株)富士総合研究所

小池秀耀  
松原 聖

# 統合プラットフォーム(PSE)

複数の応用ソフトウェアやデータベースを駆使して  
より大規模で複雑な問題を解析することを可能とする  
コンピュータ環境

# 統合プラットフォーム（PSE:problem Solving Environment）の必要性

計算科学技術は飛躍的發展期にあるが、ハードウェアの進歩や情報技術の蓄積を生かし、技術革新を実現する為には、当面、次のような技術的課題を克服する必要に迫られている。

ネットワーク上で複数のソフトウェアを統合化し、より複雑で大規模な問題を解決するソフトウェアを構築する技術の確立

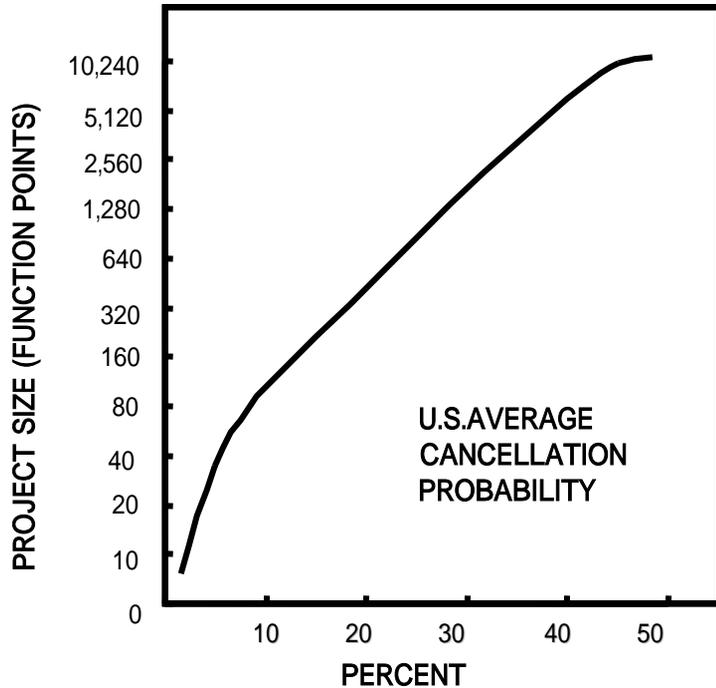
マルチスケール、マルチフィジックスといった多原理統合型シミュレーション技術の確立

