

Title	戦略的プロジェクトデザインに関する研究
Sub Title	Study of strategic project design
Author	大嶺, 徳和(Omine, Tokukazu) 狼, 嘉彰(Okami, Yoshiaki)
Publisher	慶應義塾大学大学院システムデザイン・マネジメント研究科
Publication year	2010
Jtitle	
JaLC DOI	
Abstract	<p>近年の産業製品は、大規模化、複雑化及び多機能化が進んでおり、システム・オブ・システムズと呼ばれるような巨大なシステムが出現している。それに伴い、巨大なプロジェクトが増加するとともに、複数のプロジェクトの相互関連性が増大する傾向にある。このような事業を円滑に進めるため、個別のプロジェクトをマネジメントするだけでなく、ポートフォリオやプログラムという概念でプロジェクトをグループ化しマネジメントする手法が開発されている。</p> <p>これらの手法によると、マネジメント・プロセスの最も上位にあるのはビジョンであり、ビジョンに基づき具体的な戦略目標が立てられる。プロジェクトはポートフォリオやプログラムといった、プロジェクトのグループとしてマネジメントされるが、最終的には戦略目標を達成することが重要となる。しかしながら、巨大なプロジェクトが増加するとともに複数のプロジェクトの相互関連性が増大するに従い、ポートフォリオやプログラムのマネジメントは複雑性を増し、プロジェクトグループ全体の整合性を確保することが困難となってきた。そのため、戦略目標とプロジェクトの内容に乖離が生じたり、プロジェクトとプロジェクトとの間で不整合が生じたりするなど、個別のプロジェクトを成功させても戦略目標を達成できないといった問題が発生している。</p> <p>従来のマネジメント手法では、膨大なプロジェクトの整合性を取るため、マネジメントの階層構造を構成し、それぞれの階層において様々な手法を駆使することが必要であり、複雑なマネジメントが要求される。本研究では、これらの問題を解決するため、階層構造を撤廃し全プロジェクトをプロジェクト群ととらえ、統合したひとつの手法でプロジェクトの整合性を確保する方法を提案する。プロジェクト群を統合したひとつの手法でマネジメントするためには、膨大な情報への対応や統一した評価方法が必要となるが、プロジェクトに関する情報の定量化、コンピュータ・シミュレーション及びプロジェクト群の可視化といった手法を活用することで問題を解決する。</p> <p>本手法を適用することにより、複雑化した膨大なプロジェクトを抱える組織において、戦略的な視点からプロジェクトをデザインすることが可能となる。</p>
Notes	修士学位論文. 2010年度システムデザイン・マネジメント学 第41号
Genre	Thesis or Dissertation
URL	<a href="https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002010-0017">https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002010-0017</a>

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the Keio Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

修士論文

2010年度

# 戦略的プロジェクトデザイン に関する研究

大嶺 徳和  
(学籍番号：80933072)

指導教員 狼 嘉彰

2011年3月

慶應義塾大学大学院システムデザイン・マネジメント研究科  
システムデザイン・マネジメント専攻

# 論文要旨

学籍番号	80933072	氏名	大嶺 徳和
論文題目： 戦略的プロジェクトデザインに関する研究			
<p>(内容の要旨)</p> <p>近年の産業製品は、大規模化、複雑化及び多機能化が進んでおり、システム・オブ・システムズと呼ばれるような巨大なシステムが出現している。それに伴い、巨大なプロジェクトが増加するとともに、複数のプロジェクトの相互関連性が增大する傾向にある。このような事業を円滑に進めるため、個別のプロジェクトをマネジメントするだけでなく、ポートフォリオやプログラムという概念でプロジェクトをグループ化しマネジメントする手法が開発されている。</p> <p>これらの手法によると、マネジメント・プロセスの最も上位にあるのはビジョンであり、ビジョンに基づき具体的な戦略目標が立てられる。プロジェクトはポートフォリオやプログラムといった、プロジェクトのグループとしてマネジメントされるが、最終的には戦略目標を達成することが重要となる。しかしながら、巨大なプロジェクトが増加するとともに複数のプロジェクトの相互関連性が增大するに従い、ポートフォリオやプログラムのマネジメントは複雑性を増し、プロジェクトグループ全体の整合性を確保することが困難となってきた。そのため、戦略目標とプロジェクトの内容に乖離が生じたり、プロジェクトとプロジェクトとの間で不整合が生じたりするなど、個別のプロジェクトを成功させても戦略目標を達成できないといった問題が発生している。</p> <p>従来のマネジメント手法では、膨大なプロジェクトの整合性を取るため、マネジメントの階層構造を構成し、それぞれの階層において様々な手法を駆使することが必要であり、複雑なマネジメントが要求される。本研究では、これらの問題を解決するため、階層構造を撤廃し全プロジェクトをプロジェクト群ととらえ、統合したひとつの手法でプロジェクトの整合性を確保する方法を提案する。プロジェクト群を統合したひとつの手法でマネジメントするためには、膨大な情報への対応や統一した評価方法が必要となるが、プロジェクトに関する情報の定量化、コンピュータ・シミュレーション及びプロジェクト群の可視化といった手法を活用することで問題を解決する。</p> <p>本手法を適用することにより、複雑化した膨大なプロジェクトを抱える組織において、戦略的な視点からプロジェクトをデザインすることが可能となる。</p>			
キーワード (5語) 戦略的プロジェクトデザイン、プロジェクト・マネジメント、定量化、可視化、コンピュータ・シミュレーション			

## SUMMARY OF MASTER'S DISSERTATION

Student Identification Number	80933072	Name	Tokukazu OHMINE
Title Study of Strategic Project Design			
Abstract <p>Recently, industrial products show tendency to become larger in scale, and more complicated with multiple functions. Even a single project increases its complexity. Therefore, the interrelation among the projects shows the tendency to increase and to complicate. The project is managed as well as, Various techniques such as the portfolio management and program management have been developed in addition to management of individual projects.</p> <p>According to the portfolio management is highest in the rank. The strategic target is set from this vision. Therefore, it is important to management to achieve strategic target. However, the increased interrelation among the projects, makes it difficult to ensure the adjustment of the project group. As a result, it becomes difficult to achieve the strategic target for a successful project, producing inconsistencies between projects. The problem arises from the fact that the conventional method is based on various techniques and hierarchical management.</p> <p>In this study, quantification and visualization of a group of projects have been examined from a strategic viewpoint. It has turned out effective that the project is designed from a strategic viewpoint to solve these problems. It is necessary to manage huge amount of information on the project group to achieve it. It is extremely important to quantify information on the project group in applying methods of solving this problem. To demonstrate effectiveness of this approach, a computer simulation program has been developed.</p> <p>Application of this method to a large organization with complex projects has shown a potential to design a project from a strategic viewpoint.</p>			
Key Words: Strategic Project Design, Project management, Quantify information, Visualization, Computer simulation			

# 目次

<b>1 緒言</b> .....	<b>9</b>
1.1 研究の背景.....	9
1.2 研究の目的.....	10
1.3 研究の進め方.....	11
1.3.1 システムデザインのVモデル.....	11
1.3.2 本研究のVモデル.....	12
<b>2 ユーザー要求</b> .....	<b>13</b>
2.1 プロジェクトを取り巻く環境.....	13
2.1.1 プロジェクトの相互関連性が増加.....	13
2.1.2 ユーザーの厳しい要求とリスクの増加.....	14
2.2 先行研究.....	16
2.3 先行研究の問題点.....	18
2.3.1 プロジェクトで発生している問題.....	18
2.3.2 従来のマネジメント手法の問題点.....	19
2.4 ユーザー要求の設定.....	23
2.5 トレーサビリティ.....	24
2.6 現状とあるべき姿のイメージ.....	25
<b>3 要求分析</b> .....	<b>26</b>
3.1 システムの境界の明確化.....	26
3.1.1 ライフサイクル.....	26
3.1.2 コンテキスト図.....	27
3.1.3 ユースケース図.....	28
3.2 システム自体の明確化.....	31
3.3 システム仕様.....	32
3.4 トレーサビリティ.....	34

<b>4</b>	<b>アーキテクティング</b>	<b>35</b>
4.1	アーキテクティングの方針	35
4.1.1	プロジェクト群	35
4.1.2	定量化	37
4.1.3	可視化	39
4.1.4	汎用化	39
4.2	機能設計	39
4.3	物理設計	41
4.4	インターフェイス設計とアーキテクチャ設計結果	42
4.5	トレーサビリティ	43
<b>5</b>	<b>システムデザイン</b>	<b>45</b>
5.1	データベースソフト	45
5.1.1	既存のソフトウェアの選択	45
5.1.2	データ入力形式	45
5.2	データサーバー	45
5.3	SPD ツール	46
5.3.1	開発言語	46
5.3.2	データファイル読み込み	46
5.3.3	リスクケース別プロジェクトデータ算出	48
5.3.4	プロジェクト群データ算出	49
5.3.5	戦略目標と制約条件の整合性評価	49
5.3.6	可視化手法の選択	50
5.4	トレーサビリティ	52
<b>6</b>	<b>インテグレーション</b>	<b>53</b>
6.1	インテグレーション計画	53
6.2	インテグレーション結果	54
6.3	ユーザーズ・マニュアル	56

<b>7 Verification&amp;Validation.....</b>	<b>58</b>
7.1 Verification&Validation の方針.....	58
7.2 Verification(システムデザイン).....	58
7.3 Verification(アーキテクティング).....	59
7.4 Validation.....	61
7.4.1 Validation の方針.....	61
7.4.2 プロジェクトデータ.....	61
7.4.3 Validation 条件の設定.....	63
7.4.4 シナリオの設定.....	64
7.4.5 Validation.....	64
7.4.6 Validation 結果.....	68
<b>8 結論.....</b>	<b>69</b>
<b>9 参考文献.....</b>	<b>70</b>
<b>10 謝辞.....</b>	<b>71</b>
<b>11 付録.....</b>	<b>72</b>
11.1 ユーザーズ・マニュアル.....	72
11.1.1 データベースソフト.....	72
11.1.2 SPD ツール.....	76
11.2 SPD ツールのプログラム.....	78

## 略語(1/2)

略語	正式語	意味
AHP	Analytic Hierarchy Process	意思決定における問題の分析において、人間の主観的判断とシステムアプローチとの両面からこれを決定する問題解決型の意思決定手法
CBS	Cost Breakdown Structure	プロジェクトの費用を詳細に区分して階層構造で表現したコスト構成図
CPM	Critical Path Method	工程計画・管理手法のひとつで、各作業工程をネットワーク図に表してクリティカルパスを分析し、所定期間内に計画が完了する最適解を求める手法。
CSV	Comma-Separated Values	いくつかの項目をカンマで区切ったテキストデータおよびテキストファイル
CVCA	Customer Value Chain Analysis	どんな価値がどこで誰に提供されるのかを分析するため、利害関係者のお金、サービス及び情報などの流れをネットワーク状に図示する手法
DSM	Design Structure Matrix	設計や製品開発をどのような工程で行えばよいか分析するため、工程間の関係をマトリックスで図示する手法
FFBD	Function Flow Block Diagram	システムの動作の流れをフロー図で記述したものであり、階層化による細分化を行いながら設計する手法
FMEA	Failure Mode Effects Analysis	設計の不完全や潜在的な欠点を見出すために構成要素の故障モードとその上位アイテムへの影響を解析する方法
FTA	Fault Tree Analysis	下位アイテム又は外部事象等が、定められたフォールトモードを発生させ得るか決めるための、フォールトの木形式で表された解析法
IDEF	Integration DEFinition	アメリカ国立標準技術研究所により米国国家標準となっている、事業活動をモデリングする手法
MECE	Mutually Exclusive and Collectively Exhaustive	各事柄間に重なりがなく、全体として漏れがない状態
NASA	the National Aeronautics and Space Administration	アメリカ航空宇宙局
JAXA	Japan Aerospace eXploration Agency	独立行政法人 宇宙航空研究開発機構



## 略語(2/2)

略語	正式語	意味
QCD	Quality Cost Delivery	品質 (Quality)、コスト(Cost)、納期 (Delivery)のことであり、製造業における3つの柱
QFD	Quality Function Deployment	顧客の声を聞き、定量化とマトリクスを利用して機能を展開していくことで何が重要であるのかを分析する手法
SPD	Strategic Project Design	本研究で提案するツールの名称
TRL	Technology Readiness Level	アメリカ航空宇宙局が開発した技術成熟度を評価するために用いられる指標
WBS	Work Breakdown Structure	プロジェクト管理における作業項目の細分化及び洗い出しを行うための手法

# 1 緒言

## 1.1 研究の背景

近年の産業製品は、大規模化、複雑化及び多機能化が進んでおり、システム・オブ・システムズと呼ばれるような巨大なシステムが出現している。それに伴い、巨大なプロジェクトが増加するとともに、複数のプロジェクトの相互関連性が増大する傾向にある。大規模及び複雑化した産業製品の例を図 1.1 に示す。



H-IIIB ロケット



戦闘機 F-2

図 1.1 大規模及び複雑化した産業製品

このような事業を円滑に進めるため、個別のプロジェクトをマネジメント<sup>(1)</sup>するだけでなく、ポートフォリオやプログラムという概念でプロジェクトをグループ化しマネジメントする手法<sup>(2)(3)(4)</sup>が提案されている。これらの手法によると、マネジメント・プロセスの最も上位にあるのはビジョンであり、ビジョンに基づき具体的な戦略目標が立てられる。プロジェクトはポートフォリオやプログラムといった、プロジェクトのグループとしてマネジメントされるが、最終的には戦略目標を達成することが重要となる。しかしながら、巨大なプロジェクトが増加するとともに複数のプロジェクトの相互関連性が増大するに従い、ポートフォリオやプログラムのマネジメントは複雑性を増し、プロジェクトのグループ全体の整合性を確保することが困難となってきている。そのため、戦略目標とプロジェクトの内容に乖離が生じたり、プロジェクトとプロジェクトとの間で不整合が生じたりと、個別のプロジェクトを成功させても戦略目標を達成できないといった問題が発生している。

一方、プロジェクトを取り巻く環境は厳しさを増しており、高品質、低価格及び短納期を求めるユーザーの声が強まっている。経済産業省の 2010 年版ものづくり白書<sup>(5)</sup>によると、我

が国の製造業各社は、自社の生産技術力を継続的に強化していく意向を示すとともに、「高品位な製造品を製造する製造・生産設備の導入・開発」、「コスト競争力を備えるための有力な調達先の確保」及び「製品開発期間・サイクルの短期化」などを図り、更なる競争力を得ることが必要としている。

しかしながら、ユーザーの要求を満足するようにプロジェクトを計画した結果、リスクの増大を招き、リスクが顕在化することでプロジェクトの計画変更を余儀なくされる場合もある。近年ではプロジェクトの計画変更が頻繁に行われる傾向にある。膨大なプロジェクトを抱える組織にとっては、ひとつのプロジェクトがさまざまなプロジェクトに影響を与える場合が多いため、リスクが顕在化した場合の影響を事前に把握し、迅速に対応することが特に重要となっている。

## 1.2 研究の目的

本研究では、複雑化した膨大なプロジェクトを抱える組織において、戦略目標を達成するプロジェクトのデザイン方法を提案することを目的とする。具体的な目的については、プロジェクトを取り巻く環境と現状の問題点からユーザー要求を設定し、ユーザー要求を満足することと設定する。

この手法を適用することにより、組織が設定した戦略目標を達成できるようにプロジェクトを計画できるようになるとともに、複雑化するプロジェクト間の整合性を保ちプロジェクトの実行可能性を確保することが可能となる。

本研究で提案する手法は、戦略的プロジェクトデザイン法と定義する。

## 1.3 研究の進め方

### 1.3.1 システムデザインのVモデル

本研究は、システムデザインのVモデル<sup>6)</sup>に沿った手順で研究を行う。システムデザインのVモデルとは、対象とするシステムが何であるかに関わらず適用可能な、システムのデザインやマネジメントを行うための共通の方法論である。慶應義塾大学大学院システムデザイン・マネジメント研究科のホームページのシステムデザインのVモデルを図 1.2 に示す。

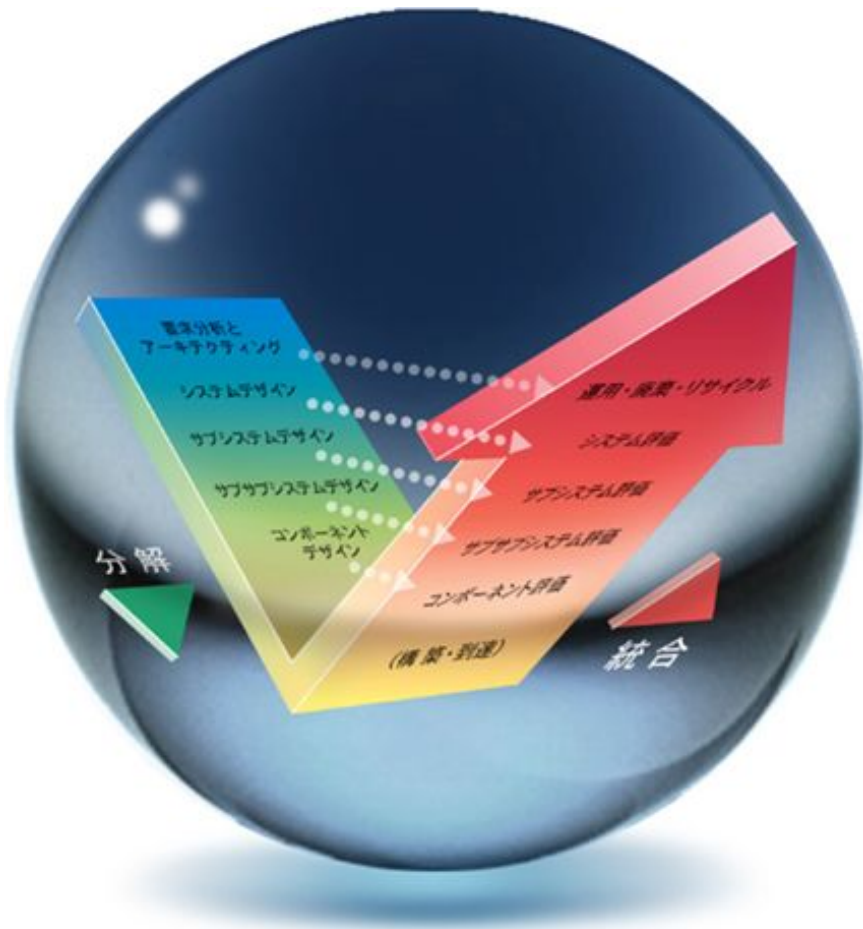


図 1.2 システムデザインのVモデル

Vモデルは、要求を分析し、システムを分解、明確及び確定する前半のフェーズと、システムを構築及び統合を行う後半のフェーズに分かれており、それぞれV字の左側と右側を構成している。Vモデルは水平方向で概念のレベルが同じであり、左側で確定した内容が、右側で構築したシステムに対応するモデルとなっている。Vモデルでは、確定した内容が構築したシステムと合致していることを評価することを重要視しており、Vモデルの上層では、正しいシステムができているかを評価する **Validation**、Vモデルの下層では、正しくシステムができているかを評価する **Verification** を実施する。

### 1.3.2 本研究のVモデル

本研究に適用するVモデルを図 1.3 に示す。

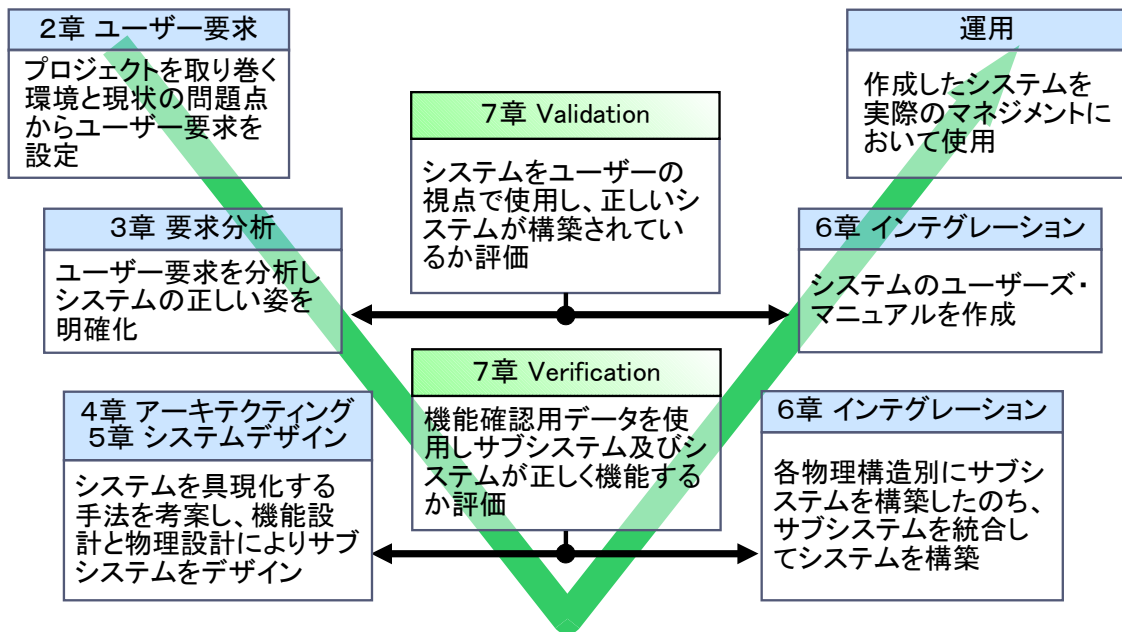


図 1.3 本研究のVモデル

まず始めに、2章において、プロジェクトを取り巻く環境と先行研究の問題点を分析し、本研究でデザインするシステムである「戦略的プロジェクトデザイン法」に対するユーザーの要求を設定する。次に、3章では、ユーザー要求を分析しシステムの正しい姿を明確化してシステム仕様としてまとめる。4章では、システムを具現化するための手法を考案し、機能設計、物理設計及びインターフェイス設計を行いアーキテクチャ設計結果としてまとめる。5章では各サブシステムのデザインを実施する。2章から5章でシステムを分解、明確及び確定するVモデルの前半のフェーズを終了する。6章では、物理構成別にサブシステムを構築し、それを統合して全体のシステムを完成させるとともに、ユーザーズ・マニュアルを提供する。6章がシステムの構築及び統合を行うVモデルの後半のフェーズとなる。そして最後に、7章において、システムが正しくできているか確認する **Verification** と正しいシステムができているかを確認する **Validation** を実施する。**Validation** をもってシステムは完成し、実際のマネジメントにおいて使用可能となるが、本研究ではシステムを構築し **Validation** による検証までを研究の対象とする。

## 2 ユーザー要求

### 2.1 プロジェクトを取り巻く環境

#### 2.1.1 プロジェクトの相互関連性が増加

最近の産業製品は、システム・オブ・システムズと呼ばれる製品が増加しており、大規模複雑化、多機能化が進む傾向にある。そのため、巨大なプロジェクトが増加するとともに、複数のプロジェクトの相互関連性が増加する傾向にある。プロジェクトの相互関連性の例を図 2.1 に示す。

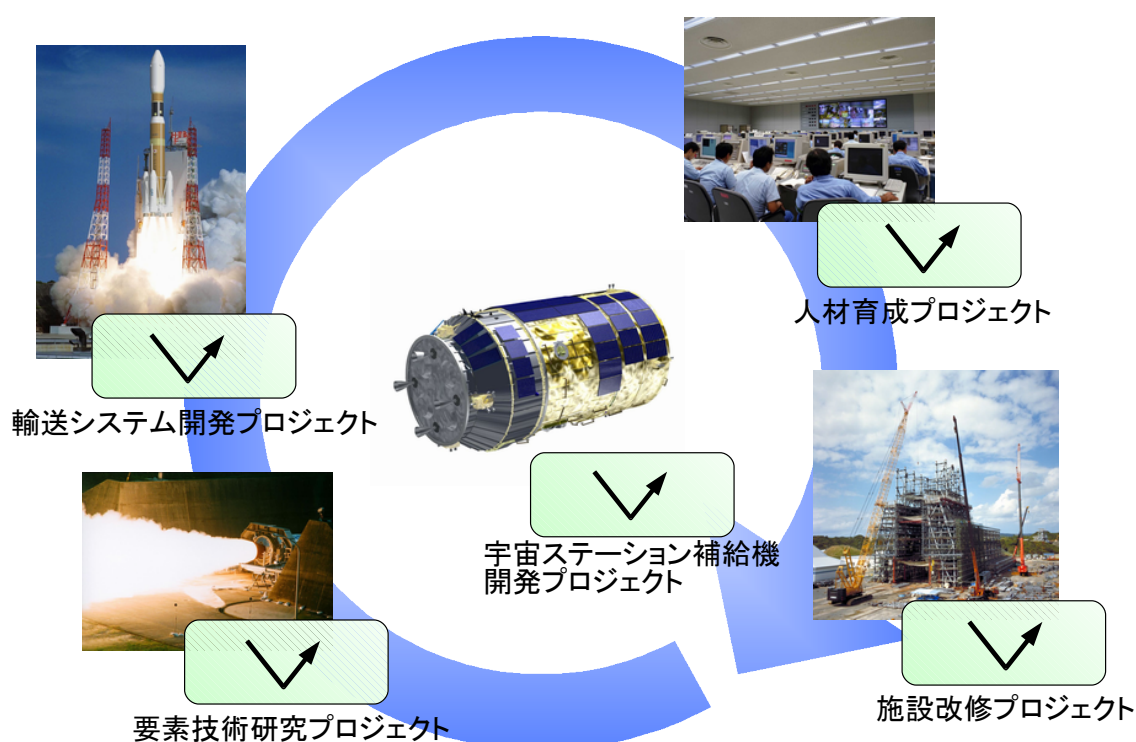


図 2.1 プロジェクトの相互関連性の例

宇宙ステーション補給機開発プロジェクトだけでは、宇宙ステーションに物資を輸送することはできない。宇宙ステーション補給機を宇宙に輸送するための輸送システムが必要である。輸送システムを開発する技術がなければ要素技術を研究するプロジェクトも必要となる。また、宇宙ステーション補給機と直接接続していない部分でも、施設を改修するプロジェクトや人材を育成するプロジェクトなどが必要となる場合もある。

このように、ひとつの戦略目標を達成するためには、様々なプロジェクトが密接に関係しており、これらの整合性をとることが重要となっている。しかしながら、関係するプロジェクトが増加するに従い、プロジェクト間の関係は複雑性を増し整合性を取ることが難しくなる。プロジェクトの相互関連性が増加すると次のような影響がある。

- プロジェクト間の因果関係の把握が難しい。
- プロジェクトが他のプロジェクトに与える影響が大きい。
- 戦略目標が異なるなど、関係がないと考えていたプロジェクトに影響を与える。
- 考慮していなかった将来のプロジェクトに影響を与える。

複数のプロジェクトをプロジェクト群と見た場合、プロジェクト群は始まりと終わりが存在せず持続的に複数のプロジェクトを実施している状態にある。そのため、プロジェクト群を進めながら、常にプロジェクトの変更や組み替えに対応しなくてはならないといった特徴がある。プロジェクトの変更や組み換えは、プロジェクト間の不整合を発生させる可能性があるが、その不整合は非常に長期的な問題となり大きな影響を与え続ける場合がある。これらの問題は、プロジェクトでしばしば行われるような対処療法的な対応では解決できない問題も多い。そのため、プロジェクト群を戦略の視点から把握し問題に対処することは非常に重要な課題であるといえる。

## 2.1.2 ユーザーの厳しい要求とリスクの増加

製造業は、品質と経費と納期の3つの柱で成り立っているとされる。品質をQ(Quality)、経費をC(Cost)、納期をD(Delivery)とおき3つの柱をQCDと表現する。QCDは互いに独立しておらず、それぞれトレードオフの関係となっている。そのため、QCDは鉄の三角形(Iron Triangle)<sup>7)</sup>とも呼ばれている。産業製品が、高品質、低価格及び短納期であることが良いことは言うまでもないが、高品質を追求すると価格が高くなったり納期が延びる、低価格を追求すると品質が低下したり納期が延びる、納期を短くすると品質が低下したり価格が高くなるといった事象が発生する。しかしながら、産業製品に対して、高品質、低価格及び短納期を求めるユーザーの声は強まっており、限界を超えて高品質、低価格及び短納期を追求することを余儀なくされる場合もある。そのため、ユーザーの要求を受け入れてプロジェクトに着手しても、計画通りにプロジェクトが進行しないとといった問題が発生することとなる。これは、あるべき鉄の三角形に対してより大きな成果を求めた結果、リスクが増大しそのリスクが顕在化した結果であると考えられることができる。正常な鉄の三角形を図2.2aに、無理な要求に対応した鉄の三角形のイメージを図2.2bに示す。

リスクが顕在化しプロジェクトの計画変更が行われる事例は増加する傾向にあり、プロジェクトの計画変更に対応することの重要性が高まっている。特に、膨大なプロジェクトを抱える組織にとっては、ひとつのプロジェクトがさまざまなプロジェクトに影響をあたえる可能性があり、リスクが顕在化した場合の影響を事前に把握し、迅速に対応することが難しく大きな課題となっている。



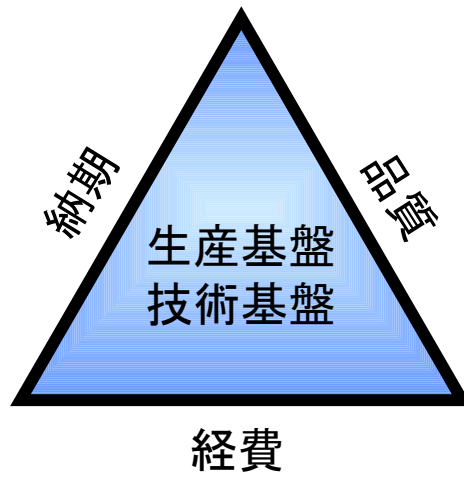


図 2.2a 正常な鉄の三角形

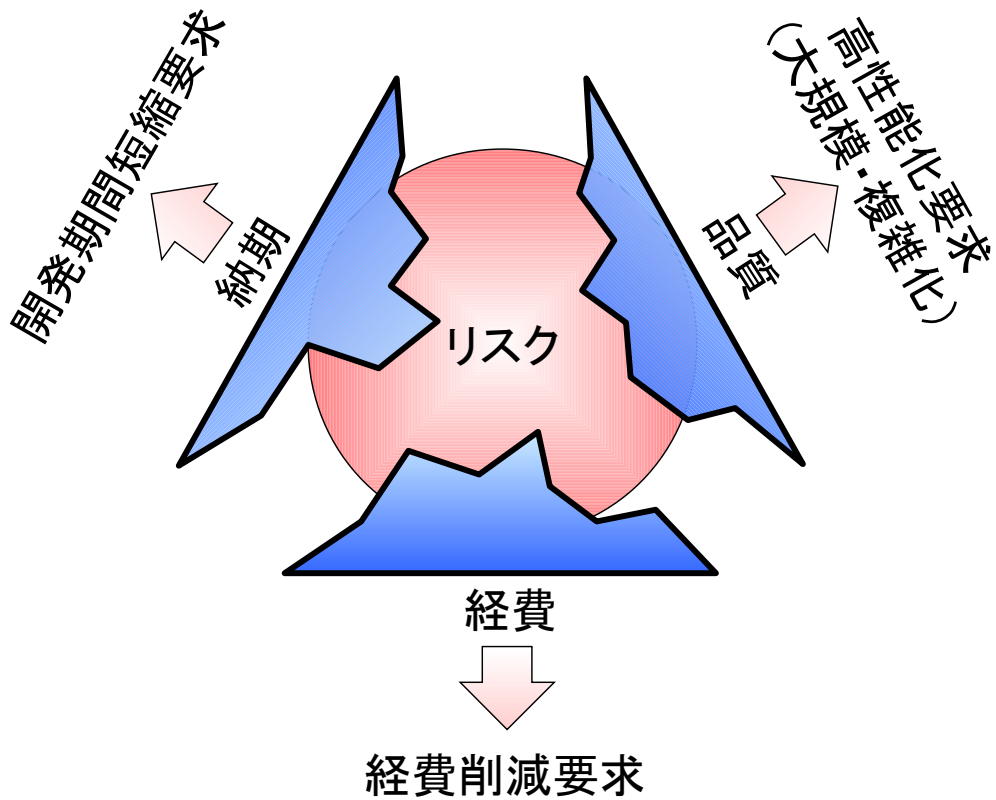


図 2.2b 無理な要求に対応した鉄の三角形のイメージ



## 2.2 先行研究

プロジェクトとは独自の製品やサービス等を創出するために実施される有機的な業務であり、マネジメントを行う基本的なコンポーネントである。プロジェクトをグループ化しマネジメントする手法にプロジェクト・ポートフォリオ・マネジメントとプログラム・マネジメントの手法<sup>(2)(3)(4)</sup>がある。

組織において最も上位にあるのがビジョンであり、具体的な目標として戦略目標が設定される。ポートフォリオとは、戦略的なビジネス目標を達成するため、仕事を効果的にマネジメントすることを目的にまとめられたコンポーネントであり、下位のポートフォリオ、プログラム、プロジェクト及び他の活動のコンポーネントで構成されている。ポートフォリオ・マネジメントを実施することにより、ポートフォリオ内のプログラムやプロジェクトなどの整合をとることが可能となる。プログラムとは、プロジェクトを個々にマネジメントすることでは、得られないベネフィットとコントロールを実現するため、相互に関連するプロジェクトをグループ化したものであり、プログラム・マネジメントを実施することにより、プログラム内のプロジェクトの整合を取ることが可能となる。戦略目標、ポートフォリオ、プログラム及びプロジェクトの関係は図 2.3 のとおり。

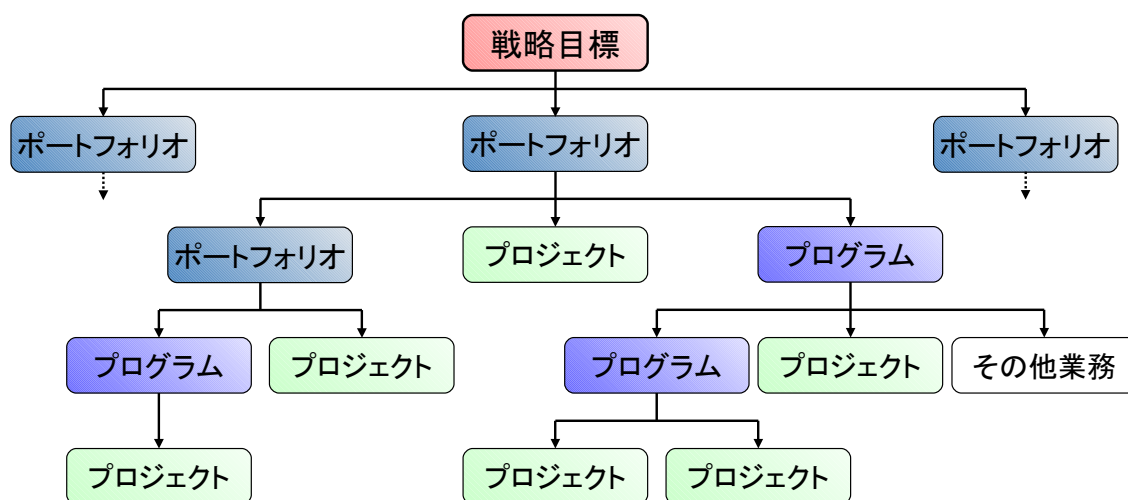


図 2.3 戦略目標とコンポーネントの階層図<sup>(3)</sup>

戦略目標を達成するためには、ポートフォリオ、プログラム及びプロジェクト等の各コンポーネントが戦略目標と整合することが重要であり、それぞれのコンポーネントの整合性が確保され実行可能であることが必要である。プロジェクト・ポートフォリオ・マネジメントでは、戦略目標と整合したコンポーネントを選択し、コンポーネントの整合性を確保するため、プロジェクト・ポートフォリオ・マネジメント・プロセスが行われる。プロジェクト・ポートフォリオ・マネジメント・プロセスのハイレベル図を図 2.4 に示す。

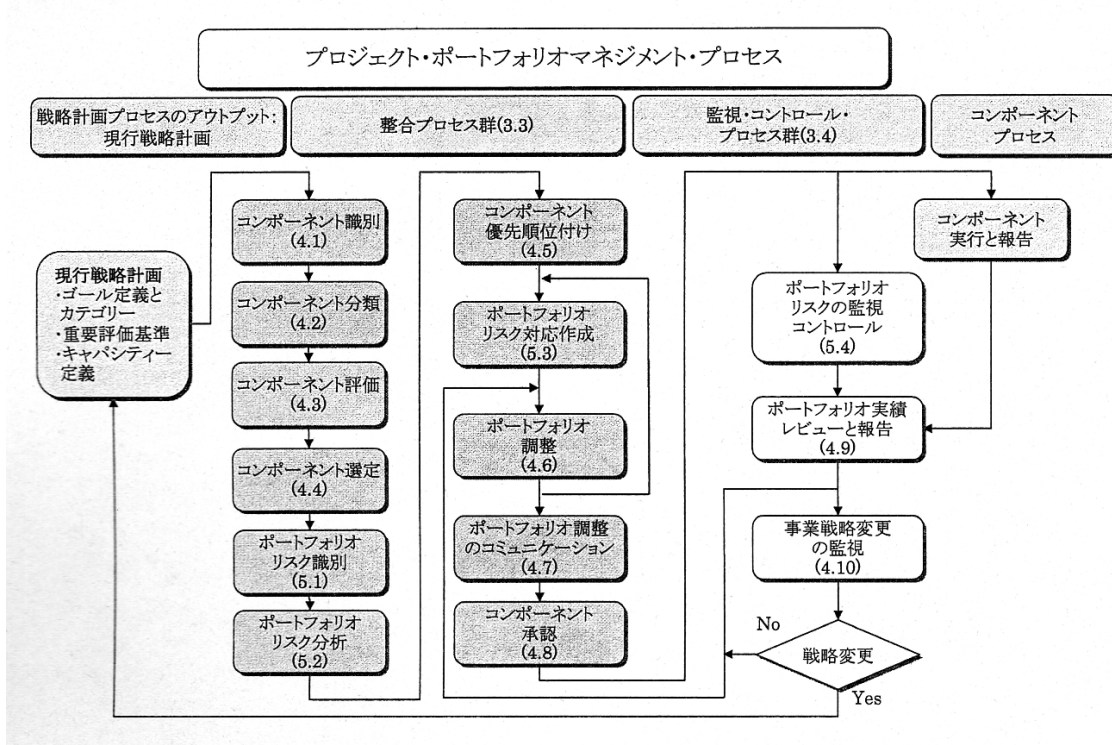


図 2.4 プロジェクト・ポートフォリオ・マネジメント・プロセスのハイレベル図<sup>(3)</sup>

プロジェクト・ポートフォリオ・マネジメント・プロセスでは、現行戦略計画から始まり識別、分類及び評価等が行われ、ポートフォリオ調整でコンポーネントの選択及び調整が行われる。このプロセスでは、このポートフォリオ調整を実施することにより、戦略目標とコンポーネントの整合性やコンポーネント間の整合性を取ることが可能となっている。コンポーネントが実行され、状況や結果が報告されると、必要に応じて再度ポートフォリオ調整が行われる。これにより、整合性を確保し続けることが可能な、循環型のプロセス構造となっている。

## 2.3 先行研究の問題点

### 2.3.1 プロジェクトで発生している問題

プロジェクト・ポートフォリオ・マネジメント、プログラム・マネジメント及びプロジェクト・マネジメントを行うことにより、それぞれのコンポーネントは戦略目標と整合がとれており、実行可能性も確保されるはずである。しかしながら、戦略視点からプロジェクトの状況を見ると、様々な不整合が発生している場合がある。不整合の例を次に示す。

- ポートフォリオやプログラム内の整合性は確保できていたが、他のポートフォリオやプログラムに属するプロジェクトと不整合や重複が発生した。
- 複数のポートフォリオやプログラムに関連するプロジェクトがあり、特定のポートフォリオやプログラムの要求を満足する計画を立てるのが難しい。
- あるプロジェクトで採用したシステムや方式などが、それを活用しようとした別のプロジェクトで適合せず問題となった。
- ある戦略項目を達成することを優先した結果、別の戦略項目や制約条件との不整合を誘発した。
- プロジェクトの計画変更などの情報が、他のプロジェクト・マネージャに伝達されるのに時間がかかり、計画変更に伴うプロジェクトの不整合を誘発するのを防止できなかった。

上記の様な問題が、プロジェクト間で発生しているイメージを図 2.5 に示す。

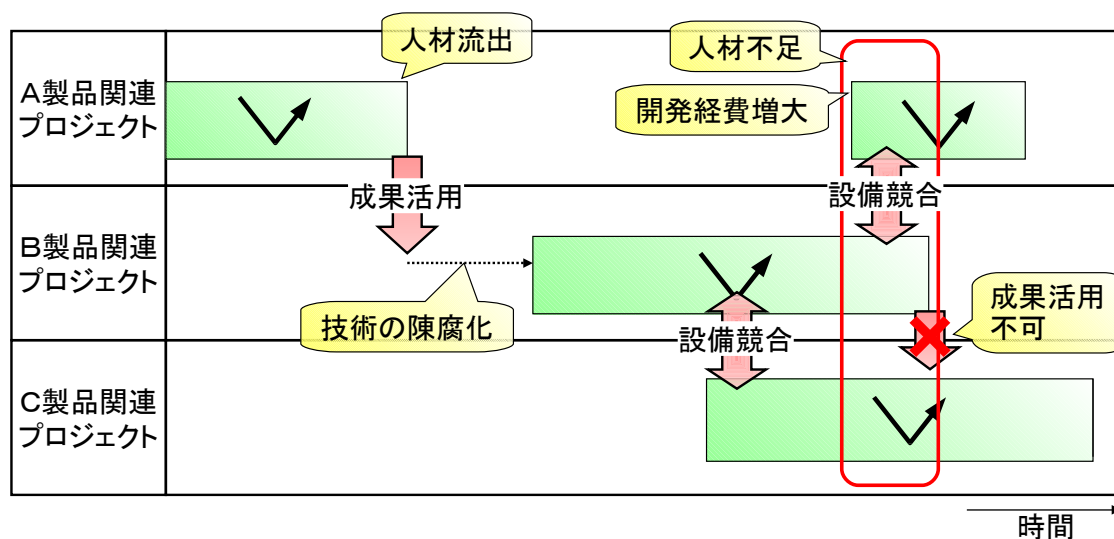


図 2.5 プロジェクト間で不整合が発生している例

図 2.5 の各プロジェクトは、別のポートフォリオでマネジメントされているプロジェクトである。これらのプロジェクトは別の戦略目標のために活動を行っているが、経費、人的資源、技術及び設備などは共有の資源を使用しているため、不整合が発生する場合がある。

## 2.3.2 従来のマネジメント手法の問題点

### 2.3.2.1 階層的なマネジメントによる情報の簡略化

プロジェクト・ポートフォリオ・マネジメントでは、ポートフォリオ調整のプロセスにより、プロジェクトなどのコンポーネントの調整が行われる。プロジェクトは、スコープ、タイム、コスト、品質、人的資源及びリスク等、様々なマネジメントを行う必要があるが、これらに関する情報は非常に膨大であるため、プロジェクトからプログラム、プログラムからポートフォリオと、情報がマネジメントの上層に伝達するのに従い情報が集約し簡略化されていくのが一般的である。特に、複数のポートフォリオを抱える組織では、プロジェクトの数が膨大となるため、その傾向が顕著となる。複数のポートフォリオを保有している場合のプロジェクトの状況を図 2.6a に、プロジェクトの情報が上層に伝達するのに従い簡略化される状況を図 2.6b に示す。

通常、別々のポートフォリオで管理されているプロジェクトは事業の内容としては関係がないのが前提である。しかしながら、経費、人的資源、技術及び設備などは資源を共有しているため、プロジェクトの新規着手、計画変更及び中止などにより、共有資源に影響を与えると、異なるポートフォリオの属するプロジェクトであっても影響を受ける可能性がある。そのため、プロジェクトの情報は他のプロジェクト・マネージャにも伝達されなくてはならない。しかしながら、プロジェクトの情報が上層に伝達するに従い簡略化されると、プロジェクトをマネジメントするのに必要な情報が伝達されない場合がある。階層的なマネジメント構造は、効率よくプロジェクトを管理することを可能とするが、反面、ポートフォリオが異なるプロジェクト間の情報伝達を阻害する要因にもなっている。産業製品が大規模・複雑化している近年においては、プロジェクト間の関連性が増大及び複雑化している傾向にあり、情報伝達の阻害は大きな問題となる。

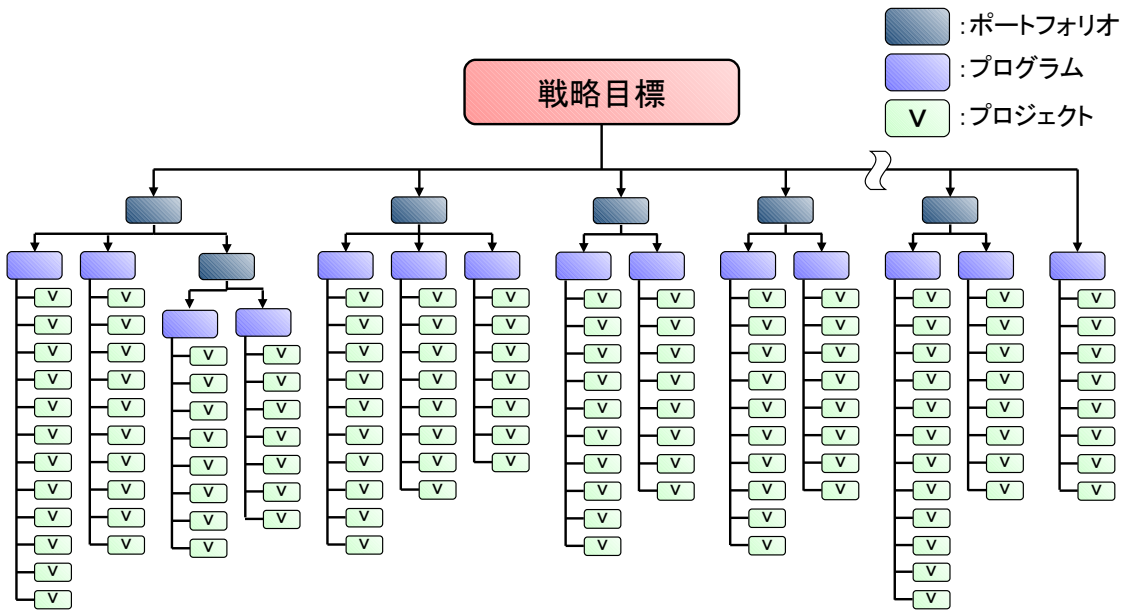


図 2.6a 複数のポートフォリオがある場合のプロジェクトの状況

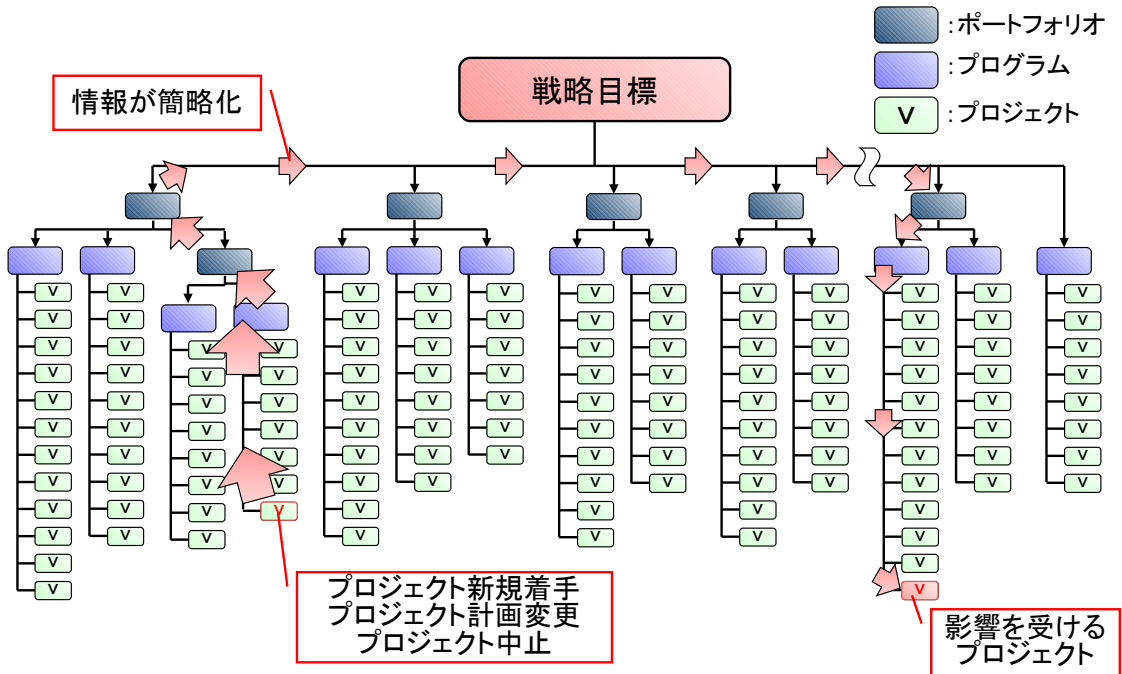


図 2.6b プロジェクトの情報が簡略化されて伝達される状況

### 2.3.2.2 複数のポートフォリオに属するプロジェクトの存在

複数のポートフォリオやプログラムに属するプロジェクトは、MECE(Mutually Exclusive and Collectively Exhaustive)な階層構造を前提としている従来のマネジメント手法が適用できない。