タスク並列処理技術の確立

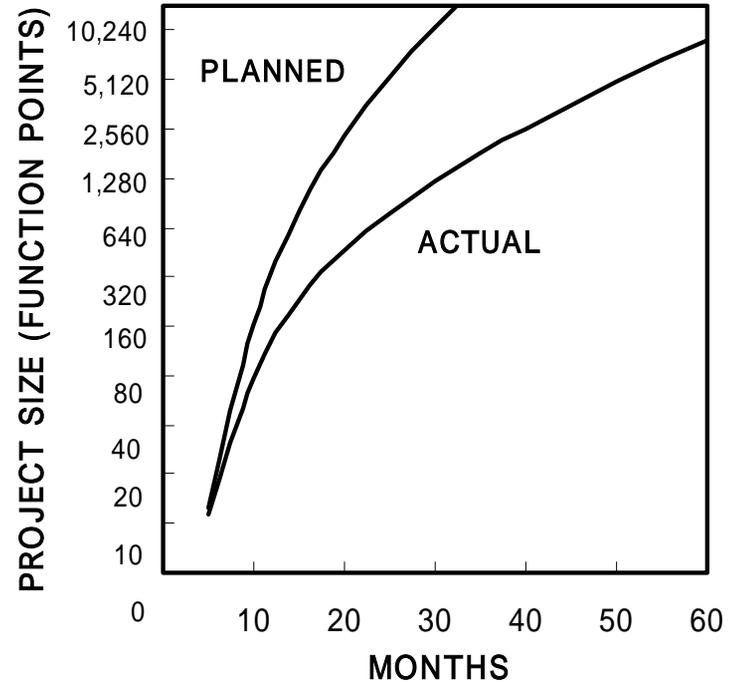
# ソフトウェア開発の難易度

| ソフトウェアの規模 | 難易度    | 開発人数         | 開発期間   |
|-----------|--------|--------------|--------|
| 1千行以下     | 非常に簡単  | 1            | 数週間    |
| 1万行まで     | 簡単     | 2, 3         | 半年     |
| 10万行まで    | やや難しい  | 6 ~ 10       | 2, 3年  |
| 1百万行まで    | 難しい    | 50 ~ 100     | 3 ~ 5年 |
| 1千万行まで    | 不可能に近い | 経験者はほとんどいない  |        |
| 1千万行以上    | 非常識    | Star Wars 計画 |        |

# 大規模ソフトウェアの開発の困難さ



プロジェクトの失敗率



プロジェクトの計画と  
実際の差異

SOURCE: Software Productivity Research

# 統合プラットフォーム(PSE)に要求される機能

ネットワーク上に分散したコンピューティング資源(ソフトウェア、データ)を、統合化して、より高度なソフトウェア・システムを開発するための機能

多数のソフトウェアやデータベースを駆使した解析を、正確に再現するための機能。すなわち解析手順、使用するソフトウェア、データ、計算結果、中間の計算結果、計算条件などを保存し、解析を正確に再現できる機能