近年の産業製品は多機能化が進んでおり、ひとつの産業製品を製造することで、複数の戦略目標を達成することを要求される場合がある。複数の戦略目標が、それぞれ別のポートフォリオで実現するよう計画してある場合、この産業製品を開発・製造するプロジェクトをどのように評価し管理するのか、ポートフォリオ間で混乱を生じる可能性がある。複数のポートフォリオに属するプロジェクトを図 2.7 に示す。

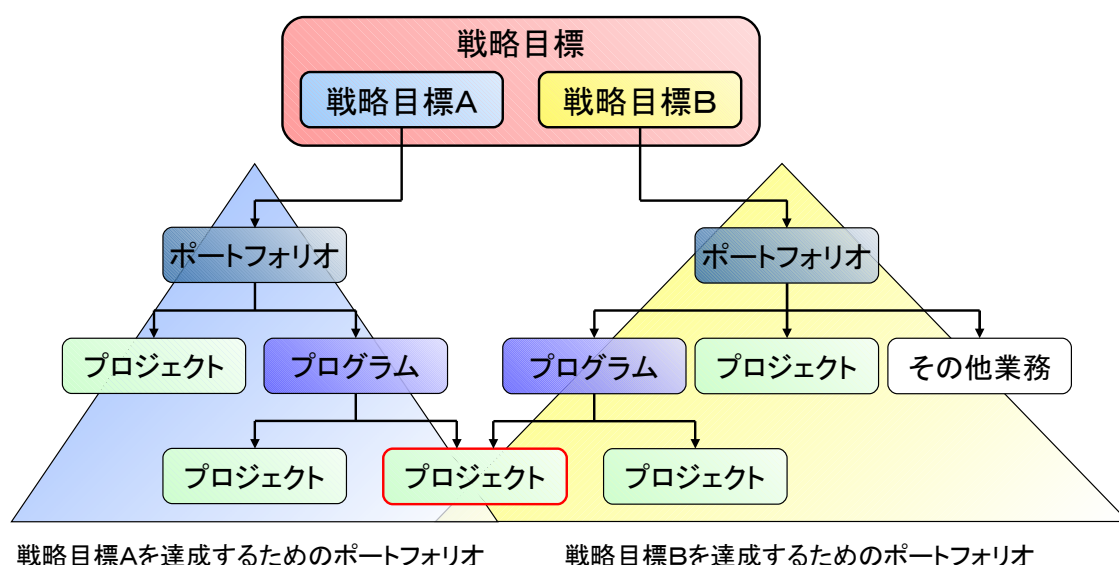


図 2.7 複数のポートフォリオに属するプロジェクトの例

戦略目標との整合性は、ポートフォリオ単位で評価される。そのため、複数のポートフォリオに属するプロジェクトは、ポートフォリオ単位で評価することが難しい。整合性を保つためにはポートフォリオ間での調整が必要となるが、複数のポートフォリオにまたがるプロジェクトが増加すると、その調整は極めて困難となる。複数のポートフォリオにまたがるプロジェクトについては、ポートフォリオの枠組みではなく、戦略目標と合致しているか直接評価をする枠組みが必要となる。

### 2.3.2.3 様々なツールと技法を使用することの必要性

プロジェクト・ポートフォリオ・マネジメントでは、コンポーネントの選択や調整を行うためにコンポーネント調整のプロセスが行われる。これにより、戦略目標との整合の最適化が図られ、組織がポートフォリオを効果的に選定し実行することが可能となる。また、それぞれのコンポーネント間の整合がとられ、実行可能性も確保される。

コンポーネント調整では、戦略目標に掲げている項目だけでなく、利用可能な資源など制約条件も考慮しなくてはならないため、プロジェクトやプロジェクト間の調整で考慮しなくてはならない多くの項目について分析が必要となる。これらの項目について分析するため、プロジェクト・ポートフォリオ・マネジメントでは、費用対効果分析、定量分析、シナリオ分析、確率分析及びグラフ分析など、様々なツールと技法が用意されている。しかしながら、それぞれの項目はかならずしも独立しているわけではなく、相関関係がある項目もあるため、単一のツールと技法だけでは整合性がとれた解を見つけることができない。相関関係のある例を次に示す。

- 費用対効果を最大にしたら、リスクが増大した。
- リスクを最小にしたら、使用する試験設備の競合が発生した。
- 試験設備の競合を回避したら、設計の技術者が不足した。
- 設計技術者を確保したら、費用対効果が悪化した。

このような問題を回避するためには、様々なツールと技法を駆使することが要求される。例えば、プロジェクトの計画が変更になった場合、次のようなツールや技法を駆使することが必要である。

- ヒストグラム、棒グラフ、円グラフ又はバブル・ダイアグラム<sup>(4)(8)</sup>などのチャートを使用し戦略目標と整合しているか分析
- マイルストーン・チャート<sup>(8)</sup>やクリティカル・パス法(CPM: Critical Path Method)<sup>(1)(8)</sup>などを使用し、スケジュールを調整、
- コスト・ベースライン<sup>(1)(8)</sup>などを使用しコストを分析
- モンテカルロ分析<sup>(8)</sup>やデシジョン・ツリー<sup>(8)</sup>などを使用しリスクを分析

この他にも、人的資源、材料、設備及び器材などプロジェクト間で競合する事項については、すべて調整を行い整合性を確保する必要がある。このように、コンポーネント調整ではパズルを組み立てるような作業となり、どのような組み合わせであれば整合がとれるのか分析するのが難しいといった問題がある。

### 2.3.2.4 階層構造に起因する情報伝達のタイムラグ

プロジェクト・ポートフォリオ・マネジメントでは、実行されたコンポーネントは実行した結果をコンポーネント・データとして、ポートフォリオ実績のレビューと報告に入力される。ここで、コンポーネントはレビューされ、必要があればポートフォリオ調整や戦略の変更が行われる。このように、コンポーネントで計画の変更が行われれば、それはコンポーネント・データとして現れ、レビューを行うことで、ポートフォリオ調整や戦略の変更により戦略目標との整合がとれることとなる。このように、ポートフォリオは特定の一時点における選択されたコンポーネントの概観を示し、組織の戦略目標を反映したものとなる。

しかしながら、日々状況が変化するプロジェクトのようなコンポーネントの計画変更と、プロジェクト・ポートフォリオ・マネジメントで行われるポートフォリオ調整や戦略の変更では、通常、実施される頻度が異なっている。例えば、プロジェクトについては週間でスケジュールを管理するが、ポートフォリオ調整は年1回しか行われない場合がある。このように、ポートフォリオ調整の実施される頻度が低い場合、コンポーネントの計画変更が発生しても、他のコンポーネントと整合が取れるまでにタイムラグが発生することになり、これがコンポーネント間の不整合を誘発する原因となる。

コンポーネントの計画変更が発生した場合、それは戦略目標と合致しているのか、プロジェクト・ポートフォリオ・マネジメントによって、詳細な分析を行うことは必要である。しかしながら、コンポーネントの計画変更が必要となった場合、それが他のコンポーネントとの不整合を誘発しないか瞬時に識別し、計画変更が妥当であるのかを判断することも必要である。計画変更が妥当でなければ、計画変更の内容を見直す、緊急にポートフォリオ調整を行うなど、対策を立てることが可能となる。

## 2.4 ユーザー要求の設定

2.3項で先行研究の問題点について述べたが、これらの問題点を解決することができれば、戦略目標とプロジェクトの整合性及びプロジェクト間の整合性が確保できることとなる。よって、先行研究の問題点を解決できる手法を考案し、それをシステムとして提供することを本研究の目的とし、本研究で構築するシステムに対する「ユーザー要求」として設定する。設定したユーザー要求は次のとおり。

- プロジェクト間で必要な情報を伝達できること。
- 複数のポートフォリオやプログラムに属するプロジェクトをマネジメントできること。
- 様々なツールや技法を駆使することなく、戦略目標とプロジェクトの整合性やプロジェクト間の整合性を確保できること。
- プロジェクトの計画変更の影響が全てのマネージャが瞬時に判断できること。

この要求はユーザーの視点から見た要求であり、正しいシステムができていないか検証する Validation の評価項目となる。



## 2.5 トレーサビリティ

先行研究の問題点とユーザー要求のトレーサビリティを確保するため、先行研究の問題点とユーザー要求の応表を作成する。先行研究の問題点とユーザー要求の対応表は、表2のとおり。ユーザー要求を達成することができれば、先行研究の問題点は解決することとなる

表2 問題点とユーザー要求の対応

問題点	ユーザー要求
プログラムやポートフォリオでは、プロジェクトの情報が簡略化して扱われるため、プログラム間、ポートフォリオ間では詳細な情報が伝達されない。	プロジェクト間で必要な情報を伝達できること。
複数のポートフォリオやプログラムに属するプロジェクトは、階層的なマネジメント手法が適用できない。	複数のポートフォリオやプログラムに属するプロジェクトをマネジメントできること。
コンポーネント調整では多くの項目を考慮する必要があり、整合をとるためには様々なツールと技法を使用することが必要となるが、項目の中には相関関係のある項目もあるため、どのようなプロジェクトを組み合わせるのが良いのか把握するのが困難である	様々なツールや技法を駆使することなく、戦略目標とプロジェクトの整合性やプロジェクト間の整合性を確保できること。
戦略からプロジェクトは階層構造となっており、計画変更の頻度が異なるため情報伝達にタイムラグが発生する。そのタイムラグがプロジェクト間の不整合を誘発する。	プロジェクトの計画変更の影響が全てのマネージャが素早く判断できること。

## 2.6 現状とあるべき姿のイメージ

本研究成果を適用しプロジェクト間の不整合を防ぐことの効果、現状である As Is のイメージとあるべき姿である To Be のイメージ図で比較する。イメージ図は図 2.8 のとおり。

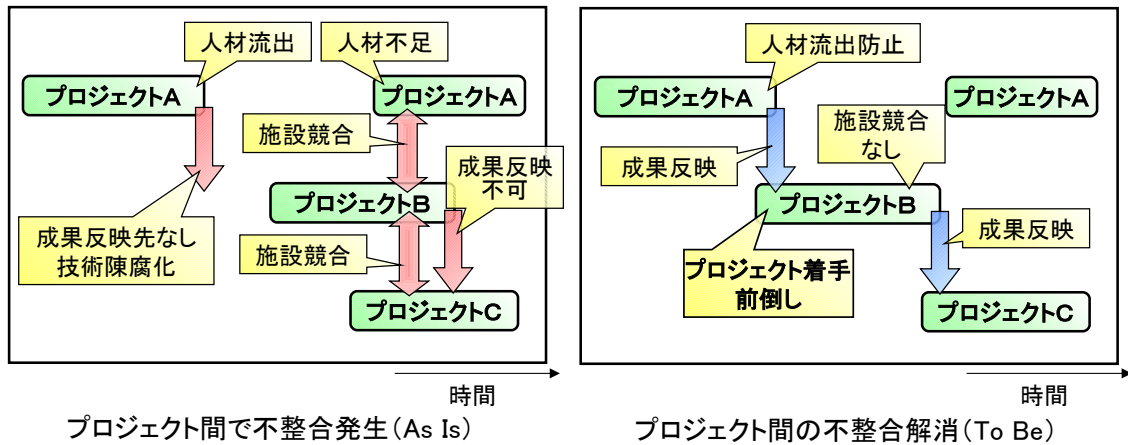


図 2.8 本研究の As Is と To Be のイメージ

プロジェクト A, B, C は別のポートフォリオに属するプロジェクトである。現状である As Is の場合は図 2.5 と同様にプロジェクト間で不整合が発生しているが、あるべき姿の To Be では、プロジェクト B の着手を少し早めるよう調整することにより、As Is で発生している問題を解決することができている。このように、戦略的な視点からプロジェクト群を直接マネジメントすることにより、プロジェクト間で発生する問題を防止することが可能となる。

### 3 要求分析

ユーザーが満足するシステムを構築するためには、ユーザー要求を満たすのみだけでなく、ユーザーの要求を分析し、システムが満たすべき仕様を明確化することが必要になる。そのためには、ユーザー要求に対し、欲しい成果物がどこまでであるのか境界を明確化するとともに、どういった成果物が欲しいのか成果物自体を明確化する必要がある。明確化した内容についてはアーキテクティングへの要求となるシステム仕様としてまとめる。

#### 3.1 システムの境界の明確化

ユーザーがどこまでの成果物が欲しいのかを明確にするために、システムのライフサイクルを明確化するとともに、コンテキスト図とユースケース図を使ってシステムの境界を明確化する。これにより、本研究の範囲が確定し、提案するシステムの境界が明確となる。

##### 3.1.1 ライフサイクル

システムのライフサイクルを明確化する。システムのライフサイクルは図 3.1 のとおり。

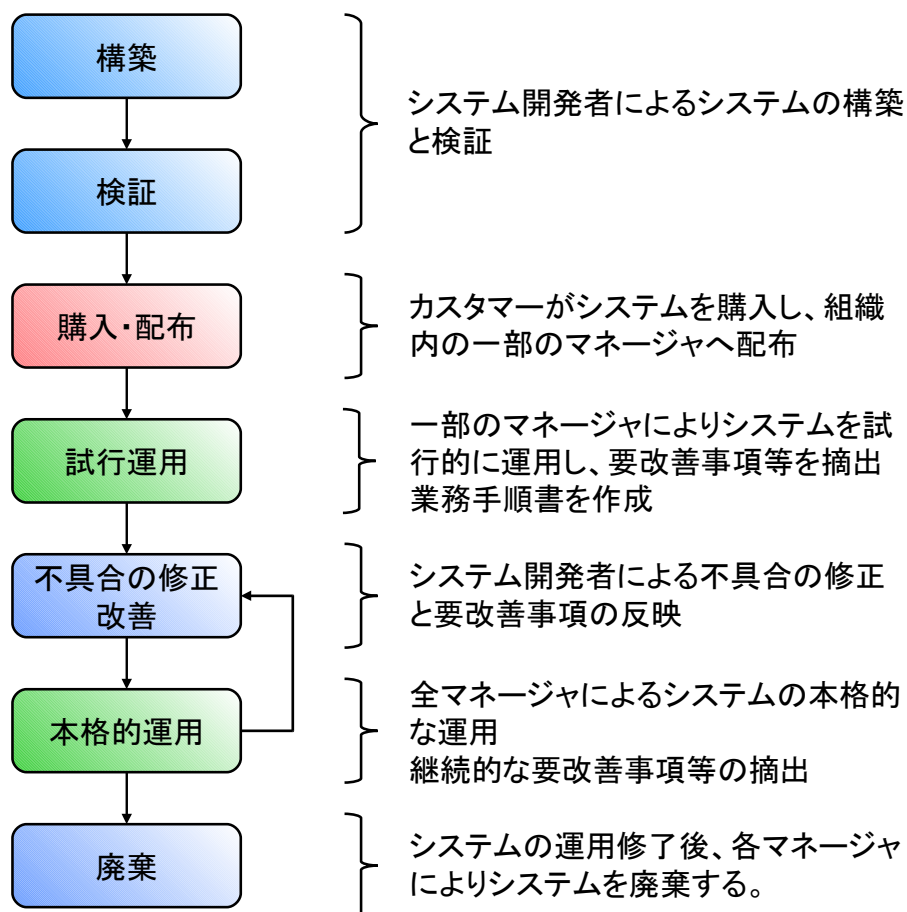


図 3.1 システムのライフサイクル

まず始めに、システムの開発者により本システムを構築し、ユーザーの要求を満足しているか検証を実施する。システム完成後、カスタマーが本システムを購入し、組織内の一部のマネージャに本システムを配布する。本システムが組織内で有効に活用できることを確認するとともに要改善事項を抽出するため、一部のマネージャにより試行的にシステムを運用する。また、システムの運用に必要な業務手順書を作成する。試行的にシステムを運用した結果、システムに不具合及び要改善事項があれば、システム開発者が不具合の修正及びシステムの改善を実施する。その後、組織内の全マネージャによる本格的な運用を開始する。本格的な運用期間内においても、要改善事項等の抽出を行い、必要に応じてシステム開発者がシステムの改善等を行う。本システムは運用終了後、各マネージャによりシステムを廃棄する。

### 3.1.2 コンテキスト図

システムの目的のひとつとして、戦略目標と複数のプロジェクトの整合を取ることがある。複数のプロジェクトはプログラムやポートフォリオといったコンポーネントの単位で管理されており、戦略目標とコンポーネントの整合をとることが必要である。そのため、システムの視点で見るとコンポーネントはデザインの対象となり、戦略目標はコンポーネントの整合を取るための要件となる。システムのもうひとつの目的として、プロジェクトの実行可能性を確保することがある。そのためには、プロジェクト間で競合する様々な要素を調整する必要がある。プロジェクト間で競合する要素を制約条件と置くと、制約条件も戦略目標と同様にプロジェクトの整合を取るための要件となる。

システムの境界を検討するにあたり、システムが必要とされる環境も明確化する必要がある。システムの環境としては、ユーザー要求で述べたとおり、大規模・複雑化・相互関連性の増加の傾向といった産業製品の傾向と、高品質・低価格・短納期といった産業製品に対するユーザーの要求がある。このような環境下においてユーザーの要求を満足するようにプロジェクトを行うためには、リスクを考慮することが重要となる。リスクを適切にマネジメントし、予定していたとおりプロジェクトを実施できれば問題はないが、やむを得ずプロジェクトの計画を変更することもある。これを、プロジェクト間の整合といった視点から見ると、戦略目標と制約条件を満足するようプロジェクトを計画しても、プロジェクトの計画変更により整合が崩れてしまう問題と考えられる。そのため、過去の経験からリスクを判断し、それがプロジェクトの整合性にどの程度影響を与えるのか把握することも必要となる。

システムのカスタマーは色々と想定されるが、戦略目標を設定しそれを達成することを必要とするエグゼクティブ・マネージャが最も典型的なカスタマーである。システムを使用するユーザーは、コンポーネントをマネジメントする、ポートフォリオ・マネージャ、プログラム・マネージャ及びプロジェクト・マネージャであるが、戦略目標等の妥当性を分析することも必要であることから、エグゼクティブ・マネージャもユーザーとなる。

本研究で提供するシステムは、方法論だけでなく、実際に各マネージャが使用できるシステムとして提供する。システムの動作環境は、各マネージャが簡単に使えることを考慮し、一般的なコンピュータ上で動作するソフトウェアとして提供をすることとする。

最後に、システムを提供するシステム開発者もシステムの環境として明確化する。システム開発者はコンピュータの開発環境を使用して、システムを開発する。

以上の点を踏まえたコンテキスト図を図 3.2 に示す。

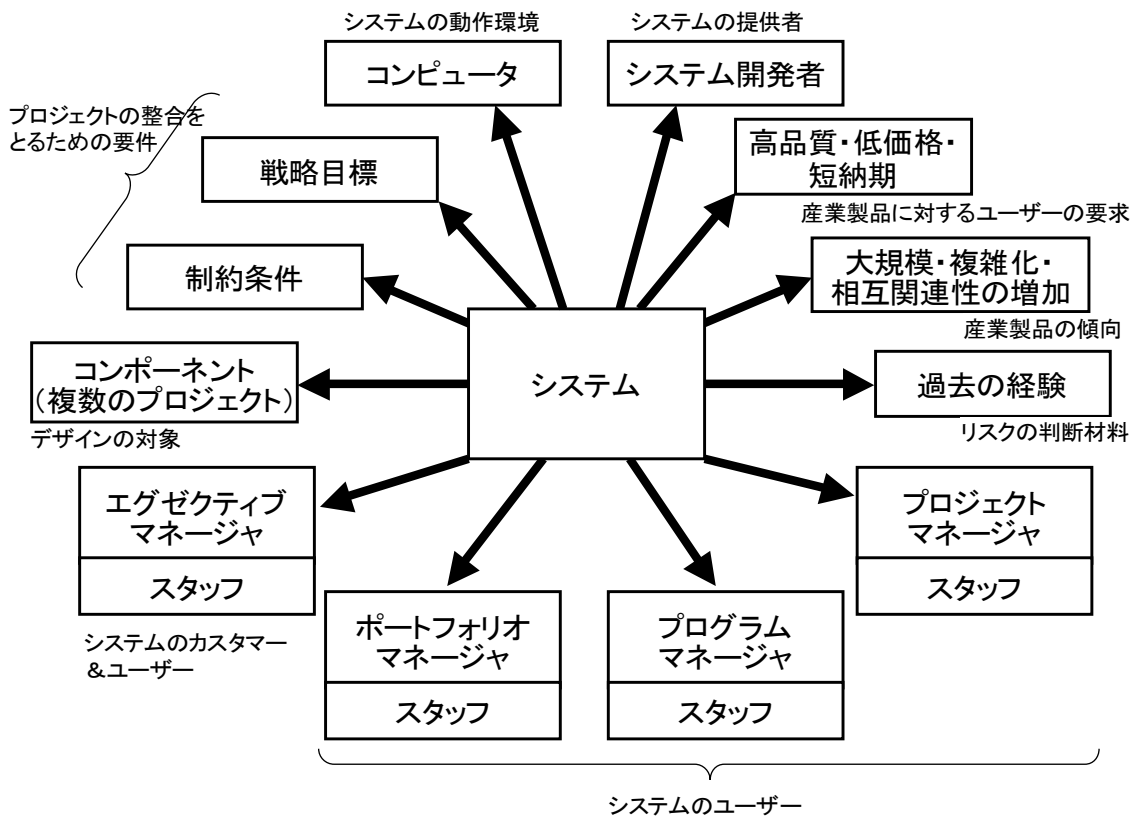


図 3.2 コンテキスト図によるシステムの境界の明確化

### 3.1.3 ユースケース図

システムのユーザーである各マネージャは、プロジェクトの整合を取る以外にも様々な活動を行う必要がある。各マネージャにとって、プロジェクトの整合を取ることはマネジメント活動のひとつと位置づけられる。よって、本システムのユースケースとしては、従来の階層的なマネジメントは維持しつつ、プロジェクトの整合をとる部分について本システムを使うといった方法となる。

システムと相互作用する外部要因を明確にするため、ユースケース図を使用してシステムに要求されている機能を明確にする。図 3.2 のコンテキスト図より、プロジェクトの整合を取るためには、プロジェクトの要件である戦略目標と制約条件のデータが必要であり、デザインの対象であるコンポーネントのデータが必要である。コンポーネントは複数のプロジェクトで構成されていることから、コンポーネントのデータとしてプロジェクトのデータが必要となる。また、過去の経験からリスクを判断する必要があるため、リスクケースを設定するデータも必要である。これらのデータはデータベースとしてシステム内に保持し、ユーザーからの要求によりデータベースのデータを使用してプロジェクトの整合がとれているかシミュレーションを行い、その結果をユーザーに表示する機能が必要となる。

システムを使用するユーザーは、コンテキスト図で示したように、エグゼクティブ・マネージャ、ポートフォリオ・マネージャ、プログラム・マネージャ及びプロジェクト・マネージャの各マネージャである。戦略目標データ、制約条件データ、リスクケースデータについては、戦略目標を設定し全てのコンポーネントをマネジメントする立場にあるエグゼクティブ・マネージャがデータを入力するが、戦略目標、制約条件及びリスクケースが変更

になった場合どのような影響があるのか把握するため、他のマネージャもデータを修正して分析を行うことが考えられる。よって、戦略目標データ、制約条件データ及びリスクケースデータを設定する権限はエグゼクティブ・マネージャが有するが、データを入力する機能としては、各マネージャが共通して必要となる。また、プロジェクトデータについては、プロジェクトの内容を一番良く知っているプロジェクト・マネージャがデータを入力するが、プロジェクトが変更になった場合どのような影響があるのか把握するため、ポートフォリオ・マネージャやプログラム・マネージャもプロジェクトデータを修正して分析を行うことが考えられる。よって、ポートフォリオ、プログラム及びプロジェクトのマネージャもプロジェクトデータを入力する機能が必要となる。ユースケース図を図 3.3 に示す。

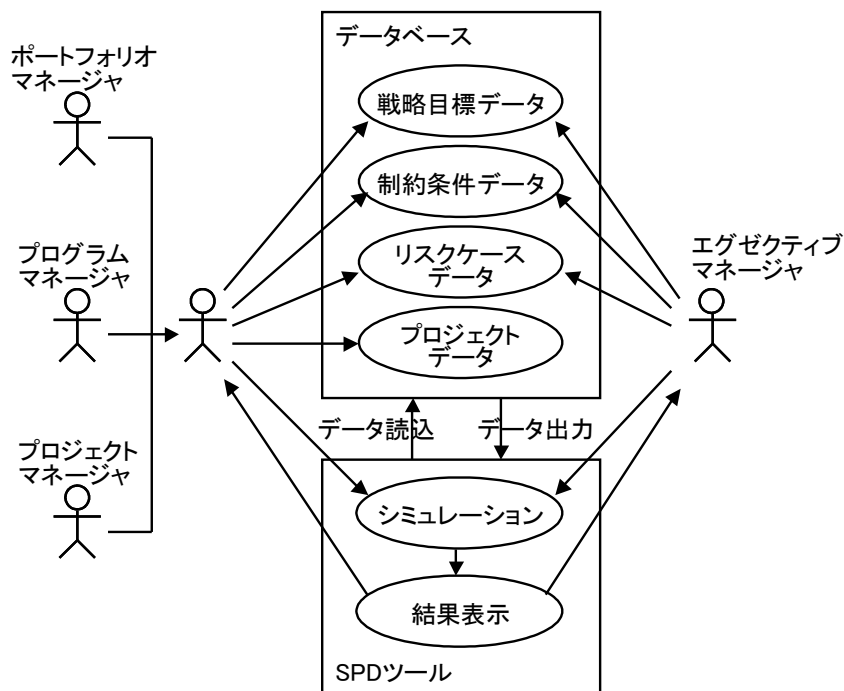


図 3.3 システムのユースケース

システムを、各マネージャがデータを入力しそのデータを保持するデータベースと、データを読み込みシミュレーションと結果を表示する戦略的プロジェクトデザインツールに分割する。戦略的プロジェクトデザインツールは、SPD ツール(Strategic Project Design Tool)と記述することとする。プロジェクトなどのデータは本研究で提案するシステム以外でも使用されるものであり、組織によってはすでにデータベース化されている場合もある。そのため、データベースについては本システム専用のシステムを提供することなく、既存のデータベースシステムを活用して機能を提供することとする。本研究で提供するシステムのみで必要となる部分については、SPD ツール側に機能を集約し提供することとする。

本システムはと従来の階層的なマネジメントとの関係を図 3.4 に示す。

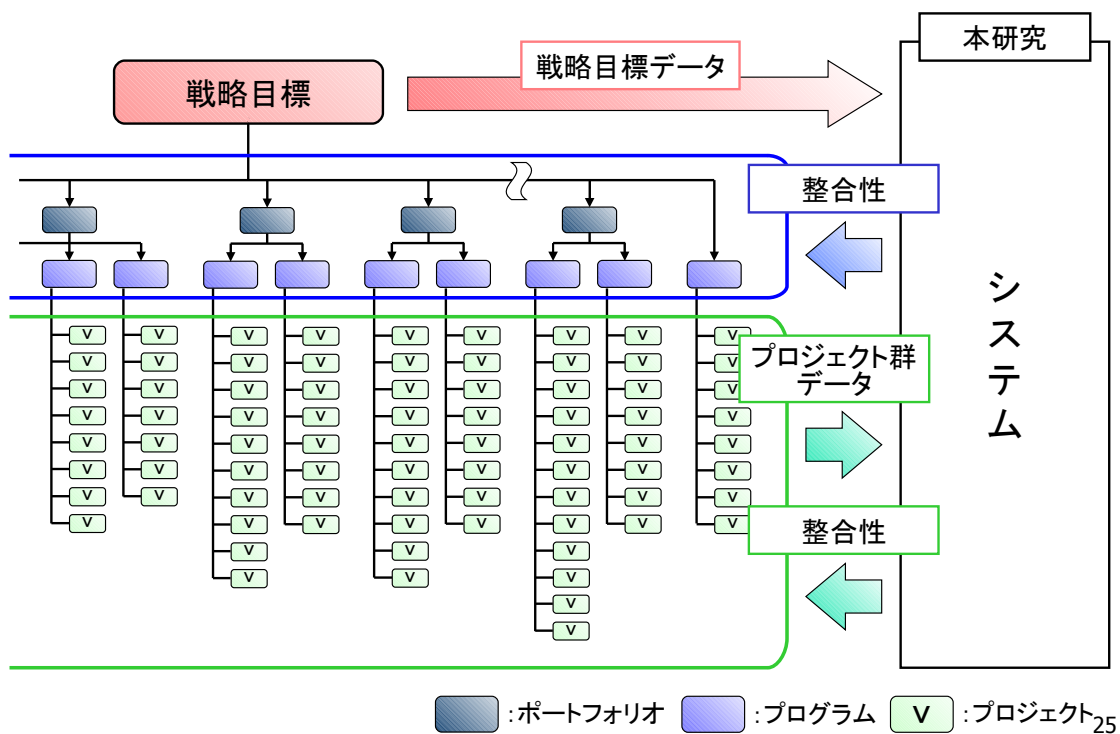


図 3.4 本システムと従来の階層的なマネジメントとの関係

従来の階層的なマネジメントにおいて設定される戦略目標をデータとしてシステムに入力する。プロジェクトのデータは階層構造をとることなく、それぞれのプロジェクトデータをプロジェクト群データとしてシステムに入力する。システムは入力されたデータをシミュレーションし、プロジェクトの整合性を評価した結果を出力する。出力したデータは、各マネージャに提供される。このように、本研究で提案するシステムは、各マネージャに対し従来の階層的なマネジメント手法を補助するシステムとなる。

## 3.2 システム自体の明確化

成果物自体を明確化するためには、要求を実現するためのシステムの挙動を定義するとともに、要求段階におけるシステム要求機能の抜けをなくす、機能分析が必要である。機能分析として、システムの動作の流れをフロー図で記述する FFBD(Function Flow Block Diagram)<sup>9)</sup>を作成する。システムはデータベースと SPD ツールに分かれている。データベースには各種データを入力し保存する機能が必要であり、SPD ツールにはデータベースのデータを読み込みシミュレーションを行い、その結果を表示する機能が必要である。データベースと SPD ツールは別々の独立した別々のサブシステムとして構築するため、FFBD も別々に作成する。データベースの FFBD を図 3.5a に SPD ツールの FFBD を図 3.5b に示す。

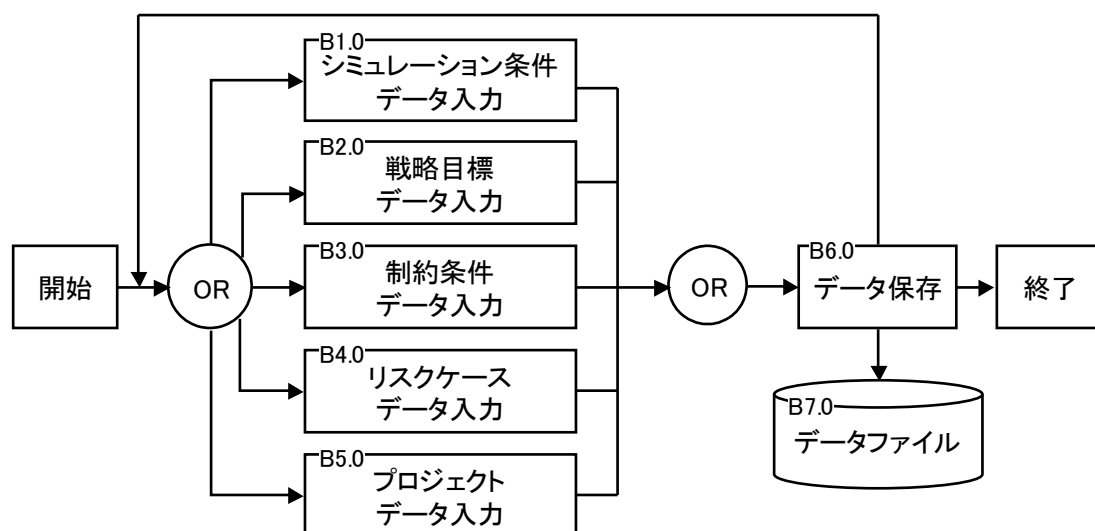


図 3.5a FFBD によるデータベースの機能分析

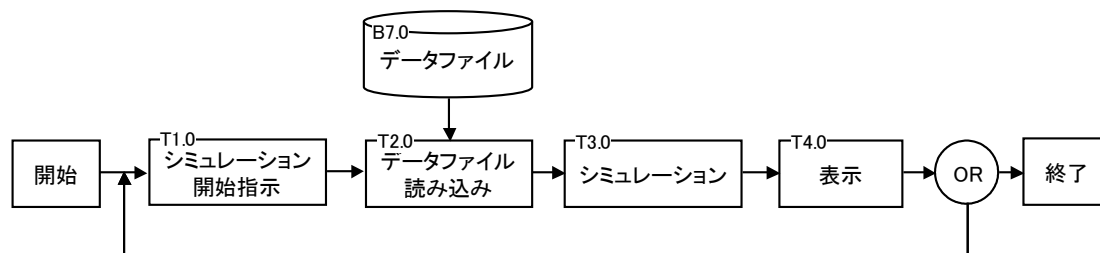


図 3.5b FFBD による SPD ツールの機能分析



### 3.3 システム仕様

システムの境界の明確化とシステム自体の明確化をした結果に基づき、システムに対する仕様を設定する。設定した仕様は「システム仕様」と定義する。また、システム仕様を設定した理由も明記する。システム仕様の記述については、要求の内容が検証可能な記述になるよう具現化するとともに、各要求が文章として明確となるよう記述する。システム仕様は表 3.1 のとおり。表中の No. については、システム全体に関する内容については「S」、データベースに関する内容については「D」、SPD ツールに関する内容については「T」をつけて番号を付与した。

システム仕様はアーキテクティングへの要求となり、アーキテクティングの Verification を行う際の評価項目となる。

表 3.1 システム仕様

No.	システム仕様	設定理由
S1	システムは各マネージャが使用するコンピュータで動作すること。	コンテキスト図のコンピュータ
D1	データベースは既存のデータベースを使用すること。	ユースケース図のデータベース
D2	各マネージャは、データベースを共用できること。	ユースケース図の各マネージャ
D3	データベースに保存されているデータは、SPD ツールで読み取り可能なデータ形式であること。	FFBD の B7.0 及び T2.0
D4	各マネージャは、データベースのシミュレーション条件データを入力、修正及び削除できること。	FFBD の B1.0
D5	各マネージャは、データベースの戦略目標データを入力、修正及び削除できること。	ユースケース図の戦略目標データ FFBD の B2.0
D6	各マネージャは、データベースの制約条件データを入力、修正及び削除できること。	ユースケース図の制約条件データ FFBD の B3.0
D7	各マネージャは、データベースのリスクケースデータを入力、修正及び削除できること。	ユースケース図のリスクケースデータ FFBD の B4.0
D8	各マネージャは、データベースのプロジェクトデータを入力、修正及び削除できること。	ユースケース図のプロジェクトデータ FFBD の B5.0
D9	各マネージャは、入力、修正及び削除したデータを保存できること	FFBD の B6.0
T1	各マネージャは、SPD ツールに対しシミュレーションの実行を指示できること。	ユースケース図のシミュレーション FFBD の T1.0

T2	SPD ツールは、データベースからデータを読み込めること。	FFBD の T2.0
T3	SPD ツールは、プロジェクトの整合性をシミュレーションできること。	ユースケース図のシミュレーション FFBD の T3.0
T4	SPD ツールは、シミュレーションの結果を表示できること。	ユースケース図の結果表示 FFBD の T4.0
T5	各マネージャは、SPD ツールの表示により、プロジェクトの整合性が確認できること。	ユースケース図の結果表示
T6	各マネージャは、プロジェクトの整合性を確認後、SPD ツールに対し再度シミュレーションの実行を指示できること。	ユースケース図のシミュレーション FFBD の T4.0 後の OR 処理
C2	各マネージャとは、エグゼクティブ・マネージャ、ポートフォリオ・マネージャ、プログラム・マネージャ及びプロジェクト・マネージャとする。ただし、各マネージャを補佐するスタッフが実施してもよいものとする。	ユースケース図の各マネージャ

### 3.4 トレーサビリティ

ユーザー要求とシステム仕様のトレーサビリティを確保するため、ユーザー要求とシステム仕様の対応表を作成する。ユーザー要求とシステム仕様の対応表は、表 3.2 のとおり。システム仕様の文章の先頭に記述している番号は表 3.1 の No.に対応している。システム仕様の内容を満足することにより、ユーザー要求を満足することができる。

表 3.2 ユーザー要求とシステム仕様の対応

ユーザー要求	システム仕様	説明
プロジェクト間で必要な情報を伝達できること。	D1 各マネージャは、データベースを共用できること。 D8 各マネージャは、データベースのプロジェクトデータを入力、修正及び削除できること。	プロジェクト間で必要な情報をプロジェクトデータとして入力し、そのデータベースを各マネージャで共用することにより、プロジェクト間で必要な情報を伝達することが可能なシステムとする。
複数のポートフォリオやプログラムに属するプロジェクトをマネジメントできること。	T5 各マネージャは、SPD ツールの表示により、プロジェクトの整合性が確認できること。	SPD ツールはプロジェクト単位での整合性を確認する仕様とし、プロジェクトがどのポートフォリオやプログラムに属しているか関係のないシステムとする。
様々なツールや技法を駆使することなく、戦略目標とプロジェクトの整合性やプロジェクト間の整合性を確保できること。	S1 システムは各マネージャが使用するコンピュータで動作すること。 T5 各マネージャは、SPD ツールの表示により、プロジェクトの整合性が確認できること。	本システムひとつを使用するだけで、戦略目標とプロジェクトの整合性やプロジェクト間の整合性を確認することができる。また、システムは各マネージャが使用するコンピュータで動作するため簡単に整合性が確認できる。
プロジェクトの計画変更の影響が全てのマネージャが素早く判断できること。	D1 各マネージャは、データベースを共用できること。 D8 各マネージャは、データベースのプロジェクトデータを入力、修正及び削除できること。 T5 各マネージャは、SPD ツールの表示により、プロジェクトの整合性が確認できること。	各マネージャがプロジェクトの計画変更をプロジェクトデータの修正として入力し、SPD ツールでシミュレーションすることで、その影響を素早く判断できる。

## 4 アーキテクティング

### 4.1 アーキテクティングの方針

アーキテクティングとは、システム仕様で要求されている機能・性能を、システムを構成する要素に配分して構成要素の仕様を明確にするとともに、構成要素間のインタフェースを明確化することにある。アーキテクティングは、システム仕様が同じであってもアーキテクティングの方針が異なると、システムが大きく異なるといった特徴がある。そのため、アーキテクティングを行う前に、アーキテクティングの方針を明確にする。

#### 4.1.1 プロジェクト群

膨大なプロジェクトを効率良くマネジメントするため、従来ではポートフォリオやプログラムといった階層を構成しマネジメントをする手法が行われている。本システムはこの階層構造に起因して発生する問題を解決することを目的のひとつとしていることから、プロジェクトを階層構造として管理しないシステムとして構築する。膨大なプロジェクトを階層構造を持たないプロジェクト群ととらえ、戦略視点からプロジェクト群を直接デザインする手法をとる。階層構造を構成するプロジェクトと階層構造をとらないプロジェクト群の違いを図 4.1a と図 4.1b に示す。

プロジェクト間で考慮すべき情報は戦略レベルで集約して管理する構造とし、プロジェクトの整合性も戦略視点から評価する。これにより、階層構造に起因する問題を発生させないことが可能となる。しかし、この手法を採用することにより別の問題が発生する。膨大なプロジェクトをまとめてマネジメントするためには、膨大な情報を処理する必要があり、マネージャの管理能力を超える。従来ではプロジェクトを階層的にわかりやすくまとめ、情報を簡略化することによりマネージャの管理能力に納まるようにして問題を回避している。本研究では膨大な情報の処理という問題に対し、定量化と可視化を行うことで問題を解決する。定量化と可視化については 4.1.2 項と 4.1.3 項で述べる。

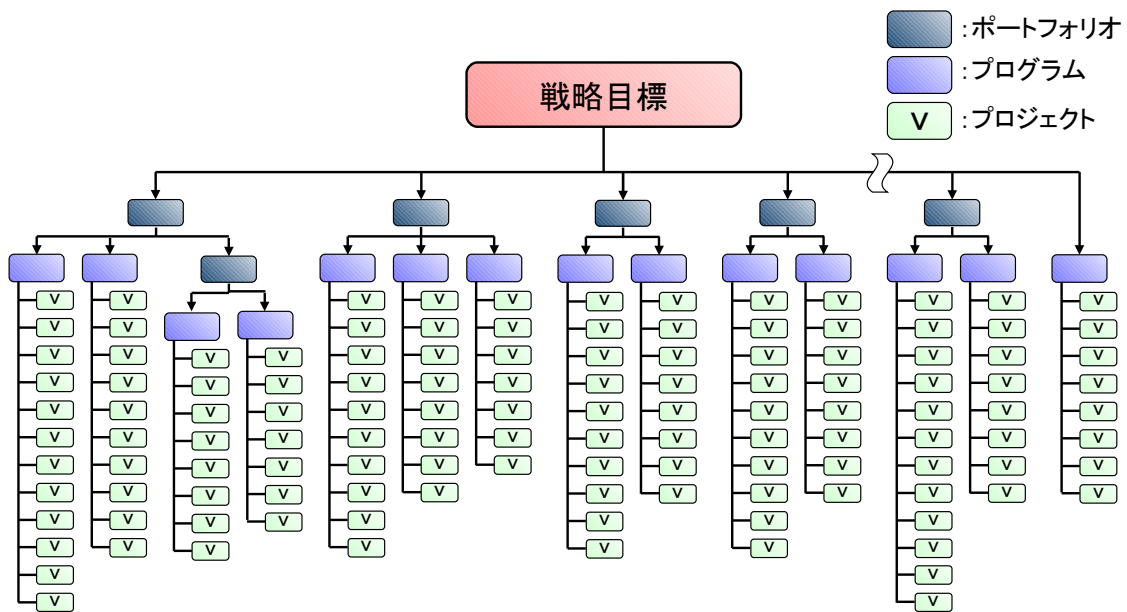


図 4.1a 階層構造を構成するプロジェクト

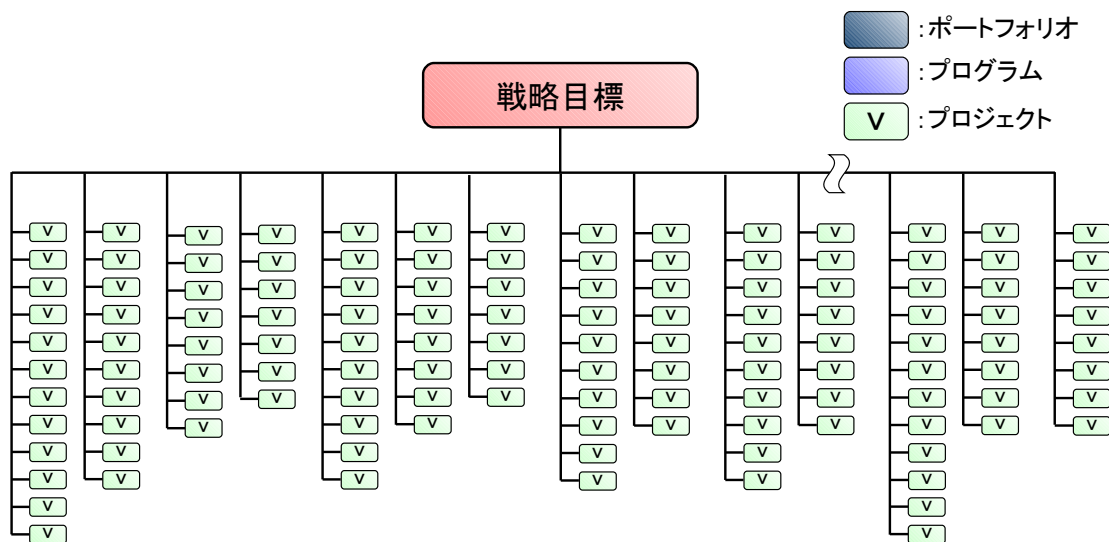


図 4.1b 階層構造をとらないプロジェクト群

## 4.1.2 定量化

システムズ・エンジニアリングやマネジメントの分野では、さまざまな技法やツールが提案されているが、複雑な問題を解決するために定量化の手法が使われることが多い。品質機能展開(QFD: Quality Function Deployment)<sup>(10)</sup>の手法では、顧客の声を聞き重要性を0139で定量化し、マトリックスを利用して機能を展開していくことで何が重要であるのかを分析している。階層的意思決定法(AHP: Analytic Hierarchy Process)<sup>(8)</sup>の手法では、人間の主観的判断を一対比較で定量化する等の工程を行うことで、合理的に意思を決定することを可能としている。Pugh Selection<sup>(11)</sup>の手法では、ある項目を基準として、それより優れているか劣っているかを比較し定量化することにより最適な項目を絞り込む手法をとっている。この他にも、故障モード影響解析(FMEA: Failure Mode Effects Analysis)<sup>(1)(12)</sup>や設計構造行列(DSM: Design Structure Matrix)<sup>(13)</sup>など、難しい問題を要素を定量化することにより解決する手法が多くある。

本研究においても、様々な問題を解決するため、戦略目標データ、制約条件データ、リスクケースデータ及びプロジェクトデータは定量化して扱うことを原則とする。次に、定量化することによりどのような問題が解決できるか、どのように定量化するのかを述べる。

### 4.1.2.1 膨大な情報の処理

階層構造のマネジメント行わない場合、プロジェクト群の膨大な情報を処理する必要がある。一方、コンピュータは定量化された膨大なデータを処理するのに優れている。この特性を生かし、プロジェクト群で考慮すべき情報をすべて定量化し、コンピュータによるシミュレーションを実施する。これにより、プロジェクト群の膨大な情報を処理することが可能となる。

### 4.1.2.2 評価方法の統一

プロジェクト群の整合性を取るためには、様々な事項を考慮しなくてはならない。考慮しなくてはいけない項目として、例えば、利益、経費、人的資源、技術、施設、器材及び材料などがある。これらの項目にはそれぞれ特徴があり、整合性を取る場合には各々に応じたツールや技法が必要となる。これらの項目に対し、統合したひとつの手法により整合性を取るためには、各々の項目を同じ土俵で評価できるようにする必要がある。そのための手法としてすべての項目のデータを定量化する手法をとる。すべての情報を定量化することができれば、「定量化したデータの比較」という統一した手法により評価を行うことが可能となる。