コンピュータネットワーク上のコンピューティング資源に自由にアクセスできる分散処理機能

巨大なデータをネットワーク上で効率よく蓄積、管理し、必要に応じて、必要なデータを取り出せる機能

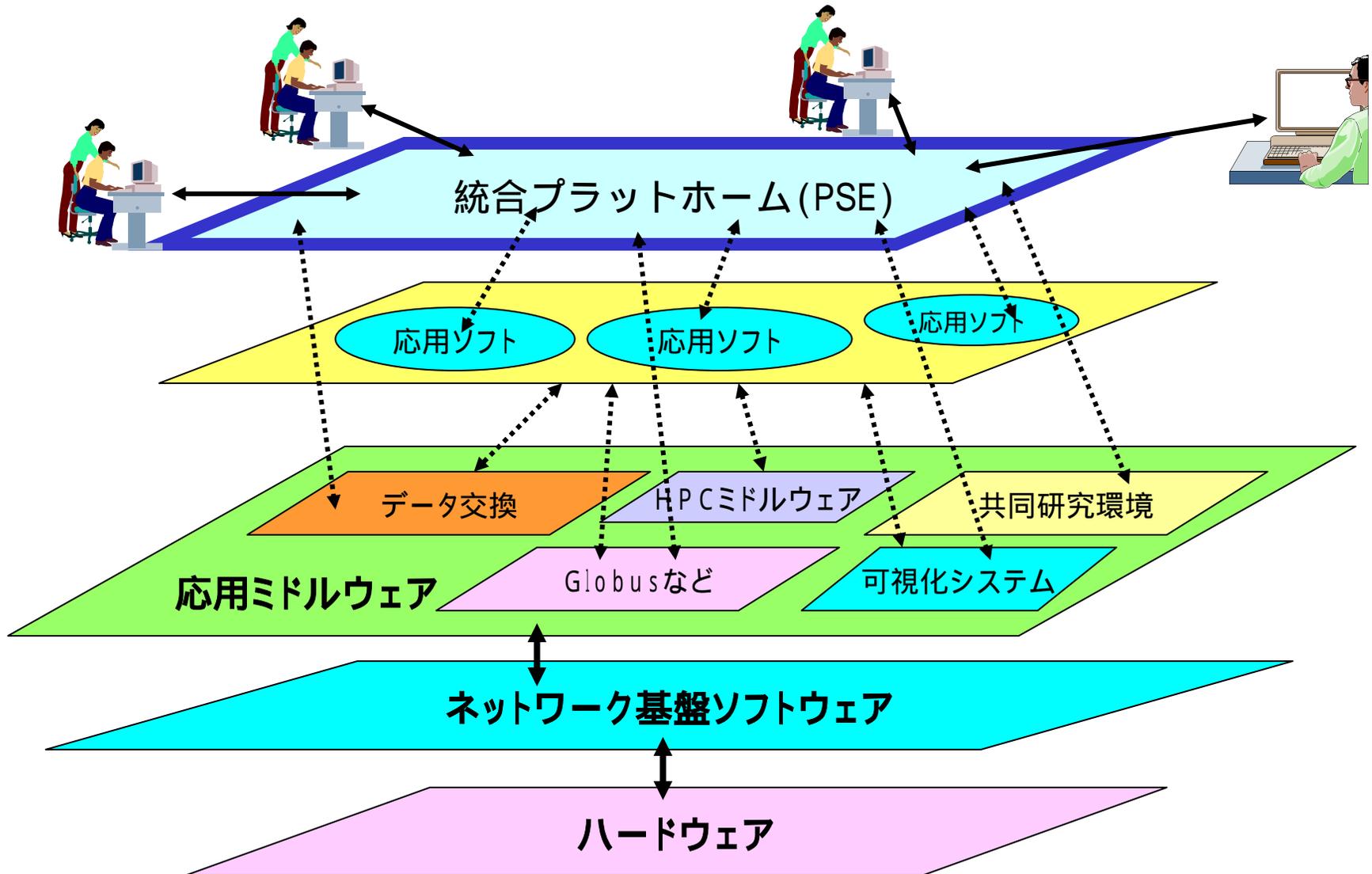
ビジュアライゼーションなどの革新的ヒューマン・インターフェイス

タスク並列処理ソフトウェアの開発ツール

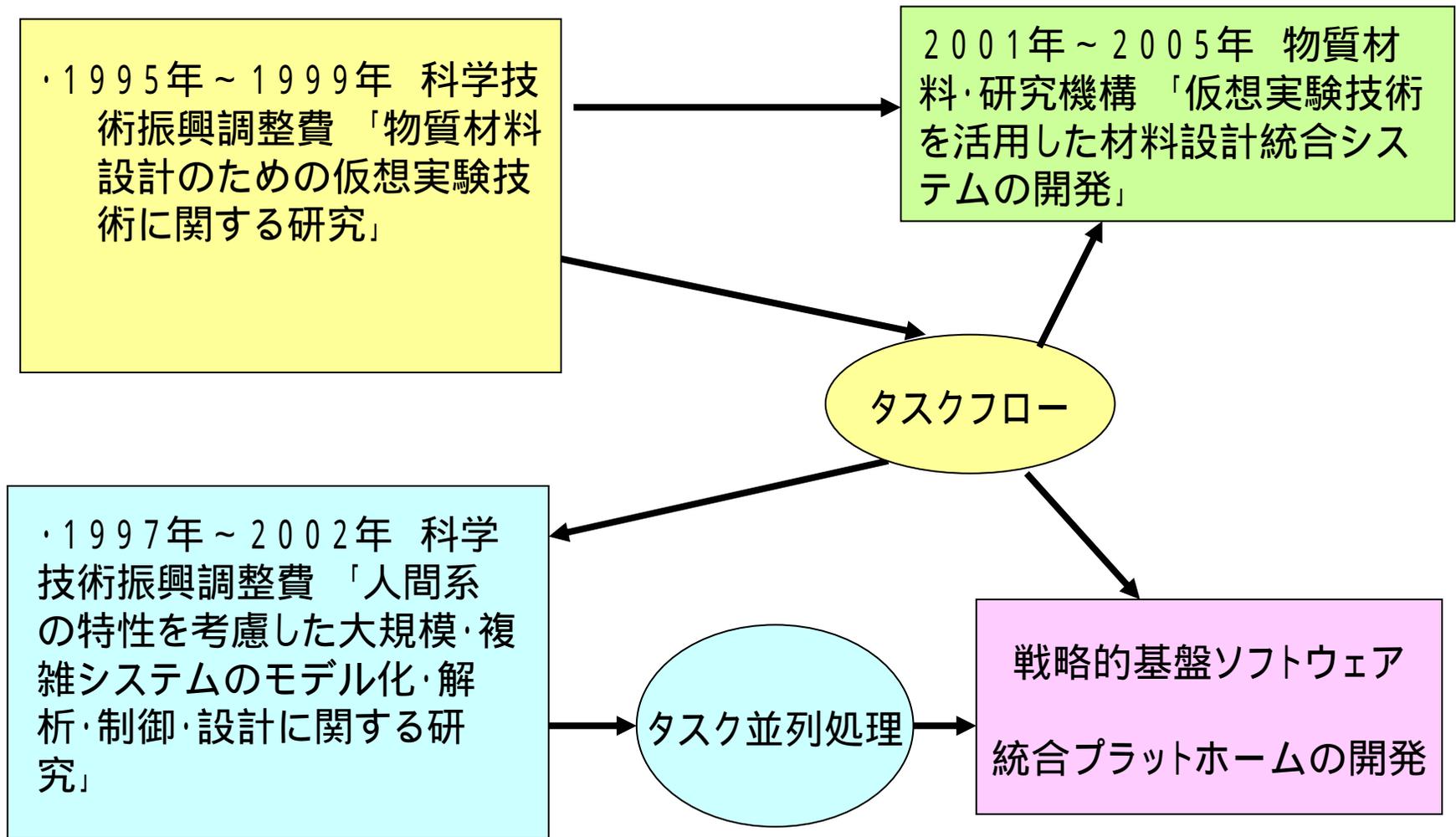
大規模シミュレーションをリアルタイムで対話的に制御するステアリング機能

上記の機能をユーザーが自由に駆使できるワークベンチ機能

# 統合プラットフォームの概念図



# 統合プラットフォーム (PSE)の研究開発の経緯



# 大規模、複雑な問題解決のための統合シミュレーション環境(PSE)

## Problem Solving Environment (PSE)

### PSEの構成

ワークベンチ

コンテンツ / リソース

### ワークベンチの主な機能

タスクフロー

タスク並列シミュレータ (タスク並列プログラミングツール)

ステアリングシステム

データフロー管理 (データ管理・保存・転送)

ヒューマン・インターフェイス

## 統合プラットフォームと従来のソフトウェアとの比較

| 項目             | 統合プラットフォーム  | 従来のシステム  |
|----------------|---|--|
| 全般             | タスクフローにより解析・設計に必要なソフトウェア群が統合化されている。解析・設計用コンピュータ環境 | ソルバー、データベース、可視化などのソフトをGUIから利用可能。設計用に統合化されていない。 |
| 解析<br>設計<br>支援 | 解析・設計シナリオに沿って全作業を統合的に支援                           | 単独の作業を支援                                       |
| 拡張<br>性        | タスクフローの追加により簡単に拡張できる。                             | 解析ツール、データベースを追加                                |
| 対象             | タスクフローを記述すれば任意の解析・設計が可能                           | 対象が限定される                                       |

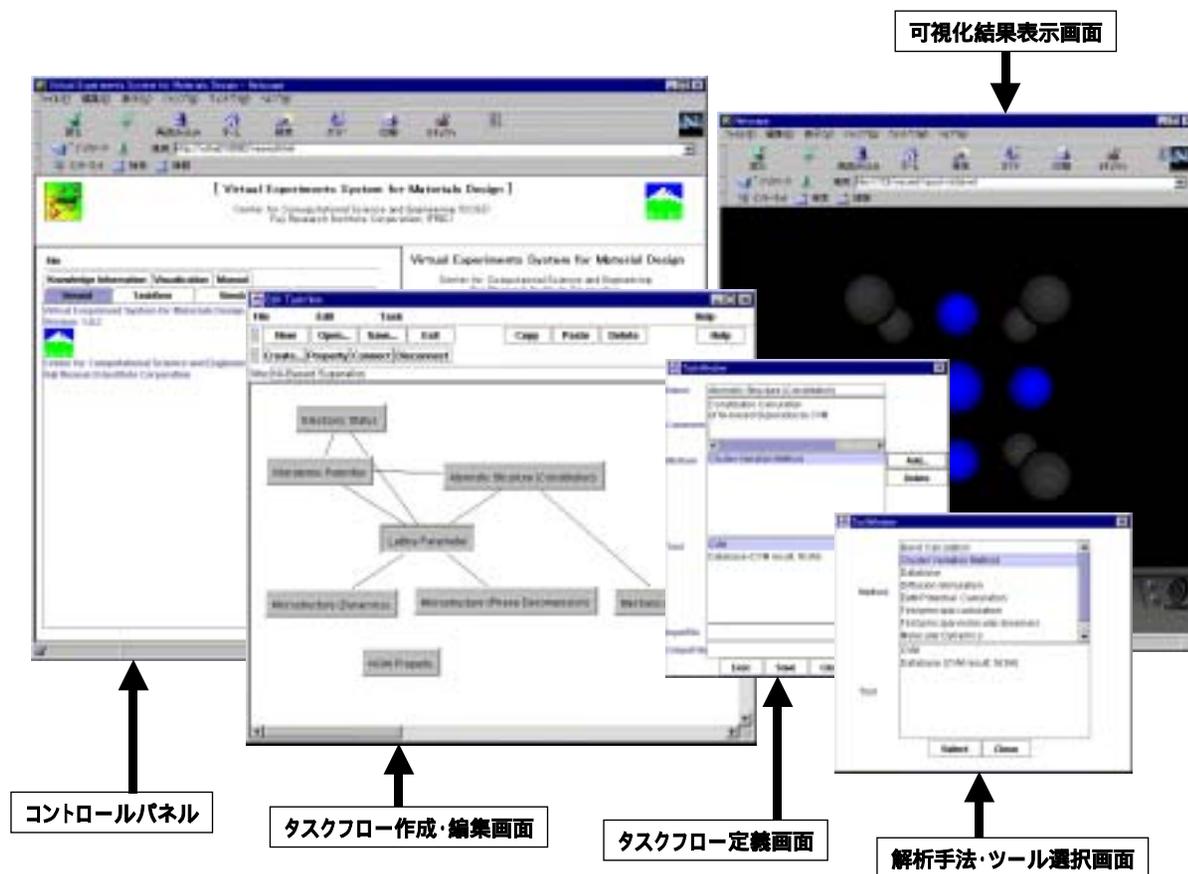
# タスクフロー

- 要素技術を統合化するための方法として、**スクフロー**という手法を確立した。
- タスクフローの概念および特徴
  - 解析・設計の一連の作業を**タスク**と定義する。
  - 解析・設計はタスクの連結で表現でき、このタスクの連結を**タスクフロー**と呼ぶ。
  - 要求仕様に基づき、タスクフローを作成し解析・設計を行うことができる。
  - 各タスクにおける処理に対して様々な解析手法、解析ツールを関連づけられる。
  - タスクフローの保存により、解析・設計シナリオ、解析・設計ノウハウ、成功事例等を蓄積、再利用が可能である。

# 統合プラットフォームのヒューマン・インターフェースの例

- 統合プラットフォームのヒューマン・インターフェース

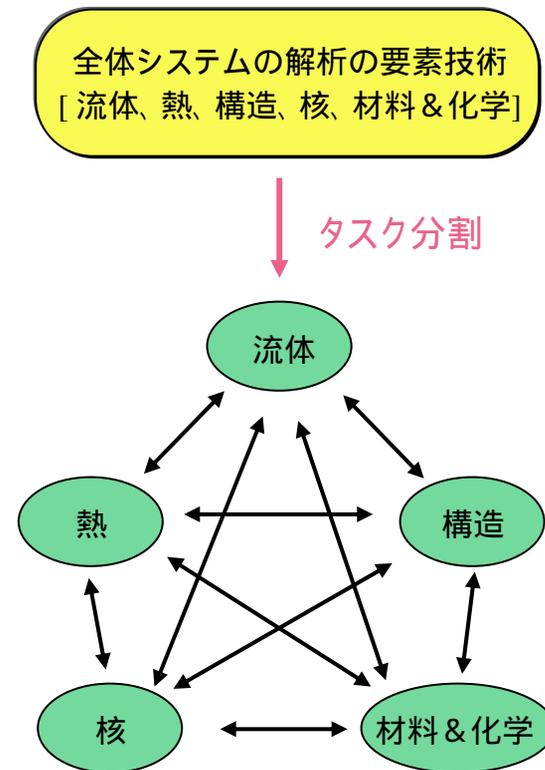
- コントロール・パネル
- タスク作成・編集
- タスク定義
- シミュレーション
  - 入力データ作成
  - プログラム実行
- データベース
  - データベース構築
  - データベース検索
- 知識情報処理
- 可視化