### 4.1.2.3 曖昧性の排除

プロジェクト群の整合性を正しく確保するためには、考慮すべき要素から曖昧性を排除することが重要である。戦略目標や制約条件を曖昧に設定すると、目標を達成したのか、制約条件をクリアできているのかを判断することができない。また、プロジェクトのデータが曖昧であると、プロジェクトが整合しているのか不整合なのかが曖昧になる。文章で戦略目標を設定する場合、「・・・と吻合を図りつつ・・・」、「・・・を重視し・・・」などの表現を使うことがあるが、曖昧性を排除するためには、すべての情報を定量化することが重要である。

#### 4.1.2.4 定量化の方法

定量化には、数値、割合及びプラスマイナスなど、様々な種類がある。本システムでは、定量化を行えばどのような種類であってもよいこととする。どのような定量化であっても、コンピュータでシミュレーションを行うことは可能であり、プロジェクトの整合性を評価することが可能である。

次に定量化の方法であるが、大きく2つに分けることができる。1つ目は、利益、経費及び工数など、はじめから定量化されているものである。これらの項目については、その数値をそのまま使うことができる。次に技術力や知名度など、定量化されていないものである。これらについては、項目の特性や評価の方法に応じて指標や優劣などにより定量化する。例えば、技術力については、アメリカ航空宇宙局（NASA: the National Aeronautics and Space Administration）が開発した技術の成熟度を定量化する TRL( Technology Readiness Level) <sup>(14)</sup> のような有名な指標があり、このような指標を活用する方法もある。宇宙航空研究開発機構（JAXA: Japan Aerospace eXploration Agency）で使用されている TRL を図 4.2 <sup>(15)</sup> に示す。

### 技術の成熟度(TRL) Technology Readiness Level

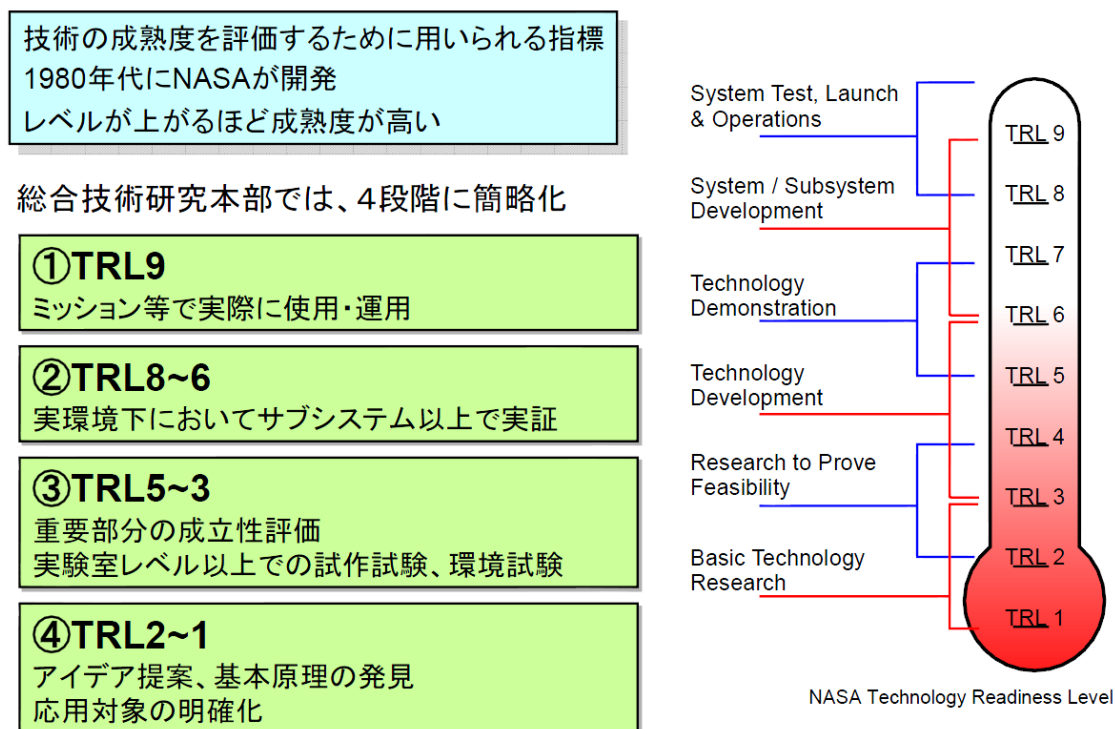


図 4.2 JAXA で使用している TRL <sup>(15)</sup>

### 4.1.3 可視化

結果の表示方法として可視化の手法がある。可視化は、人が直接見ることのできない現象・事象・関連性を見えるようにすることで、直感的に分かり易くなるため、様々な分野で取り入れられている手法である。システムズ・エンジニアリングの分野でも様々な手法がある。顧客価値連鎖分析(CVCA: Customer Value Chain Analysis)<sup>(10)</sup>では、どんな価値がどこで誰に提供されるのかということ进行分析するため、利害関係者のお金、サービス及び情報などの流れをネットワーク状に図示している。特性要因図(Fish bone diagram)<sup>(1)(8)</sup>では、問題の原因追求を容易にするため、特性と要因の関係を系統的に結んで魚の骨のようなツリー状の図を作成する。設計構造行列(DSM: Design Structure Matrix)<sup>(13)</sup>では、設計や製品開発をどのような工程で行えばよいのか分析するため、工程間の関係をマトリックスで図示する。そのほかにも、統合化定義方法論(IDEF0: Integration DEfinition 0)<sup>(16)</sup>、クリティカル・パス法(CPM: Critical Path Method)<sup>(1)(8)</sup>、故障の木解析(FTA: Fault Tree Analysis)(FTA)<sup>(12)</sup>及びWBS(Work Breakdown Structure)<sup>(1)(8)</sup>など、多くの手法がある。

本研究においても、プロジェクトの整合性を評価した結果を可視化して表示することとする。プロジェクト群の膨大なデータを整理してわかり易く可視化することにより、現在の問題点が明確化するとともに、各マネージャが容易にプロジェクト群の状況を把握することができる。これにより、戦略からプロジェクトまで共通の認識を持つことが可能となるとともに、迅速な対応が可能となる。

### 4.1.4 汎用化

プロジェクトの整合を取るために考慮すべき項目は、組織やプロジェクトの内容により、異なっている。また、同じ項目であっても、ある組織にとってはそれは戦略目標であり、ある組織にとっては制約条件となる場合もある。また、同じプロジェクトであっても、プライムメーカのレベルまで整合をとるのか、ベンダーのレベルまで整合を取りたいのかによって、考慮しなくてはならない項目及び入力しなくてはならないデータの範囲が異なってくる。これらの多様なニーズに対応できるようにするためには、戦略目標、制約条件、リスクケース及びプロジェクト群といった個別の組織によらないモデルについてはシステムとして提供し、その中の項目はユーザーが設定できるようにする必要がある。また、評価方法もいくつかのケースを提供しユーザーが選択できるようにするとともに、新たな評価方法を追加できるようにする必要がある。本システムは、特定の組織に特化せず汎用化を図ることとさまざまな分野で活用できるようにアーキテクティングを実施する。

## 4.2 機能設計

機能設計とは、システム仕様で設定した機能を分割し、その機能を構成する下位機能の集合に置き換える作業である。要求分析における機能分析で作成したFFBD(Function Flow Block Diagram)<sup>(9)</sup>に対し、機能の細分化を行う。

データベースについては、既存のデータベースを活用することから、データベースの機能の細分化は実施せず、SPDツールの機能の細分化のみを実施する。SPDの機能は、T1.0シミュレーション開始指示、T2.0データファイル読み込み、T3.0シミュレーション及びT4.0表示の4つの機能から構成されている。T1.0シミュレーション開始指示は、ユーザーからのシミュレーション開始操作を読み取り処理を開始する機能のみである。T2.0データファイルの読み込みの機能についてはデータファイルに記録されている情報を



順番に読み込む機能のみである。また、T4.0表示の機能についてはT3.0シミュレーションで得られた結果を画面に表示する機能のみである。よって、これらの機能については細部化を行う必要はない。T3.0シミュレーションの機能について細分化を行う。細分化を行ったFFBDを図4.3に示す。

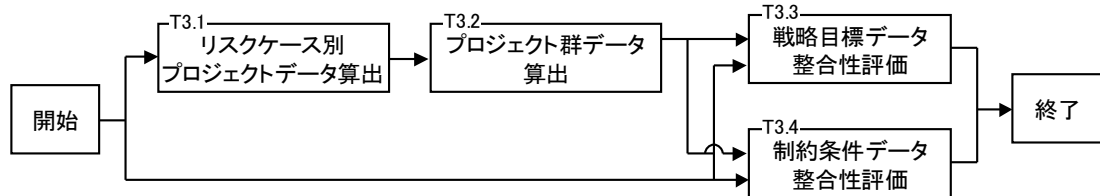


図 4.3 シミュレーションの機能設計

T1.0シミュレーション開始指示、T2.0データファイルの読み込みの後、T3.0シミュレーションの処理が行われる。T3.1リスクケース別プロジェクトデータ算出の機能は、プロジェクトデータとリスクケースデータを使用し、リスクケース別にプロジェクトデータがどのように変化するかを算出する機能である。T3.2プロジェクト群データ算出の機能は、戦略目標及び制約条件とプロジェクトデータの整合性が評価できるよう、すべてのプロジェクトデータを統合したプロジェクト群データを算出する。T3.3戦略目標データ整合性評価の機能は、シミュレーション条件に基づき戦略目標データとプロジェクト群データを比較し整合性を評価する。同様に、T3.4制約条件データ整合性評価の機能は、シミュレーション条件に基づき制約条件データとプロジェクト群データを比較し整合性を評価する。それぞれの評価の完了をもってT3.0シミュレーションの処理は終了し、T4.0表示の処理が開始される。SPDツール全体の機能フローのイメージを図4.4に示す。

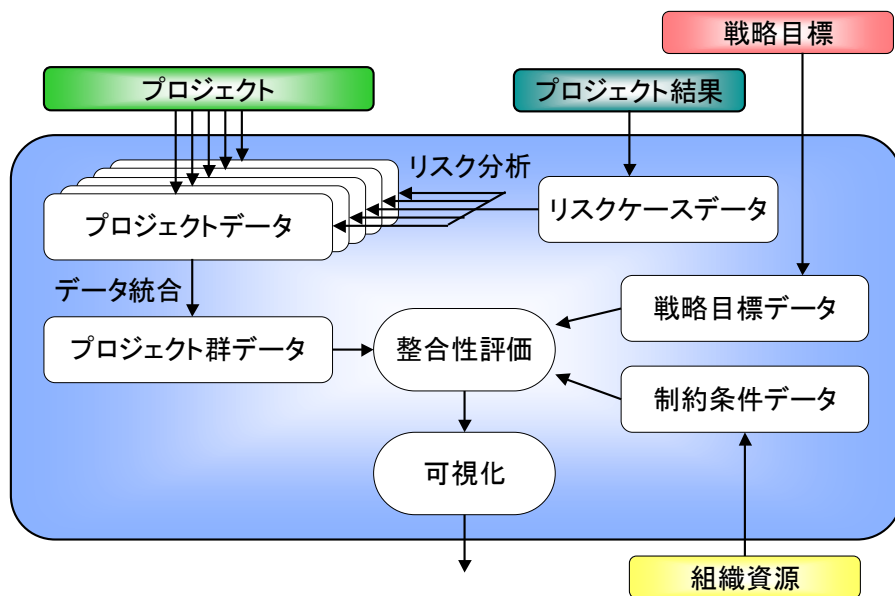


図 4.4 SPD ツールの機能フローのイメージ

### 4.3 物理設計

物理設計とは、物理階層構造を設定するとともに、機能設計にて細分化した機能を物理階層に割り付ける作業である。物理設計結果を図 4.5 に示す。図中の赤色で示した番号は、FFBD の番号であり、機能が物理構造のどこに割り付けられるかを示している。

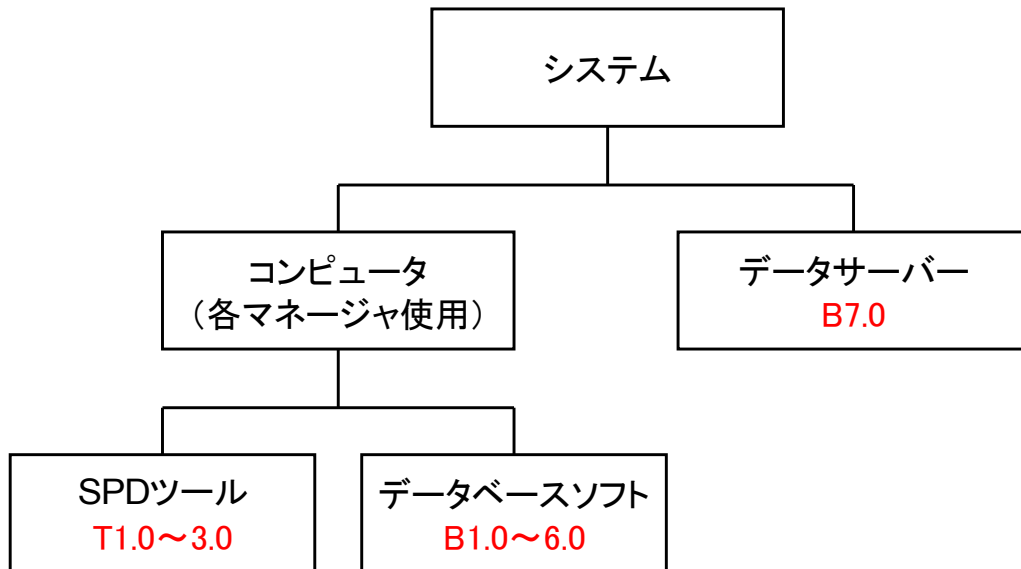


図 4.5 物理設計

システムは、各マネージャが使用するコンピュータとデータを共有するデータサーバーから構成する。コンピュータには、データサーバーにデータの入力等を行うデータベースソフトと、プロジェクトの整合性をシミュレーションする SPD ツールから構成する。アーキテクティングで細分化したシミュレーションの機能はすべて SPD ツールの機能であり、また、SPD ツールはソフトウェアとして実装することから、物理構成をこれ以上の細分化することなく機能を割り付けた。

## 4.4 インターフェイス設計とアーキテクチャ設計結果

インターフェイス設計として、システムとの境界であるユーザーとのインターフェイスと、システムの内部であるサブシステム間のインターフェイスを設計する。

ユーザーはキーボードとマウスを使用しデータベースソフトにデータを入力する。データベースソフトからデータサーバーへのデータ保存形式は、データ交換用のデータ形式のデファクトスタンダードとして古くから多くの表計算ソフトやデータベースソフトで使われている **Comma-Separated Values** 形式（以下「**CSV 形式**」という。）を採用する。**CSV 形式**はいくつかの項目をカンマで区切ったテキストデータおよびテキストファイルとなっている。データ保存形式にデファクトスタンダードの **CSV 形式** を採用することにより、ユーザーが様々なデータベースソフトを使用することに対応できる。**SPD** は、**CSV 形式** のデータ読み込みに対応するよう設計する。ユーザーは **SPD ツール** にシミュレーションの開始を指示すると、**SPD ツール** はユーザーにシミュレーションの結果を表示する。インターフェイス設計結果を図 4.6 に示す。

図 4.6 は、機能設計、物理設計及びインターフェイス設計の結果であり、アーキテクチャ設計結果となる。アーキテクチャ設計は、サブシステムにおけるシステムデザインへの要求となり、システムデザインの **Verification** を行う際の評価項目となる。

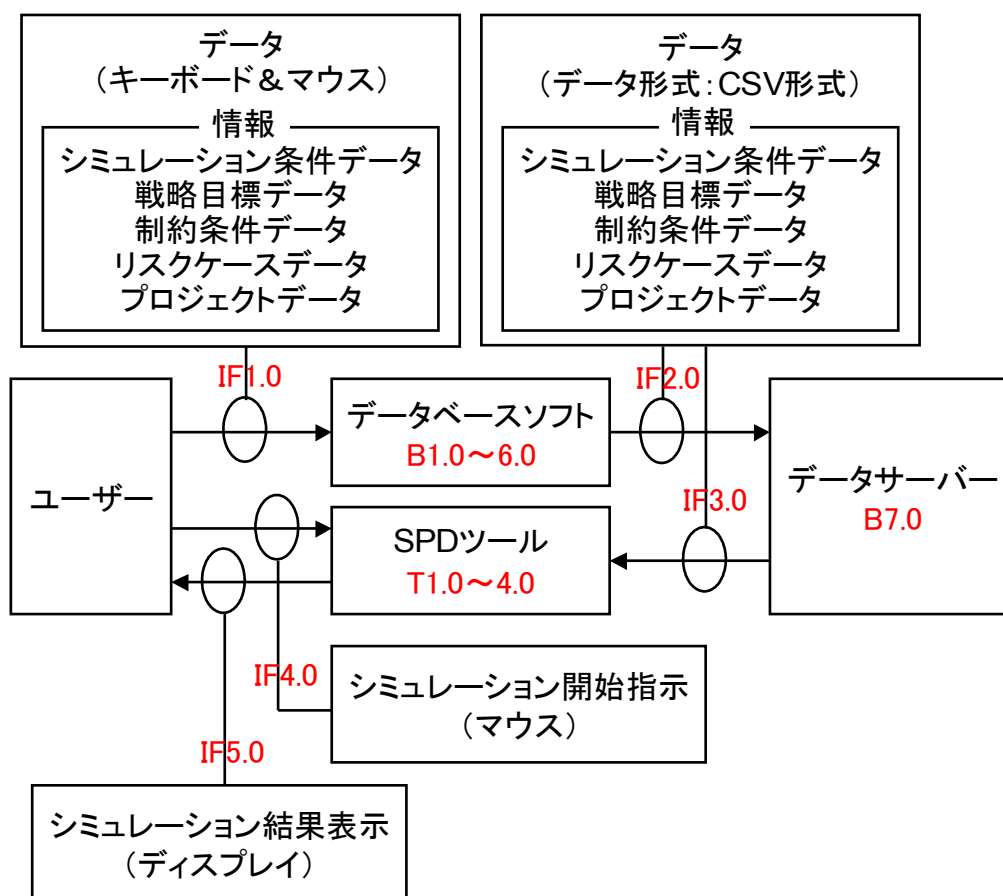


図 4.6 アーキテクチャ設計結果

## 4.5 トレーサビリティ

システム仕様とアーキテクチャ設計結果のトレーサビリティを確保するため、システム仕様とアーキテクチャ設計結果の対応表を作成する。システム仕様とアーキテクチャ設計結果の対応表は、表4のとおり。アーキテクチャ設計結果の内容を満足することにより、システム仕様を満足することができる。

表4 システム仕様とアーキテクチャ設計結果の対応

システム仕様	アーキテクチャ設計結果	説明
システムは各マネージャが使用するコンピュータで動作すること。	物理設計：データベースソフト及びSPDツール	データベースソフト及びSPDツールは一般事務用パソコンで動作する仕様とする。
データベースは既存のデータベースを使用すること。	システムデザインへの要求とする。	アーキテクティングに関する要求事項ではないため、システムデザインへの要求とする。
各マネージャは、データベースを共用できること。	物理設計：データサーバー	データベースはデータサーバーで共有される。
データベースに保存されているデータは、SPDツールで読み取り可能なデータ形式であること。	インターフェイス設計：IF3.0 データ	データ交換用のデファクトスタンダードの形式を採用した。また、SPDツールにCSV形式の読み込み機能を付加した。
各マネージャは、データベースのシミュレーション条件データを入力、修正及び削除できること。	機能設計：B1.0 シミュレーション条件データ入力 インターフェイス設計：IF1.0 データ	データを入力し上書きすることにより、入力、修正及び削除が可能である。
各マネージャは、データベースの戦略目標データを入力、修正及び削除できること。	機能設計：B2.0 戦略目標データ入力 インターフェイス設計：IF1.0 データ	
各マネージャは、データベースの制約条件データを入力、修正及び削除できること。	機能設計：B3.0 制約条件データ入力 インターフェイス設計：IF1.0 データ	
各マネージャは、データベースのリスクケースデータを入力、修正及び削除できること。	機能設計：B4.0 リスクケースデータ入力 インターフェイス設計：IF1.0 データ	

各マネージャは、データベースのプロジェクトデータを入力、修正及び削除できること。	機能設計：B5.0 プロジェクトデータ入力 インターフェイス設計：IF1.0 データ	データを入力し上書きすることにより、入力、修正及び削除が可能である。
各マネージャは、入力、修正及び削除したデータを保存できること	機能設計：B6.0 データ保存 インターフェイス設計：IF2.0 データ	機能設計でデータの保存機能を付加した。
各マネージャは、SPD ツールに対しシミュレーションの実行を指示できること。	機能設計：T1.0 シミュレーション開始指示 インターフェイス設計：IF4.0 シミュレーション開始指示	機能設計でシミュレーションの開始機能を付加した。
SPD ツールは、データベースからデータを読み込めること。	機能設計：T2.0 データファイル読み込み 物理設計：データサーバー インターフェイス設計：IF3.0 データ	データベースと SPD ツールでデータサーバーを共有することにより、データの読み込みを可能とする。
SPD ツールは、プログラムの整合をシミュレーションできること。	機能設計：T3.0 シミュレーション	機能設計でシミュレーション機能を付加した。
SPD ツールは、シミュレーションの結果を表示できること。	機能設計：T4.0 表示 インターフェイス設計：IF5.0 シミュレーション結果表示	機能設計で表示機能を付加した。
各マネージャは、SPD ツールの表示により、プロジェクトの整合性が確認できること。	機能設計：T4.0 表示 インターフェイス設計：IF5.0 シミュレーション結果表示 表示内容はシステムデザインへの要求とする	機能設計で表示機能を付加した。 表示内容についてはアーキテクティングに関する要求事項ではないため、システムデザインへの要求とする。
各マネージャは、プロジェクトの整合性を確認後、SPD ツールに対し再度シミュレーションの実行を指示できること。	機能設計：T4.0 表示の次の OR 分岐 インターフェイス設計：IF4.0 シミュレーション開始指示	OR 分岐により、再度シミュレーションの実行が可能である。
各マネージャとは、エグゼクティブ・マネージャ、ポートフォリオ・マネージャ、プログラム・マネージャ及びプロジェクト・マネージャとする。ただし、各マネージャを補佐するスタッフが実施してもよいものとする。	物理設計：データベースソフト及び SPD ツール	データベースソフト及び SPD ツールは一般事務用パソコンで動作する仕様とする。

## 5 システムデザイン

### 5.1 データベースソフト

#### 5.1.1 既存のソフトウェアの選択

システム仕様に「データベースは既存のデータベースを使用すること。」としたため、既存のソフトウェアからデータベースソフトを選択する。定量化をしてデータベースに入力しなくてはならないデータのうち、もっとも量が膨大であるのがプロジェクトのデータである。プロジェクトのデータは基本的に各プロジェクト・マネージャ又はそのスタッフが入力することになるため、一般的な事務で使用する表計算ソフトをデータベースのソフトウェアとして選択する。また、プロジェクトに関するデータについては、すでに表計算ソフトで管理している場合も多いため、その点でもメリットがある。本研究については、表計算ソフトとして Microsoft 社の Excel を使用する。

#### 5.1.2 データ入力形式

定量化したデータを表計算ソフトに入力する方法を規定する。データの inputs は、SPD ツールでシミュレーションを実施するのに必要なデータを不足なく入力する必要があるとともに、入力業務を行う各マネージャやそのスタッフが分かり易いようにする必要がある。そのため、表計算ソフトのシートにテンプレートを作成し、ユーザーはテンプレートを参考にしながら入力を行えるようにする。具体的なデータ構造は、SPD ツールで必要とするデータを網羅する必要があるため、SPD ツールのデザインで述べることとする。