# タスク並列処理シミュレータ

大規模・複雑なシステムを解析、制御する場合、タスク分割に基づく並列処理が必要不可欠であるが、タスク分割に基づく並列処理技術はいまだ確立していない。本研究ではタスク並列処理技術の基礎的技術の確立を目的としている。

原子炉総合解析を例にしたタスク分割の例



# タスク並列処理言語開発の概要

大規模・複雑なシステムでは、様々な処理(タスク)が並列で動作していくことが考えられる。これらを表現、評価する技術の確立が必要。



タスク記述言語の設計、シミュレータを作成

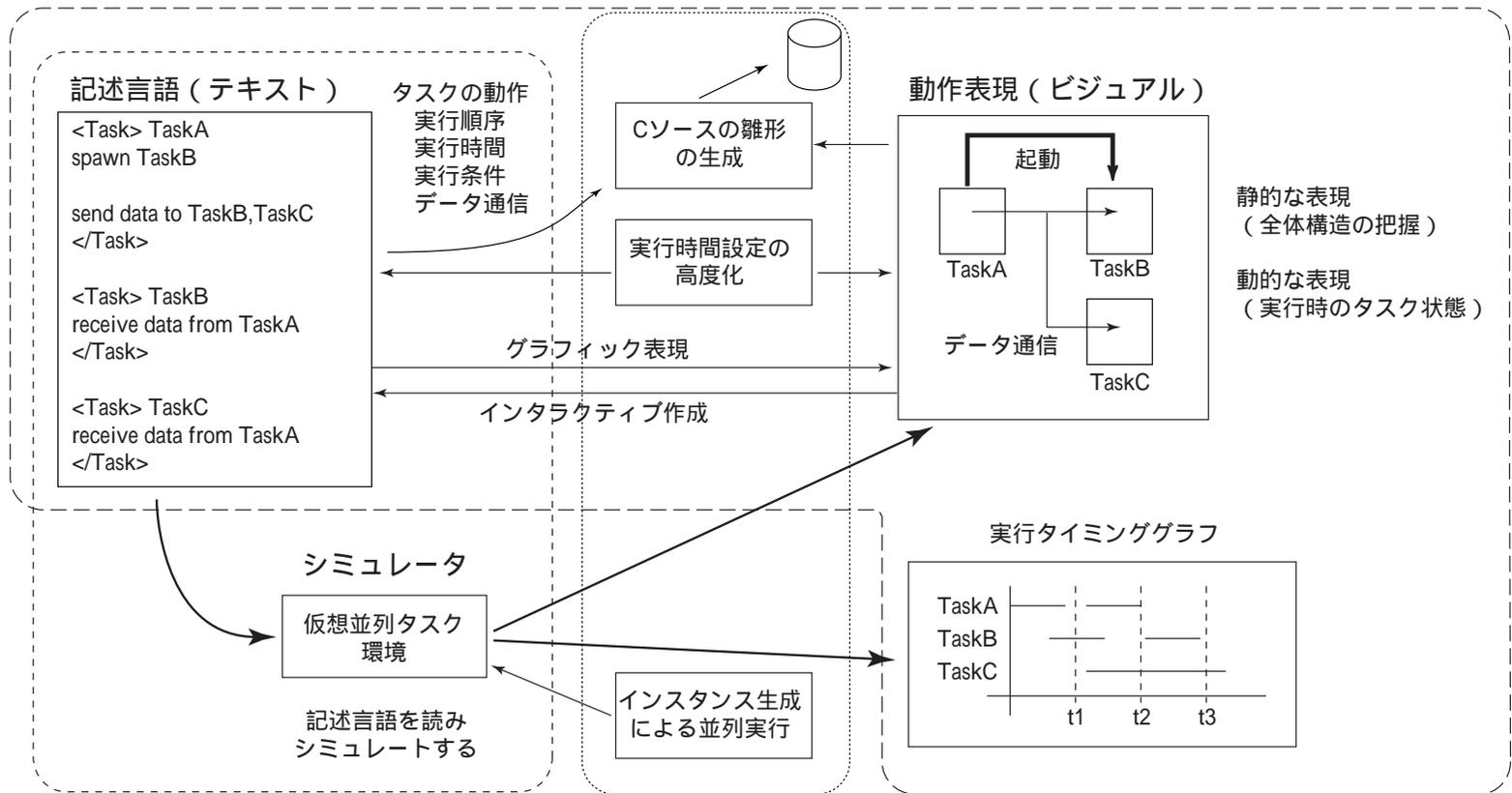


大規模・複雑なシステムの動作を表現した記述が可能となり、シミュレータでタスクの動作を評価することにより、デッドロックの検出、ネックとなるタスクの検出ができるようになる。



大規模・複雑なシステムの解析、  
制御、設計に有効な手法。

# タスク並列処理言語の開発

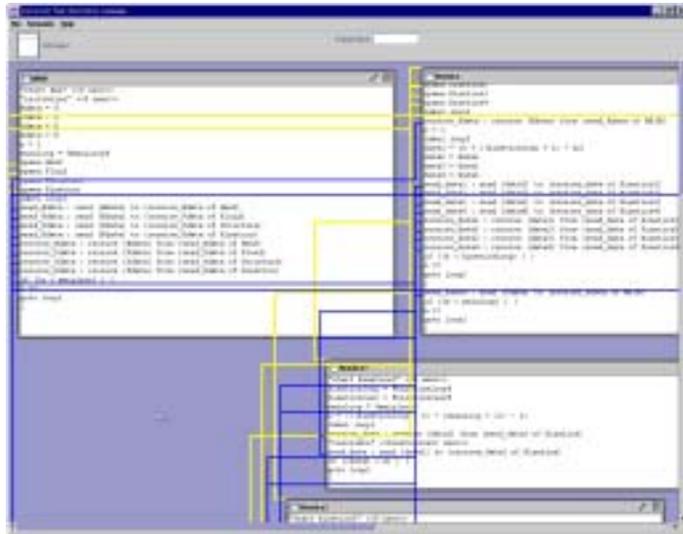


# タスク並列処理 記述言語

- タスクの起動(spawn)
- 通信(send, receive)
- 同期(sync)
- 条件分岐(if)
- 変数の定義、設定(a=10)
- 処理時間(<< n >>)
- デバッグ(pause, print)
- 説明文(コメント)

# タスク並列処理シミュレータ

- タスク並列処理シミュレータ



タスク記述言語による  
タスクの記述

シミュレータ  
仮想並列タスク環境

## 動作表現

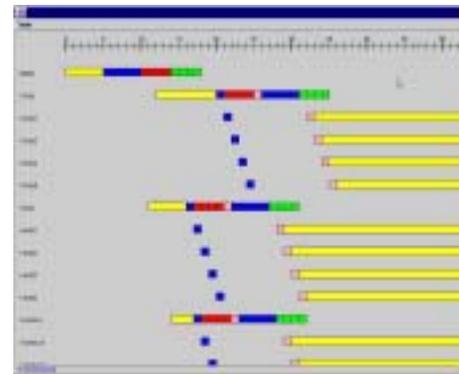
静的な表現 (全体構造の表示)

起動、データ通信の表示

動的な表現 (実行時のタスク状態)