### 5.2 データサーバー

システム仕様に「データベースは既存のデータベースを使用すること。」としたため、データサーバーについても、特別なデータサーバーを必要としないシステムとする。近年のコンピュータはネットワークに接続されているのが一般的であり、そのネットワークを活用することでファイルの共用が可能である。よって、データベースソフトで保存したデータを既存のネットワークを通じて共用することで、データサーバーの機能を実現する。

## 5.3 SPD ツール

### 5.3.1 開発言語

SPD ツールは各マネージャ又はそのスタッフがユーザーであり、様々なコンピュータで動作することが必要である。そのため、プラットフォームに依存しないアプリケーションソフトウェアの開発と配備を行うことができる Java をプログラムの開発言語とする。Java はいったんプログラムを作成すれば Java が動作する環境であればどのコンピュータでも動作させることができるといった特徴がある。また、Java は事務作業で使用するようなコンピュータだけでなく、組み込みシステムや携帯機器にも使用されている。開発言語に Java を選択することで、SPD ツールを多様な環境で使用することが可能となる。

### 5.3.2 データファイル読み込み

#### 5.3.2.1 戦略目標データ、制約条件データ及びプロジェクトデータ

戦略目標データと制約条件データはプロジェクト群データと比較し整合性を評価するため、戦略目標データと制約条件データの項目を合わせた項目がプロジェクト群の項目と対応している必要がある。また、プロジェクト群データはプロジェクトデータから計算されるため、各プロジェクトデータはプロジェクト群データのいずれかの項目と対応している必要がある。例えば、プロジェクトデータに「〇〇経費」の項目がある場合、プロジェクト群データにも「〇〇経費」の項目が必要となる。逆に、プロジェクト群データに「〇〇技術」の項目がある場合、プロジェクトデータに「〇〇技術」の項目が必ず必要ということはない。戦略目標データ、制約条件データ、プロジェクト群データ及びプロジェクトデータの関係は図 5.1 のとおり。

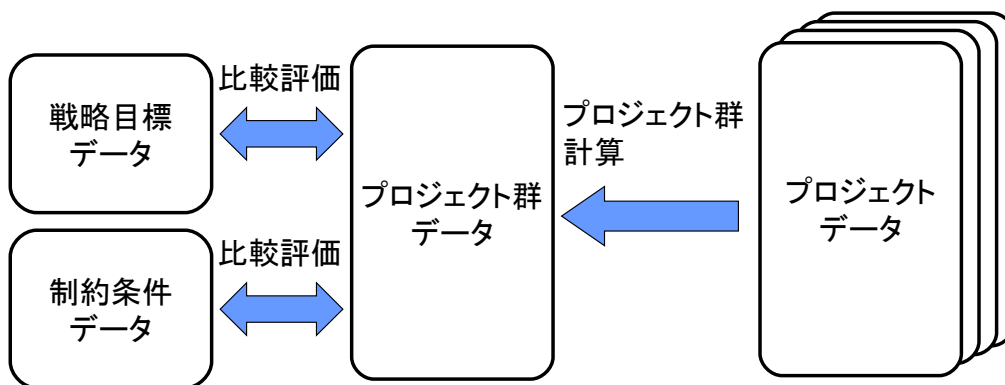


図 5.1 戦略目標データ及び制約条件データとプロジェクトデータの関係

各項目のデータについては、同じ時間軸で比較評価する必要がある。そのため、各項目のデータは時間軸上に展開し、同じ時間軸で比較を行えるようにする。以上の点を考慮するとシミュレーションに必要なデータの構造は図 5.2 のとおりとなる。プロジェクト群を計算する方法を定義している「群計算」と整合性を評価する方法を定義している「評価法」の意味はプロジェクト群データ算出と戦略目標と制約条件の整合性評価で述べる。

戦略目標データ			年度			
項目	群計算	評価法	X年	X+1年	X+2年	X+3年
利益	合計	以上	X,XXX	X,XXX	X,XXX	X,XXX
〇〇工数	合計	以上	XXX	XXX	XXX	XXX
□□技術	最大	以上				

プロジェクト群データ			年度			
項目	群計算	評価法	X年	X+1年	X+2年	X+3年
利益	合計	以上	X,XXX	X,XXX	X,XXX	X,XXX
経費	合計	以上	X,XXX	X,XXX	X,XXX	X,XXX
〇〇工数	合計	以上	XXX	XXX	XXX	XXX
□□技術	最大	以上	X	X	X	X
△△技術	最大	以上	X	X	X	X
××器材	最大	以上	X	X	X	X

制約条件データ			年度			
項目	群計算	評価法	X年	X+1年	X+2年	X+3年
経費	合計	以下	X,XXX	X,XXX	X,XXX	X,XXX
△△技術	最大	以下	X	X	X	X
××器材	合計	以下	X	X	X	X

図 5.2 シミュレーションに必要なデータの構造

### 5.3.2.2 リスクケースデータ

リスクケースは、予定通りプロジェクトが行えなかった場合のプロジェクトデータを算出するのに使用する。そのため、どの項目がどの程度、計画と異なるのか、リスクケース別のデータが必要となる。補正の方法は、各項目ごとにユーザーが設定する。また、同じ項目であってもプロジェクトの種類によって、リスクケースが異なることが考えられるため、プロジェクトの種類に応じてリスクケースを設定できる必要がある。本研究で作成するシステムにおいては、「乗算」、「加算」及び「減算」の補正方法を提供する。例えば、予定より経費が増大するリスクケースを計算するためには、補正方法として「乗算」、データとして経費が増加する係数を設定すれば良い。新たに補正方法が必要となった場合には、SPD ツールに補正方法を追加する必要がある。リスクケースに必要なデータの構造は図 5.3 のとおりとなる。

プロジェクトタイプ	項目	リスクケース					
		ケース1	ケース2	ケース3	ケース4	ケース5	
研究	航空	経費	1.0	1.2	1.4	1.6	1.8
		期間	1.0	1.1	1.2	1.3	1.4
	電子	経費	1.0	1.0	1.2	1.4	1.6
		期間	1.0	1.0	1.1	1.2	1.3
:	:	:	:	:	:	:	
開発	航空	経費	1.0	1.1	1.1	1.2	1.2
		期間	1.0	1.0	1.1	1.1	1.2
	電子	経費	1.0	1.2	1.4	1.5	1.5
		期間	1.0	1.2	1.4	1.6	1.8
:	:	:	:	:	:	:	

図 5.3 リスクケースに必要なデータの構造



### 5.3.3 リスクケース別プロジェクトデータ算出

リスクケースのプロジェクトタイプと一致しているプロジェクトデータに対し補正計算を行う。リスクケースで設定した項目とプロジェクトデータの項目が一致した場合は、各リスクケース毎にプロジェクトデータを補正する。プロジェクトデータをリスクケース別に補正するイメージを図5.4に示す。

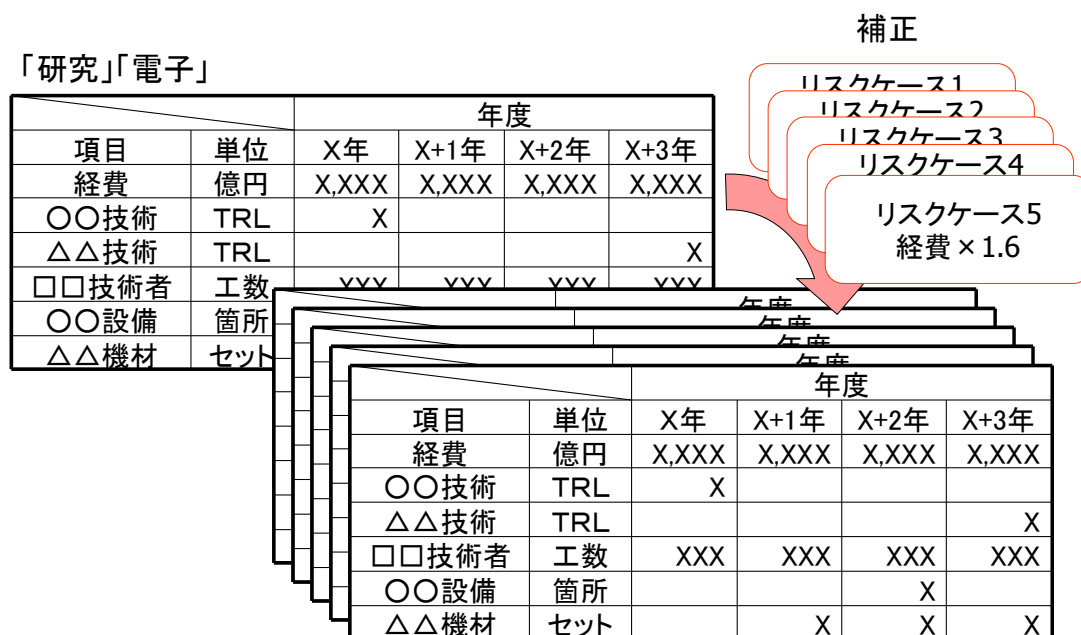


図5.4 プロジェクトデータをリスクケース別に補正

図5.4はリスクケースとして「研究」「電子」のプロジェクトに対し経費の補正を設定した例である。リスクケース5に経費が1.6倍となるケースを設定した場合、「研究」「電子」のプロジェクトデータの経費を1.6倍する補正計算を実施する。同様に、リスクケース1～4についてもリスクケースで設定した補正計算を行う。リスクケースが設定されていない項目については補正計算は行わず、リスクケース1～5まで同じデータとなる。この補正計算を実施することにより、プロジェクトデータはリスクケース別のプロジェクトデータとして算出される。

### 5.3.4 プロジェクト群データ算出

リスクケース別に補正したプロジェクトデータは、同一のリスクケースごとにプロジェクト群データを算出する。算出方法は戦略目標データ及び制約条件データの項目ごとにユーザーが設定する。本研究で作成するシステムにおいては、「合計」と「最大」の2種類の算出方法を提供する。利益、経費、工数及び必要器材等であれば、プロジェクトのデータを合計し評価することが必要であり、必要な技術力等であれば、プロジェクトデータの最大値を取り評価することが必要となる。新たに計算方法が必要となった場合には、SPD ツールに計算方法を追加する必要がある。プロジェクト群データの算出方法のイメージを図 5.5 に示す。

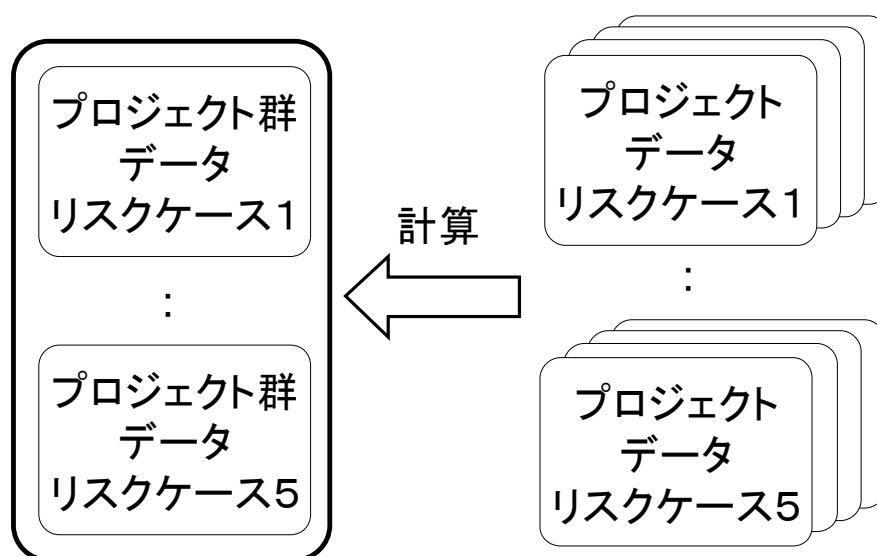


図 5.5 プロジェクト群データの算出方法

### 5.3.5 戦略目標と制約条件の整合性評価

リスクケース別に作成したプロジェクト群データは、リスクケース別に戦略目標データ及び制約条件データと比較を行い、整合性がとれているか評価を行う。比較の結果がどのような状態であれば整合が取れていると判断するかは、戦略目標データ及び制約条件データの項目ごとにユーザーが設定する。本研究で作成するシステムにおいては、「以上」と「以下」の2種類の評価方法を提供する。例えば、利益や技術力などのプロジェクト群データが戦略目標データ以上であることが良い場合は「以上」を、経費や器材などのプロジェクト群データが制約条件データ以下であることが良い場合は「以下」を選択する。新たに評価方法が必要となった場合には、SPD ツールに評価方法を追加する必要がある。

### 5.3.6 可視化手法の選択

整合性を評価した結果は可視化して表示する。可視化を行うツールや技法は非常に多いが、その手法を大別すると、概ね、ネットワーク型、ツリー型及びマトリックス型に分けられる。従来の手法と可視化の種類のマップを図 5.6 に示す。

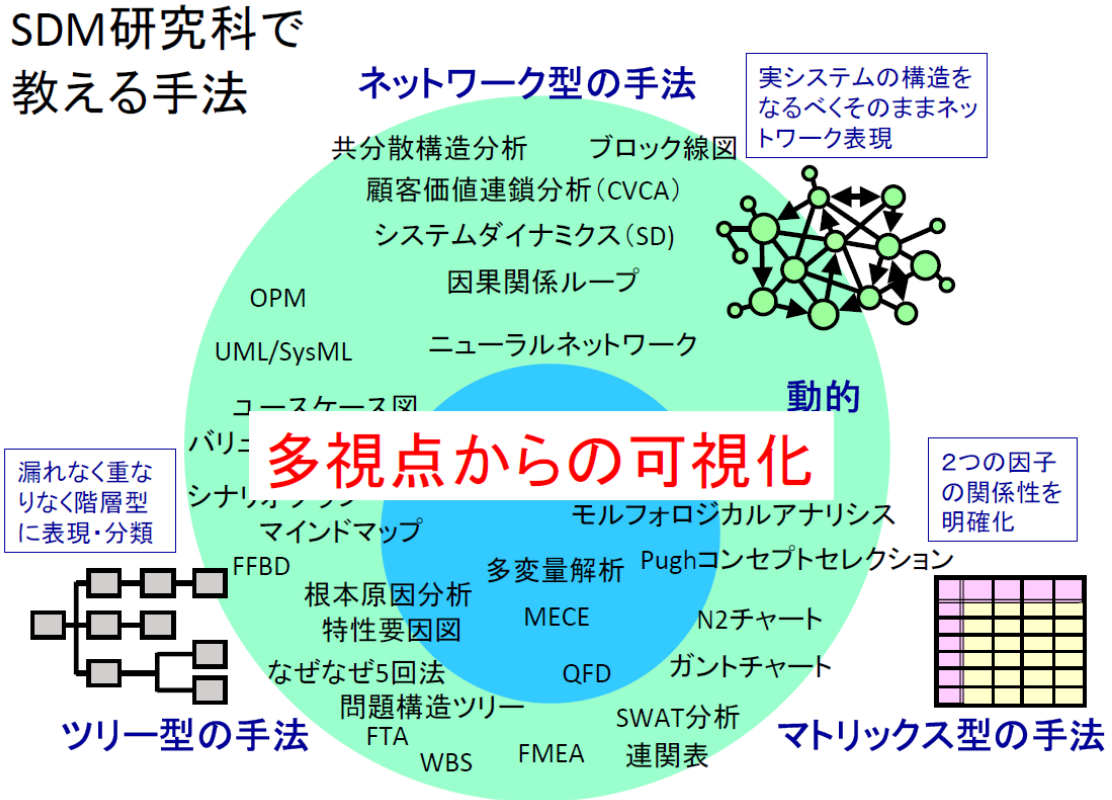


図 5.6 従来の手法と可視化の種類<sup>(17)</sup>

本システムの可視化の手法として適している手法を Pugh Selection<sup>(11)</sup> を使用して選択する。Pugh Selection を表 5.1 に示す。評価項目として、データ種類の適合性と膨大なデータ量への適合性を設定した。データ種類の適合性は、可視化の手法が本システムのデータと適合するか評価する。適合していない場合、可視化を行うことができない又は可視化をしても状態を理解するのが困難といった問題が発生する。膨大なデータ量への適合性は、本システムが膨大なデータを扱うことを想定していることから設定した。膨大なデータを扱うことに適合しない場合、可視化をしても状況が把握できないといった問題が発生する。

ネットワーク型を DATUM とし評価を行った結果、マトリックス型が優れていることが判明した。ネットワーク型は、膨大なデータを扱うとネットワークが複雑になり状況が把握できないといった問題があり、ツリー型は、本システムで扱うデータが階層的でないことから適合性が低いといった問題があった。

表 5.1 可視化手法の選択

手法	ネットワーク型	ツリー型	マトリックス型
特徴	実システムの構造をなるべくそのままネットワーク表現	漏れなく重なりなく階層型に表現・分類	2つの因子の関係性を明確化
代表的手法	CVCA <sup>(10)</sup> IDEFO <sup>(16)</sup>	特性要因図 <sup>(1)(8)</sup> FTA <sup>(12)</sup>	DSM <sup>(13)</sup> マトリックス型WBS <sup>(1)</sup>
データ内容の適合性	D A	- データ内容が階層的でないため適合しない	+ 項目と時間の2つの因子であり適合する
膨大なデータ量への適合性	T U M	+ 膨大なデータを整理して表現するのに優れている	+ 膨大なデータを整理して表現するのに優れている
結論		1 + 1 -	2 +

具体的な可視化の方法はマトリックスの縦軸を項目とし横軸を時間とする。プロジェクト群データは、リスクケース別に整合性を評価しているため、どのリスクケースまで整合性がとれているかは、マトリックスの色で表現する。青色であればすべてのリスクケースで整合性が取れていることを示し、赤色であればすべてのリスクケースで整合性が取れていないことを表わす。マトリックスの全体を見渡し、赤色に近い部分ほど不整合が起こる可能性が高いことを分かる。表示のイメージを図 5.7 のとおり。

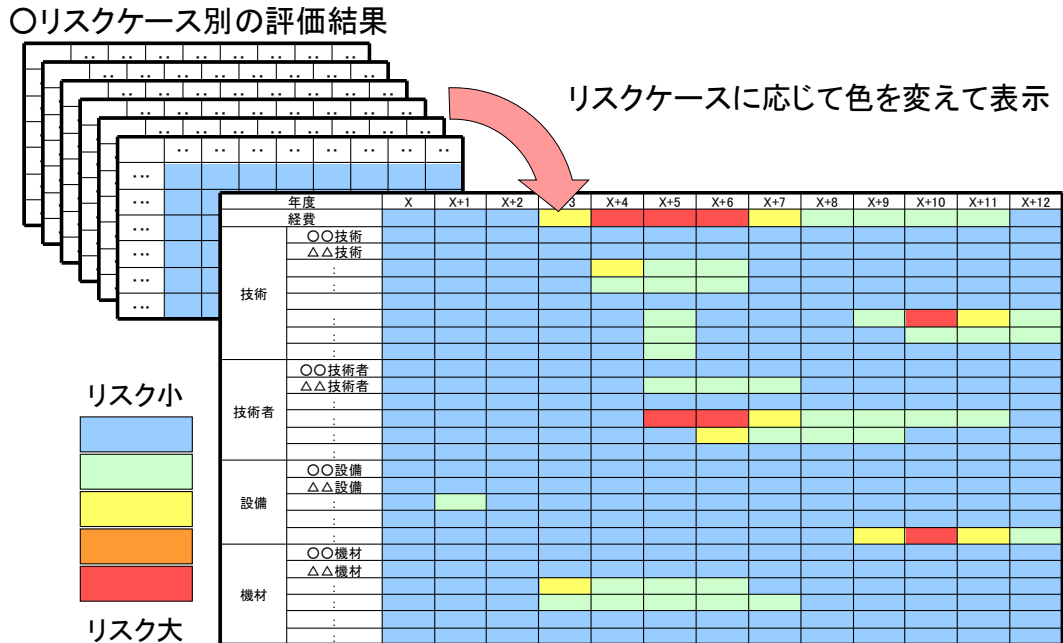


図 5.7 SPD ツールの表示イメージ

図 5.7 でリスクの大小を色で表示する方法を提示したが、白黒印刷で資料を配布する場合色の違いを判別できないといった問題が発生する。白黒印刷にも対応できるように、リスクの大小をシンボルの大きさで表示するバブルチャートのような表示機能も付加する。この表示形式では、シンボルが大きいほど不整合が起こる可能性が高いことを示すこととなる。

## 5.4 トレーサビリティ

アーキテクティングにおいてシステム仕様とアーキテクティングのトレーサビリティを確保したが、一部のシステム仕様はシステムデザインへの要求とした。それらの項目に対するトレーサビリティを確保するため、システム仕様と本項でデザインしたサブシステムの対応表を作成する。なお、アーキテクティングとシステムデザインのトレーサビリティについては、アーキテクティングの機能をそのままシステムデザインにブレイクダウンしているため記述は省略する。システム仕様とシステムデザインの対応表は、表 5.2 のとおり。システムデザインの内容を満足することにより、システム仕様を満足することができる。

表 5.2 システム仕様とシステムデザインの対応

システム仕様	システムデザイン	説明
データベースは既存のデータベースを使用すること。	表計算ソフトを使用 (本システムにおいては、Microsoft 社の Excel を使用)	データベースとして広く活用されている表計算ソフトを採用した。
各マネージャは、SPD ツールの表示により、プロジェクトの整合性が確認できること。	マトリックス型の表示方法を採用	従来の表示方法を分類及び分析し、最適な方法を選択した。

## 6 インテグレーション

### 6.1 インテグレーション計画

システムをどのような順番で構築するのかを述べる。システムの構築順番は、サブシステムの検証を、どのような順番で、どのように組み合わせて行うのかに関連する。よって、どのような組み合わせでサブシステムを検証すれば良いのかという視点でインテグレーション計画を検討する。

システムの物理構成を見ると、データベースソフト、データサーバー及びSPDツールの3つで構成されている。データベースソフトとデータサーバーは2つでデータベースとしての機能を有しており、2つのサブシステムを組み合わせて検証することが効率的である。よって、第1段階は、データベースソフトとデータサーバーをインテグレーションして検証を行い、SPDツールは単独で検証を行うこととする。第2段階で、すべてのサブシステムをインテグレーションし検証を行うこととする。物理設計にインテグレーションの段階を追記した図を図6.1に示す。

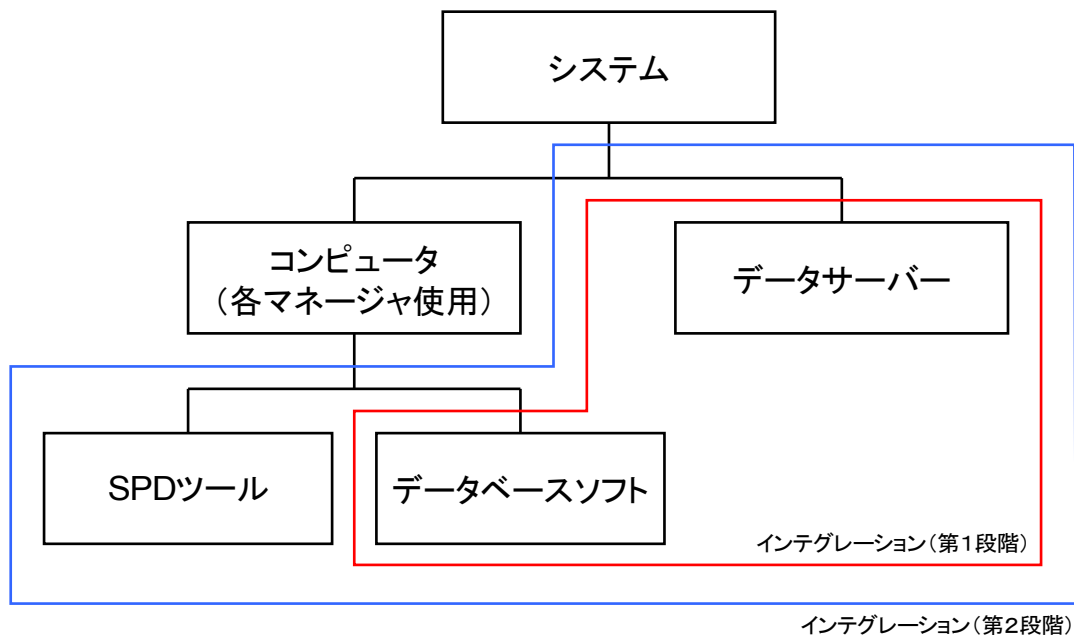


図 6.1 インテグレーションの段階

## 6.2 インテグレーション結果

インテグレーション計画に基づきインテグレーションを実施した。データベースソフトの画面例を図 6.2 に、シミュレーション実施後の SPD ツールの画面例を、カラー表示については図 6.3a に、白黒表示については図 6.3b に示す。図 6.3a と図 6.3b は同じデータを使用し、表示方式のみ変更している。SPD ツールのソースコードについては付録に添付する。

A1	B	C	D	E	F	G	H	I	J	K	L	M
1	C											
2	C	戦略的システムデザイン データベースソフト(データ入力シート)										
3	C											
4	C	シミュレーション条件データ										
5	C	表示形式 カラー=0,白黒=1	開始年度	シミュレーソ 年数	戦略目標 テナメ項目数	制約条件 テナメ項目数	リスクケース テナメ項目数	プロジェクト 数				
6	BD	0	2011	15	13	11	0	2				
7	C											
8	C	戦略目標データ										
9	C											
10	C	項目	プロジェクト 群	評価方法	年度							
11	SD	研究経費	合計	以下	2011	2012	2013	2014	2015	2016	2017	2018
12	SD	開発経費	合計	以下								
13	SD	生産経費	合計	以下								
14	SD	設計工数(航空A)	合計	以上								
15	SD	設計工数(航空B)	合計	以上								
16	SD	設計工数(宇宙A)	合計	以上								
17	SD	設計工数(エンジン)	合計	以上								
18	SD	設計工数(電子A)	合計	以上								
19	SD	製造工数(航空A)	合計	以上								
20	SD	製造工数(航空B)	合計	以上								
21	SD	製造工数(宇宙A)	合計	以上								
22	SD	製造工数(エンジン)	合計	以上								
23	SD	製造工数(電子A)	合計	以上								
24	C											
25	C	制約条件データ										
26	C											
27	C	項目	プロジェクト 群	評価方法	年度							
28	LD	技術レベル(航空A)	最大	以下	2011	2012	2013	2014	2015	2016	2017	2018
29	LD	技術レベル(航空B)	最大	以下								
30	LD	技術レベル(宇宙A)	最大	以下								
31	LD	技術レベル(エンジン)	最大	以下								
32	LD	技術レベル(電子A)	最大	以下								
33	LD	材料A	合計	以下								
34	LD	材料B	合計	以下								
35	LD	設備A	合計	以下								
36	LD	設備B	合計	以下								
37	LD	器材A	合計	以下								
38	LD	器材B	合計	以下								
39	C											
40	C	リスクケースデータ										
41	C											
42	C	プロジェクトタイプ	項目	補正方法	リスクケース							
43	RD	研究航空機	研究経費	乗算	ケース1	ケース2	ケース3	ケース4	ケース5			
44	RD	開発エンジン	開発経費	乗算								
45	C											
46	C	プロジェクトデータ										
47	PN	航空機の要素技術研究										
48	PT	研究航空機										
49	C											
50	C	項目	未使用	未使用	年度							
51	PD	研究経費			2011	2012	2013	2014	2015	2016	2017	2018
52	PD	設計工数(航空A)										
53	PD	製造工数(航空A)										
54	PD	技術レベル(航空A)										
55	PD	材料A										
56	PD	設備B										
57	PD	器材A										
58	C											
59	PN	エンジンの開発										
60	PT	開発エンジン										
61	C											
62	C	項目	未使用	未使用	年度							
63	PD	開発経費			2011	2012	2013	2014	2015	2016	2017	2018

図 6.2 データベースソフトの画面例

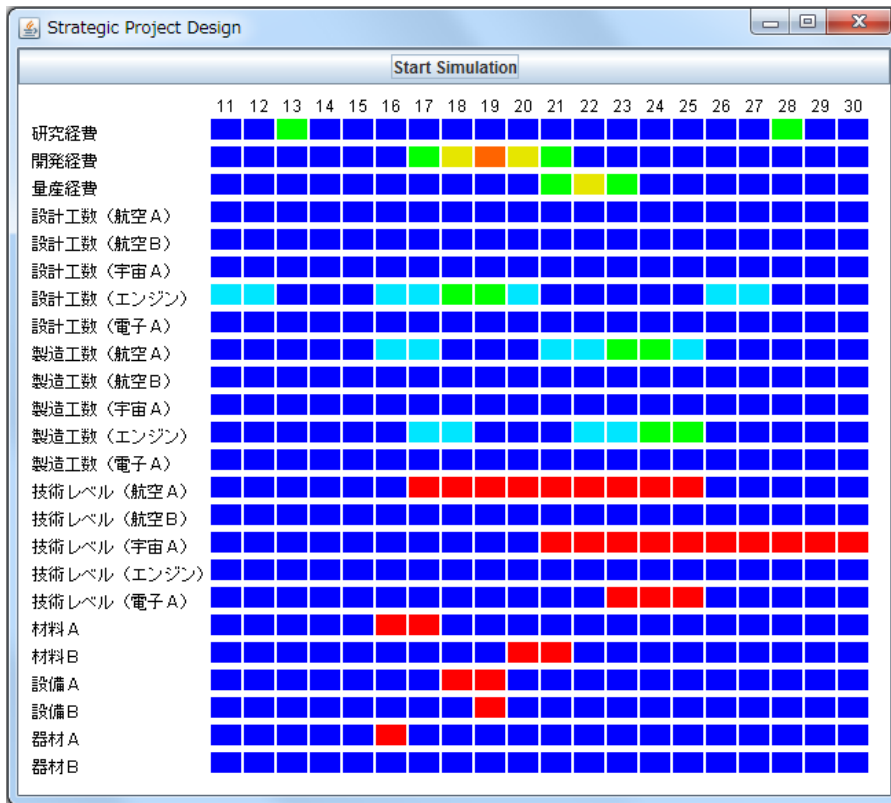


図 6.3a SPD ツールの画面例 (カラー)

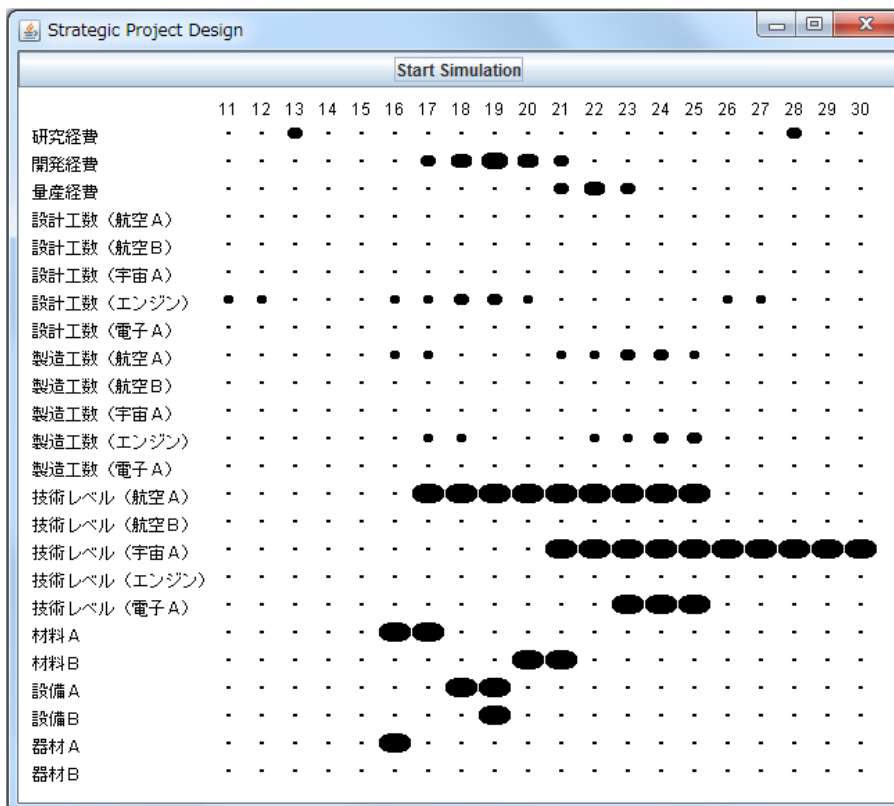


図 6.3b SPD ツールの画面例 (白黒)



## 6.3 ユーザーズ・マニュアル

システムの使用方法について述べる。まず始めに、データベースソフトにデータを入力して CSV 形式でデータを保存する。次に、SPD ツールを起動して [Start Simulation] のボタンをマウスでクリックすれば、画面にシミュレーション結果が表示される。データベースソフトのデータを修正した場合は、CSV 形式で再度データを保存し、SPD ツールの [Start Simulation] のボタンを再度クリックすればシミュレーション結果の画面が更新される。データベースソフトに入力する主要なデータを表 6 に示す。詳細な使用方法については、別添のユーザーズ・マニュアルに記述する。

表 6 データベースソフトに入力する主要なデータ

データ項目		説明
シミュレーション条件データ	表示形式	シミュレーション結果の表示形式を選択する。 「カラー」「白黒」を選択可能
	開始年度	シミュレーションを開始する年度を入力する。
	シミュレーション年数	シミュレーションを実施する年数を入力する。
	戦略目標データ項目数	戦略目標データの項目数を入力する。 Excel のテンプレートを使用する場合は、自動で計算するため入力不要である。
	制約条件データ項目数	制約条件データの項目数を入力する。 Excel のテンプレートを使用する場合は、自動で計算するため入力不要である。
	リスクケースデータ項目数	リスクケースデータの項目数を入力する。 Excel のテンプレートを使用する場合は、自動で計算するため入力不要である。
	プロジェクト数	プロジェクト数を入力する。 Excel のテンプレートを使用する場合は、自動で計算するため入力不要である。
戦略目標データ	項目	戦略目標データとして設定する項目を入力する。
	プロジェクト群	プロジェクト群の計算方法を入力する。 「合計」「最大」が選択可能
	評価方法	戦略目標とプロジェクト群の整合性の評価方法を入力する。 「以上」「以下」が選択可能
	戦略目標データ	定量化した戦略目標データをシミュレーション年数分だけデータを入力する。

制約条件データ	項目	制約条件データとして設定する項目を入力する。
	プロジェクト群	プロジェクト群の計算方法を入力する。 「合計」「最大」が選択可能
	評価方法	制約条件とプロジェクト群の整合性の評価方法を入力する。 「以上」「以下」が選択可能
	制約条件データ	定量化した制約条件データをシミュレーション年数分だけデータを入力する。
リスクケースデータ	プロジェクトタイプ	リスクケースを適用したいプロジェクトのタイプを入力する。 プロジェクトデータのプロジェクトタイプと合致している必要がある。
	項目	リスクケースを適用したい項目を入力する。 プロジェクトデータの項目と合致している必要がある。
	補正方法	リスクケースを適用した場合の補正方法を入力する。 「乗算」「加算」「減算」が選択可能
	リスクケースデータ	リスクケース1～5の補正係数を入力する。
プロジェクトデータ	プロジェクト名	プロジェクト名を入力する。各プロジェクトデータの区切りを識別するために使用している。
	プロジェクトタイプ	プロジェクトのタイプ
	項目	プロジェクトデータの項目を入力する。 戦略目標データ又は制約条件データのいずれかの項目と合致している必要がある。
	プロジェクトデータ	定量化したプロジェクトデータをシミュレーション年数分だけデータを入力する。

## 7 Verification&Validation

### 7.1 Verification&Validation の方針

本研究で提案するシステムの妥当性をシステムデザインのVモデルに従い Validation 及び Verification の2段階で検証する。

Verification については、機能確認用データを使用しモデルが正しくに機能するか評価する。Validation については、システムをユーザーの視点で使用し、ユーザーの要求を満足する正しいシステムが構築されているか評価する。

### 7.2 Verification(システムデザイン)

システムデザインの Verification は、システムデザインへの要求であるアーキテクチャ設計結果に対し、サブシステムが要求を満足することを確認する。

本研究では、データソフト、データサーバー及び SPD ツールの各サブシステムに対し、Verification を行った。また、インテグレーション計画に基づき、データソフトとデータサーバーを統合したデータベースとしての Verification を行った。Verification の結果を表 7.1 に示す。Verification の結果、システムデザインのレベルでは、システムが正しいことを確認した。

表 7.1 Verification(システムデザイン)

サブシステム	アーキテクチャ設計結果	検証内容	検証結果
データソフト	IF1.0 ユーザーからのデータ入力	キーボードとマウスを使用しデータを入力できることを確認	○
	IF2.0 データサーバーへのデータ出力	CSV 形式でローカルディスクにデータを保存できることを確認	○
データサーバー	IF2.0 データベースソフトからのデータ入力	機能確認用 CSV 形式のファイルを保存できることを確認	○
	IF3.0 SPD ツールへのデータ出力	機能確認用 CSV 形式のファイルが読み取れることを確認	○
SPD ツール	IF3.0 データサーバーからのデータ入力	機能確認用 CSV 形式のファイルを読み取れることを確認	○
	IF4.0 シミュレーション開始指示 IF5.0 シミュレーション結果表示	マウスを使用しシミュレーションの開始を指示し、結果をディスプレイに表示することを確認	○
データベース	IF1.0 ユーザーからのデータ入力 IF3.0 SPD ツールへのデータ出力	キーボードとマウスを使用しデータを入力し、データサーバーに保存されている CSV 形式のファイルが読み取れることを確認	○

### 7.3 Verification(アーキテクティング)

すべてのサブシステムを統合したのち、システムとしてアーキテクティングの Verification を実施する。アーキテクティングの Verification は、アーキテクティングへの要求であるシステム仕様に対し、システムが要求を満足することを確認する。Verification の結果を表 7.2 に示す。Verification の結果、アーキテクティングのレベルでは、システムが正しいことを確認した。

表 7.2 Verification(アーキテクティング)

No.	システム仕様	検証内容	検証結果
S1	システムは各マネージャが使用するコンピュータで動作すること。	一般的な事務用パソコンでシステムが動作することを確認	○
D1	データベースは既存のデータベースを使用すること。	Microsoft 社の Excel と一般的なネットワークシステムを使用すること確認	○
D2	各マネージャは、データベースを共有できること。	データサーバでデータを共有できることを確認	○
D3	データベースに保存されているデータは SPD ツールで読み取り可能なデータ形式であること。	データベースに保存されているデータを SPD ツールで読み取れることを確認	○
D4	各マネージャは、データベースのシミュレーション条件データを入力、修正及び削除できること。	Excel を使用しシミュレーション条件データを入力、修正及び削除できることを確認	○
D5	各マネージャは、データベースの戦略目標データを入力、修正及び削除できること。	Excel を使用し戦略目標データを入力、修正及び削除できることを確認	○
D6	各マネージャは、データベースの制約条件データを入力、修正及び削除できること。	Excel を使用し制約条件データを入力、修正及び削除できることを確認	○
D7	各マネージャは、データベースのリスクケースデータを入力、修正及び削除できること。	Excel を使用しリスクケースデータを入力、修正及び削除できることを確認	○
D8	各マネージャは、データベースのプロジェクトデータを入力、修正及び削除できること。	Excel を使用しプロジェクトデータを入力、修正及び削除できることを確認	○
D9	各マネージャは、入力、修正及び削除したデータを保存できること	Excel を使用し入力、修正及び削除したデータを保存できることを確認	○

T1	各マネージャは、SPD ツールに対しシミュレーションの実行を指示できること。	SPD ツールを使用しシミュレーションの実行が指示できることを確認	○
T2	SPD ツールは、データベースからデータを読み込めること。	SPD ツールを使用し、データベースに保存してあるデータを読み込めることを確認	○
T3	SPD ツールは、プロジェクトの整合性をシミュレーションできること。	SPD ツールでプロジェクトの整合性をシミュレーション結果が表示され、プロジェクトの整合性が確認できることを確認	○
T4	SPD ツールは、シミュレーションの結果を表示できること。		○
T5	各マネージャは、SPD ツールの表示により、プロジェクトの整合性が確認できること。		○
T6	各マネージャは、プロジェクトの整合性を確認後、SPD ツールに対し再度シミュレーションの実行を指示できること。	SPD ツールのシミュレーション開始指示が、連続して行えることを確認	○
C2	各マネージャとは、エグゼクティブ・マネージャ、ポートフォリオ・マネージャ、プログラム・マネージャ及びプロジェクト・マネージャとする。ただし、各マネージャを補佐するスタッフが実施してもよいものとする。	一般的な事務用パソコンの操作と同様の操作でシミュレーションができることを確認	○

## 7.4 Validation

### 7.4.1 Validation の方針

Validation は、システムをユーザーの視点で使用し、ユーザー要求を満足する正しいシステムが構築されているか確認する。ユーザーの視点で検証するためには、検証条件をユーザーの使用環境に近づける必要がある。Validation の方針を次に示す。

- 大規模・複雑な産業製品のプロジェクトを対象とする。
- 可能な限り実データを使用する。
- プロジェクトで起こりえる問題を設定しシステムを使用して問題を解決する。

### 7.4.2 プロジェクトデータ

実データを使用してシミュレーションを行うためには、プロジェクトにかかる経費などのデータが必要である。防衛省装備施設本部では、装備品のライフサイクルコストを「ライフサイクルコスト管理年次報告書」としてホームページに公開している。本研究では実データとして、ホームページに公開している平成22年度ライフサイクルコスト年次報告書<sup>(18)</sup>（以下「LCC年次報告書」という。）の戦闘機(F-2)のデータを使用して研究を行う。LCC年次報告書によると、戦闘機(F-2)のライフサイクルは、研究、開発、運用維持及び能力向上等、複数のプロジェクトやその他の業務からなるポートフォリオと考えることができる。公開されているデータだけでは、内訳である各プロジェクトのデータ等が分からないため、本研究では仮定をおいてプロジェクトデータを算出する。プロジェクトデータの算出方法は表7.3のとおり。

表 7.3 本研究で仮定したプロジェクトデータ算出方法

項目	算出方法
研究経費	CBS 総括表の構想段階の契約金額
開発経費	CBS 総括表の開発段階の契約金額
量産経費	CBS 総括表の量産段階の契約金額
維持廃却費	CBS 総括表の運用・維持段階及び廃却段階の契約金額の合計
設計工数	研究経費及び開発経費の合計の4分の1 10,000円/hで換算
製造工数	研究経費及び開発経費の合計の4分の1 量産経費及び維持廃却費の合計の2分の1 10,000円/hで換算
年度展開	CBS 総括表の項目名に記述している期間で均等割り 但し、維持廃却費は前提条件に記述のある「運用期間は約30年」と量産品の取得ペースを考慮して均等割りとする。

公開されている LCC 年次報告書のデータを元に、本システムの Validation 用に仮定をおいてプロジェクトデータを算出した。算出したプロジェクトデータ図 7.1 にグラフで示す。

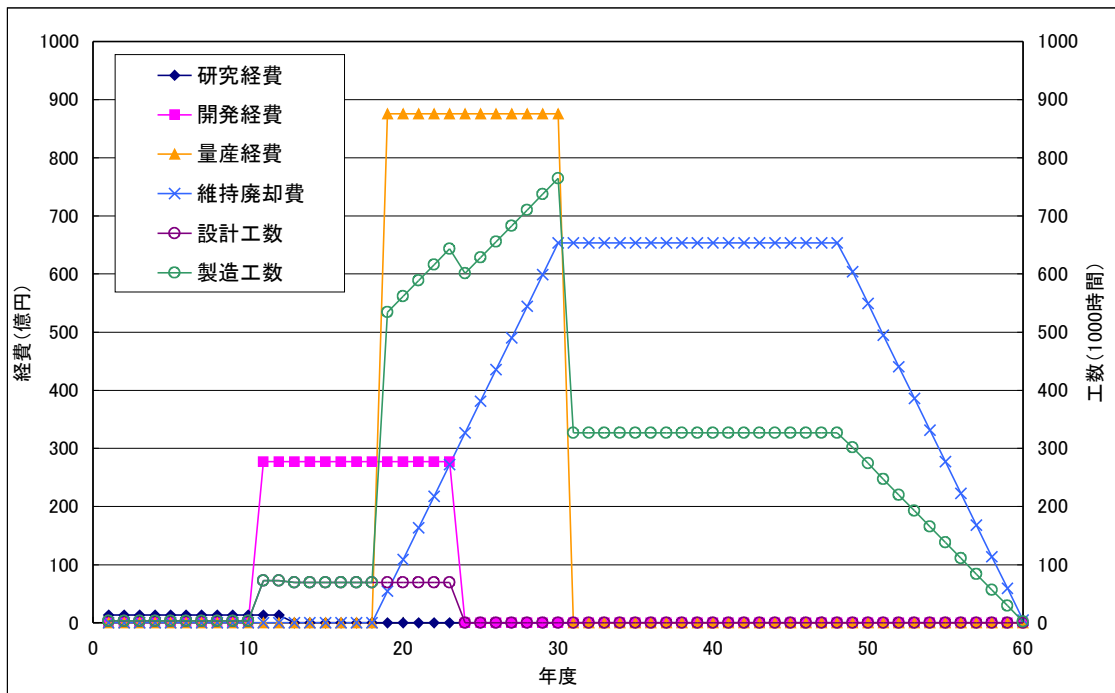


図 7.1 Validation 用プロジェクトデータ

ポートフォリオを、研究プロジェクト、開発プロジェクト及び量産プロジェクトの3つのプロジェクトに分解し、その他の業務として維持廃却を設定する。これら4つのコンポーネントをひとつのセットとして Validation を実施する。

### 7.4.3 Validation 条件の設定

Validation を行うためには、戦略目標データ、制約条件データ及びリスクデータをどのように設定するのか、プロジェクトデータをどのように使用するのかを定める必要がある。Validation 用に設定した条件を表 7.4 に示す。表中のポートフォリオ周期とは、研究プロジェクト、開発プロジェクト、量産プロジェクト及び維持廃却の4つのコンポーネントをひとつのセットとして、どの様なタイミングで着手するのかを示している。

表 7.4 Validation 条件

データ種類	項目	設定
シミュレーション条件	シミュレーション期間	2011 年度から 2070 年度の 60 年間
戦略目標	研究経費	100(億円/年)以下
	開発経費	300(億円/年)以下
	量産経費	800(億円/年)以下
	維持廃却費	1,000(億円/年)以下
	設計工数(維持)	10,000(時間/年)以上
	製造工数(維持)	100,000(時間/年)以上
制約条件	設計工数(最大)	50,000(時間/年)以下
	製造工数(最大)	500,000(時間/年)以下
リスクケース	研究経費	リスクケース 1 : 1.0 リスクケース 2 : 1.1 リスクケース 3 : 1.2 リスクケース 4 : 1.3 リスクケース 5 : 1.5
	開発経費	リスクケース 1 : 1.0 リスクケース 2 : 1.2 リスクケース 3 : 1.4 リスクケース 4 : 1.6 リスクケース 5 : 1.8
	量産経費	リスクケース 1 : 1.0 リスクケース 2 : 1.1 リスクケース 3 : 1.2 リスクケース 4 : 1.3 リスクケース 5 : 1.5
プロジェクト	ポートフォリオ周期	30 年に 1 回着手(1981 年, 2011 年, 2041 年)



## 7.4.4 シナリオの設定

ユーザー視点により、システムがユーザー要求を満足することを検証するため、ユーザーがシステムを実際に使うシナリオを設定し検証を行う。シナリオとそれに対応するユーザー要求の対応表を表 7.5 に示す。

表 7.5 シナリオとユーザー要求の対応

No.	シナリオ	ユーザー要求
1	プロジェクトがどのポートフォリオやプログラムに属しているか考慮する必要がないことを確認する。	複数のポートフォリオやプログラムに属するプロジェクトをマネジメントできること。
2	SPD ツールのシミュレーションを実施することにより、戦略目標及び制約条件とプロジェクトの整合性が確認できることを確認する。	様々なツールや技法を駆使することなく、戦略目標とプロジェクトの整合性やプロジェクト間の整合性を確保できること。
3	戦略目標とプロジェクトの不整合が、プロジェクトの計画を見直すことにより、整合性を取れることを確認する。	
4	プロジェクトが1年遅延した場合、他のマネージャはその状況を瞬時に把握することができ、それを踏まえてプロジェクトの整合性を検討できることを確認する。	プロジェクト間で必要な情報を伝達できること。 プロジェクトの計画変更の影響が全てのマネージャが瞬時に判断できること。