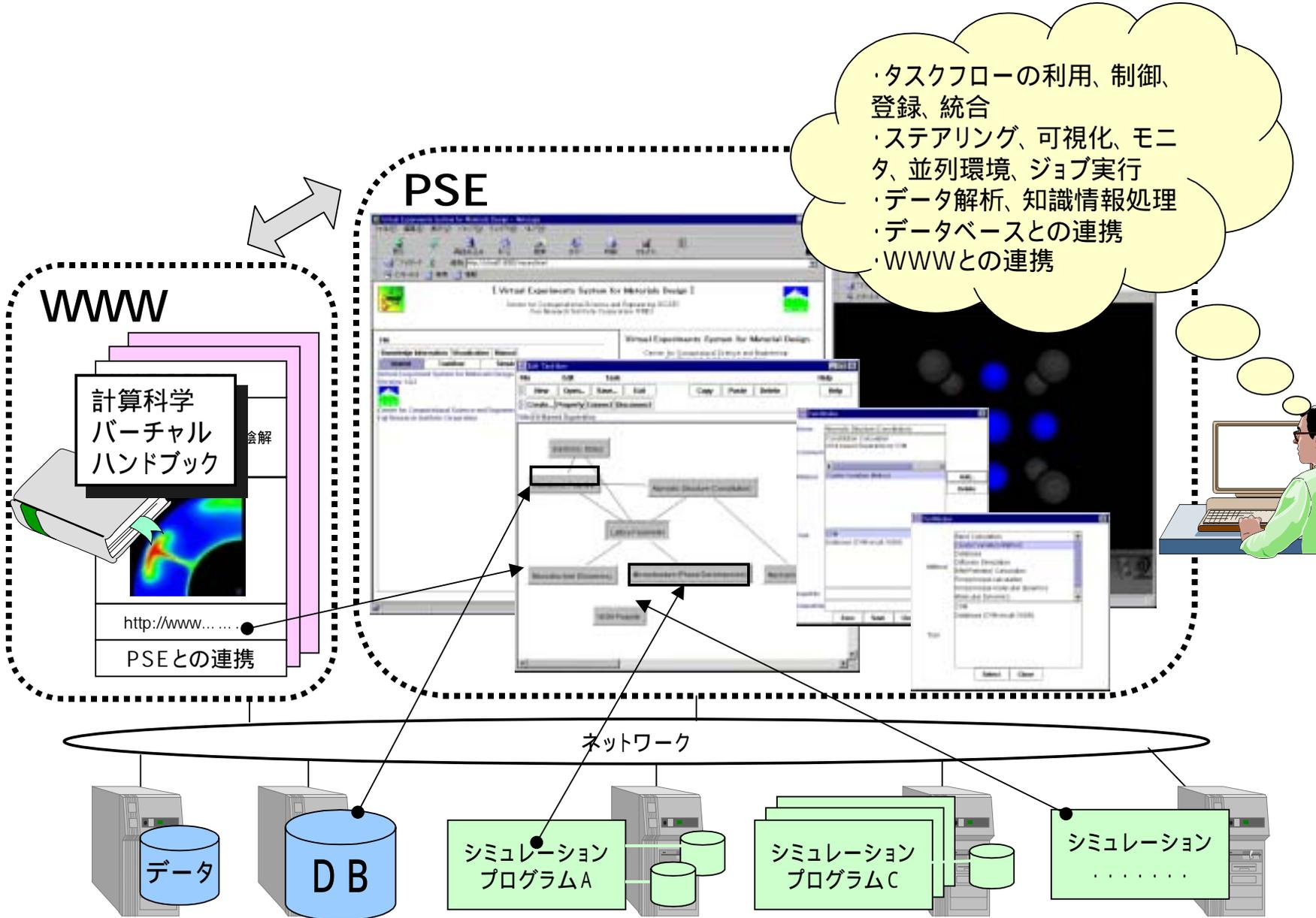
実行時のデータ、制御の動きを表示

## タスクのタイムチャート



## 個々のタスクの実行状況





# 統合化の事例

## 産業機械用 仮想設計システム

- ・流体－構造の連成解析が中心
- ・ポンプを対象

## 量子化学計算統合システム