## 7.4.5 Validation

### 7.4.5.1 シナリオ1

表 7.5 で設定したシナリオ1（プロジェクトがどのポートフォリオやプログラムに属しているか考慮する必要がないことを確認する。）の Validation を実施する。

Validation 条件に基づき、シミュレーション条件データ、戦略目標データ、制約条件データ、リスクケースデータ及びプロジェクトデータをデータベースソフトに入力する。SPD ツールのシミュレーションの開始ボタンを押して、シミュレーション結果が表示されることを確認した。プロジェクトはポートフォリオ及びプログラムに属しているが、プロジェクトデータは SPD ツールでプロジェクト群データとして計算され戦略目標及び制約条件と整合性を評価されるため、ユーザーの視点から、プロジェクトがどのポートフォリオやプログラムに属しているか考慮することなくシミュレーションを整合性を確認することができる。実際にシステムを使用し、シナリオ1を実施できることを確認した。

## 7.4.5.2 シナリオ2

表 7.5 で設定したシナリオ2（SPD ツールのシミュレーションを実施することにより、戦略目標及び制約条件とプロジェクトの整合性が確認できることを確認する。）の Validation を実施する。

シナリオ1 で入力したデータを使用し SPD ツールでシミュレーションを実施する。シミュレーション結果のうち、2011 年度～2040 年度の結果を図 7.2 に示す。

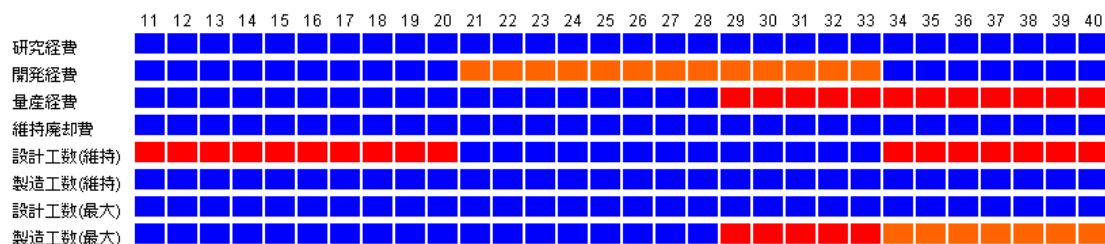


図 7.2 シミュレーション結果（シナリオ2）

図 7.2 の横軸は、見易さのため西暦の下二桁を表示させている。シミュレーションの結果により、戦略目標及び制約条件とプロジェクトの整合性が次のとおり確認できる。

- 研究経費は戦略目標を満足する。
- 開発経費が 2021～2033 年度で戦略目標を超過するリスクが高い。
- 量産経費が 2029 年度以降は戦略目標を完全に超過している。
- 維持廃却経費は戦略目標を満足する。
- 設計工数(維持)が 2011～2020 年度と 2034 年度以降で戦略目標を維持できていない。
- 製造工数(維持)は戦略目標を満足する。
- 設計工数(最大)は制約条件を満足する。
- 製造工数(最大)が 2029～2033 年度で制約条件を満足しない。

SPD シミュレーションを実施することにより、戦略目標及び制約条件とプロジェクトの整合性が確認できた。実際にシステムを使用し、シナリオ2 を実施できることを確認した。

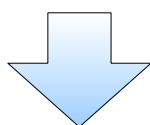
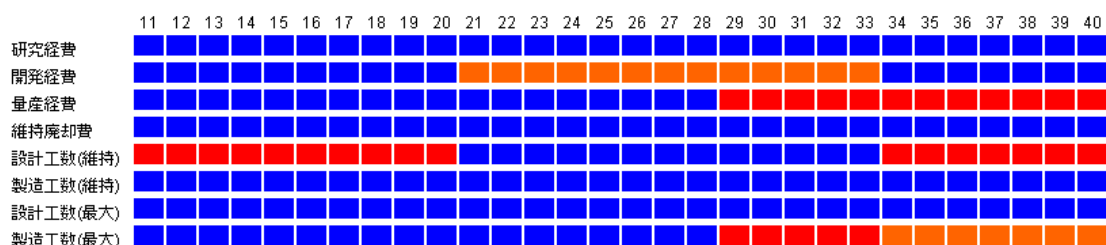
### 7.4.5.3 シナリオ3

表 7.5 で設定したシナリオ 3（戦略目標とプロジェクトの不整合が、プロジェクトの計画を見直すことにより、整合性を取れることを確認する。）の Validation を実施する。

シナリオ 2 で不整合となっている状況を踏まえ、プロジェクトの修正を行う。修正する内容は次のとおり。

- 開発時の設計の一部を研究で先行着手し、開発経費と設計工数の平準化を行う。
- 量産の取得ペースを落とし、取得経費と製造工数の平準化を行う。
- 量産着手後、能力向上のための改善事業を追加し、設計工数の維持を行う。

データベースソフトを使用しプロジェクトデータを上記の様に修正する。データ修正後 SPD ツールで再度シミュレーションを実施する。シミュレーション結果のうち、2011 年度～2040 年度の結果を図 7.3 に示す。上段が修正前の結果であり、下段が修正後の結果である。



- 開発時の設計の一部を研究で先行着手
- 量産の取得ペースを減少
- 量産着手後に能力向上事業を追加

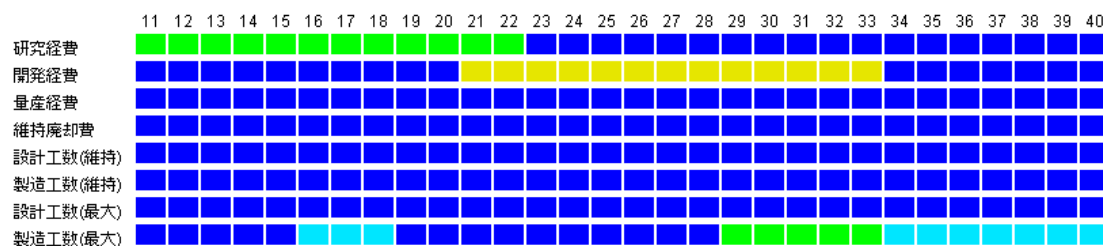


図 7.3 シミュレーション結果（シナリオ 3）

開発経費において依然としてリスクが残っているが、全体としてプロジェクトの整合性を改善することができた。開発経費はリスクケースの係数が高いため、リスクを完全に排除することができなかった。開発時についてはリスクマネジメントをしっかり行うことが重要であることが、シミュレーションの結果から確認できた。実際にシステムを使用し、シナリオ 3 を実施できることを確認した。

#### 7.4.5.4 シナリオ4

表 7.5 で設定したケース 4（プロジェクトが 1 年遅延した場合、他のマネージャはその状況を瞬時に把握することができ、それを踏まえてプロジェクトの整合性を検討できることを確認する。）の Validation を実施する。

シナリオ 3 で得られた結果に対し、開発プロジェクトが 1 年遅延した場合を想定する。データベースソフトを使用し開発期間が 1 年延期するようプロジェクトデータを修正する。データ修正後、SPD ツールで再度シミュレーションを実施する。シミュレーション結果のうち、2011 年度～2040 年度の結果を図 7.4 に示す。上段が修正前の結果であり、下段が修正後の結果である。

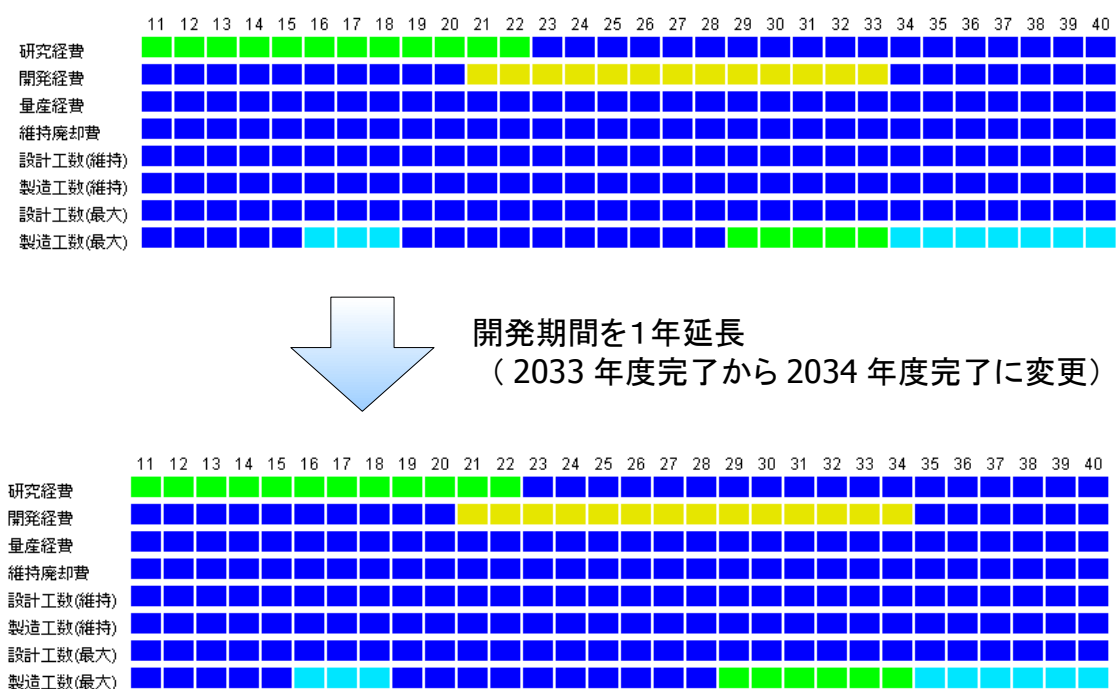


図 7.4 シミュレーション結果（シナリオ 4）

2034 年度の開発経費が増大し、戦略目標を超過するリスクが高まったことがわかる。修正したプロジェクトデータは各マネージャで共有できるため、この状況を踏まえて各マネージャはプロジェクトの整合性を検討することが可能である。実際にシステムを使用し、シナリオ 4 を実施できることを確認した。

## 7.4.6 Validation 結果

各シナリオに対する検証結果を表 7.6 にまとめる。実際にシステムを使用し、すべてのシナリオが実施できることを確認した。

表 7.6 Validation 結果

シナリオ	検証結果
プロジェクトがどのポートフォリオやプログラムに属しているか考慮する必要がないことを確認する。	○
SPD ツールのシミュレーションを実施することにより、戦略目標及び制約条件とプロジェクトの整合性が確認できることを確認する。	○
戦略目標とプロジェクトの不整合が、プロジェクトの計画を見直すことにより、整合性を取れることを確認する。	○
プロジェクトが1年遅延した場合、他のマネージャはその状況を瞬時に把握することができ、それを踏まえてプロジェクトの整合性を検討できることを確認する。	○

Validation の結果より、本研究で提案したシステムを運用することにより、先行研究の問題点を解決し、ユーザーの要求を満足することを確認できた。

## 8 結論

本研究では、複雑化した膨大なプロジェクトを抱える組織において、戦略目標を達成するプロジェクトのデザイン方法を提案した。

戦略目標を達成することを目的とするためには、プロジェクトを戦略的な視点から捉えデザインすることが重要である。本研究では、階層構造のマネジメントをとらず複数のプロジェクトをプロジェクト群としてとらえ、戦略目標とプロジェクトを直接比較して評価する手法を採用した。これにより、戦略目標とプロジェクトの整合性及びプロジェクト間の整合性を取ることが可能となり、階層構造に起因する従来手法の問題を解決した。一方、この手法は、プロジェクト群に関する膨大な情報を取り扱わなくてはならないといった問題が生じるが、プロジェクト群に関する情報を定量化しコンピュータによるシミュレーションを活用することにより、その問題を解決した。

また、システムを運用するためには、システムがその機能を保有しているだけでなく、ユーザーがシステムを使用できることが重要である。本研究では、シミュレーション結果を可視化し各マネージャが簡単に使えるシステムとすることにより、プロジェクト群の状況を迅速に把握するシステムを提供した。

## 9 参考文献

- (1) Project Management Institute, A Guide To The Project Management Body Of Knowledge 4th, Project Management Institute, 2008
- (2) Project Management Institute, The Standard for Program Management Second Edition, Project Management Institute, 2009
- (3) Project Management Institute, The Standard for Portfolio Management Second Edition, Project Management Institute, 2008
- (4) 日本プロジェクトマネジメント協会, P2M プロジェクト&プログラムマネジメント標準ガイドブック, 日本能率協会マネジメントセンター, 2007
- (5) 経済産業省、文部科学省及び厚生労働省, 2010年版ものづくり白書, 経済産業調査会, 2010, pp25
- (6) Kevin Forsberg, Hal Mooz and Howard Cotterman, Visualizing Project Management Third Edition, Wiley, 2005
- (7) 当麻哲哉, プロジェクトマネジメント第1回, 慶応義塾大学システムデザインマネジメント研究科プロジェクトマネジメント講義資料, 2009, pp.8
- (8) Dragan Z. Milosevic, Project Management Tool Box: Tools and Techniques for the Practicing Project Manager, Project Management Institute Inc, Wiley, 2003
- (9) United States Department of Defense, Systems Engineering Fundamentals, Defense Acquisition University Press, 2001  
<http://www.dau.mil/pubscats/PubsCats/SEFGuide%2001-01.pdf>
- (10) 石井浩介及び飯野謙次, 設計の科学 価値づくり設計, 養賢堂, 2008
- (11) Kurt Beiter and Whit Fowler, Pugh Concept Selection, 慶応義塾大学システムデザインマネジメント研究科ALPS第2回論講義資料, 2009
- (12) 赤木新介, システム工学 エンジニアリングシステムの解析と計画, 共立出版, 1992
- (13) Steward D., The Design Structure System: A Method for Managing the Design of Complex Systems, IEEE Transaction on Engineering Management vol.28, 1981, pp.71-74
- (14) John C. Mankins, Technology Readiness Levels, A White Paper, 1995  
<http://ehbs.org/trl/Mankins1995.pdf>
- (15) 長井健一郎, 宇宙航空研究開発機構における研究開発評価とマネジメント 文部科学省研究開発ワークショップ, 2006  
[http://www.mext.go.jp/a\\_menu/kagaku/hyouka/06122004/002.pdf](http://www.mext.go.jp/a_menu/kagaku/hyouka/06122004/002.pdf)
- (16) National Institute of Standards and Technology, Announcing the Standard for Integration Definition For Information Modeling(IDEFIX), 1993  
<http://www.itl.nist.gov/fipspubs/idef1x.doc>
- (17) 前野隆司, システムアーキテクティングとデザイン第4回, 慶応義塾大学システムデザインマネジメント研究科システムアーキテクティングとデザイン講義資料, 2009
- (18) 防衛省装備施設本部, 平成22年度ライフサイクルコスト管理年次報告書, 2010  
<http://www.epco.mod.go.jp/about/kakusyusiryoku.html>

## 10 謝辞

本研究を進めるにあたり、温かく親切丁寧なご指導、ご助言を賜りました指導教員の狼嘉彰 教授に、心より感謝申し上げます。また、副査としてご指導を頂きました、春山 真一郎 教授、当麻 哲哉 准教授に、深く感謝申し上げます。

戦略システムデザインラボ及び戦略的宇宙システムラボにおいて、日比谷 孟俊 教授、浦郷 正隆 准教授を始め、多くの方々にご助言を頂きました。深く感謝申し上げます。

システムデザイン・マネジメント研究科への入学は、手嶋 龍一 教授、防衛省航空幕僚監部防衛部装備体系課の方々の、強い思いと多大なる支援で実現しました。深く感謝申し上げます。

最後に、本研究科の在籍期間中、影から研究を支えて頂きました家族に、感謝いたします。



# 11 付録

## 11.1 ユーザーズ・マニュアル

### 11.1.1 データベースソフト

データベースソフトのデータ入力方法を説明する。

#### ●データ識別コード

データの種別を識別するため、A列にデータ識別コードを入力する。データ識別コードの画面例を図11.1に示す。図11.1の赤枠で囲った部分がデータ識別コードである。データ識別コードとデータ種類の対応は表11.1のとおり。

	A	B	C	D	E	F	G	H
1	C							
2	C	戦略的システムデザイン データベースソフト(データ入力シート)						
3	C							
4	C	シミュレーション条件データ						
5	C	表示形式 カラー:0,白黒:1	開始年度	シミュレーション 年数	戦略目標 データ項目数	制約条件 データ項目数	リスクース データ項目数	プロジェクト 数
6	BD	0	2011	15	13	11	0	2
7	C							
8	C	戦略目標データ						
9	C	項目	プロジェクト 群	評価方法				
10	C				2011	2012	2013	2014
11	SD	研究経費	合計	以下				
12	SD	開発経費	合計	以下				
13	SD	生産経費	合計	以下				
14	SD	設計工数(航空A)	合計	以上				
15	SD	設計工数(航空B)	合計	以上				
16	SD	設計工数(宇宙A)	合計	以上				
17	SD	設計工数(エンジン)	合計	以上				
18	SD	設計工数(電子A)	合計	以上				
19	SD	製造工数(航空A)	合計	以上				
20	SD	製造工数(航空B)	合計	以上				
21	SD	製造工数(宇宙A)	合計	以上				
22	SD	製造工数(エンジン)	合計	以上				
23	SD	製造工数(電子A)	合計	以上				
24	C							
25	C	制約条件データ						
26	C	項目	プロジェクト 群	評価方法				
27	C				2011	2012	2013	2014
28	LD	技術レベル(航空A)	最大	以下				
29	LD	技術レベル(航空B)	最大	以下				
30	LD	技術レベル(宇宙A)	最大	以下				
31	LD	技術レベル(エンジン)	最大	以下				
32	LD	技術レベル(電子A)	最大	以下				
33	LD	材料A	合計	以下				
34	LD	材料B	合計	以下				
35	LD	設備A	合計	以下				
36	LD	設備B	合計	以下				

図11.1 データ識別コードの画面例

表 11.1 データ識別コードとデータ種類の対応表

データ識別コード	データ種類
BD	シミュレーション条件データのデータ行
SD	戦略目標データのデータ行
LD	制約条件データのデータ行
RD	リスクケースデータのデータ行
PN	プロジェクトデータのプロジェクト名
PT	プロジェクトデータのプロジェクトタイプ
PD	プロジェクトデータのデータ行
C 又はその他のコード	コメント行 (処理しない)

●シミュレーション条件データ

データ識別コード「BD」に引き続き、「表示形式」、「開始年度」、「シミュレーション年数」、「戦略目標データ項目数」、「制約条件データ項目数」、「リスクケースデータ項目数」、「プロジェクト数」の順番で、整数で入力する。表示形式は、「0」を入力した場合カラー表示、「1」を入力した場合は白黒表示となる。「開始年度」はシミュレーション結果の X 軸の表示に使用しているだけであるため、四半期や月単位を整数で入力してもよい。その場合、戦略目標データ、制約条件データ及びプロジェクトデータも四半期又は月単位で入力する。また、「シミュレーション年数」については、時間軸に展開したデータを何回分までシミュレーションするかを示すものであり、四半期や月単位のシミュレーションを行う場合は、「シミュレーション四半期数」や「シミュレーション月数」と読み替えることができる。Excel のテンプレートを使用する場合は、「戦略目標データ項目数」、「制約条件データ項目数」、「リスクケースデータ項目数」及び「プロジェクト数」は自動で計算されるため入力は不要である。シミュレーション条件データの画面例を図 11.2 に示す。

C	シミュレーション条件データ						
C	表示形式 カラー:0,白黒:1	開始年度	シミュレーション 年数	戦略目標 データ項目数	制約条件 データ項目数	リスクケース データ項目数	プロジェクト 数
BD	0	2011	15	13	11	0	2

図 11.2 シミュレーション条件データの画面例

●戦略目標データ

データ識別コード「SD」に引き続き、「項目」、「プロジェクト群」、「評価方法」、「戦略目標データ」の順番で入力する。「戦略目標データ」はシミュレーション年数分だけ、数値で入力する。データ識別コード「SD」の行は複数設定することができる。戦略目標データの画面例を図 11.3 に示す。

○	戦略目標データ						
○	項目	プロジェクト群	評価方法	2011			
○				2012	2013	2014	
SD	研究経費	合計	以下				
SD	開発経費	合計	以下				
SD	運産経費	合計	以下				
SD	設計工数(航空A)	合計	以上				

図 11.3 戦略目標データの画面例

●制約条件データ

データ識別コード「LD」に引き続き、「項目」、「プロジェクト群」、「評価方法」、「戦略目標データ」の順番で入力する。「制約条件データ」はシミュレーション年数分だけ、数値で入力する。データ識別コード「LD」の行は複数設定することができる。制約条件データの画面例を図 11.4 に示す。

○	制約条件データ						
○	項目	プロジェクト群	評価方法	2011			
○				2012	2013	2014	
LD	技術レベル(航空A)	最大	以下				
LD	技術レベル(航空B)	最大	以下				
LD	技術レベル(宇宙A)	最大	以下				
LD	技術レベル(エンジン)	最大	以下				

図 11.4 制約条件データの画面例

●リスクケースデータ

データ識別コード「RD」に引き続き、「プロジェクトタイプ」、「項目」、「補正方法」、「リスクケースデータ」の順番で入力する。「リスクケースデータ」はリスクケース 1～5 の 5 ケースの補正係数を、数値で入力する。データ識別コード「RD」の行は複数設定することができる。リスクケースデータの画面例を図 11.5 に示す。

○	リスクケースデータ						
○	プロジェクトタイプ	項目	補正方法	リスクケース			
○				ケース1	ケース2	ケース3	ケース4
RD	研究航空機	研究経費	乗算				
RD	開発エンジン	開発経費	乗算				

図 11.5 リスクケースデータの画面例

●プロジェクトデータ

プロジェクトデータは、プロジェクト単位のブロックで管理されている。このブロックを識別するため、各プロジェクトデータの先頭行に、データ識別コード「PN」の行を入力する。データ識別コード「PN」の行については、データ識別コード「PN」に引き続き、「プロジェクト名」を入力する。

データ識別コード「PN」の次の行に、データ識別コード「PT」の行を入力する。データ識別コード「PT」の行については、データ識別コード「PT」に引き続き、「プロジェクトタイプ」を入力する。

データ識別コード「PT」の次の行に、データ識別コード「PD」の行を入力する。データ識別コード「PD」の行については、データ識別コード「PD」に引き続き、「項目」、「未使用」、「未使用」、「項目」、「未使用」、「未使用」、「2011」、「2012」、「2013」、「2014」を入力する。「未使用」については、戦略目標データ及び制約条件データと、年度の縦軸をあわせるために設定している項目であり、データの入力はいらない。「プロジェクトデータ」はシミュレーション年数分だけ、数値で入力する。データ識別コード「PT」の行は複数設定することができる。プロジェクトデータの画面例を図 11.6 に示す。

0	プロジェクトデータ							
PN	航空機の要素技術研究							
PT	研究航空機							
0	項目	未使用	未使用					
0				2011	2012	2013	2014	
PD	研究経費							
PD	設計工数(航空A)							
PD	製造工数(航空A)							
PD	技術レベル(航空A)							
PD	材料A							
PD	設備B							
PD	器材A							
0								
PN	エンジンの開発							
PT	開発エンジン							
0	項目	未使用	未使用					
0				2011	2012	2013	2014	
PD	開発経費							
PD	設計工数(エンジン)							
PD	製造工数(エンジン)							
PD	技術レベル(エンジン)							
PD	材料A							
PD	設備B							
PD	器材B							

図 11.6 プロジェクトデータの画面例

## 11.1.2 SPD ツール

SPD ツールの使用方法を説明する。

### ●操作方法と画面例

[ Start Simulation ] のボタンをマウスでクリックすると、データベースのデータを読み込み、ウィンドウ内にシミュレーション結果を表す。[ Start Simulation ] のボタンをクリックするたびにデータベースのデータを読み込むため、データベースソフトでデータを修正した場合は、新しいシミュレーション結果が表示される。カラーの SPD ツールの画面例を図 11.7 に示す。

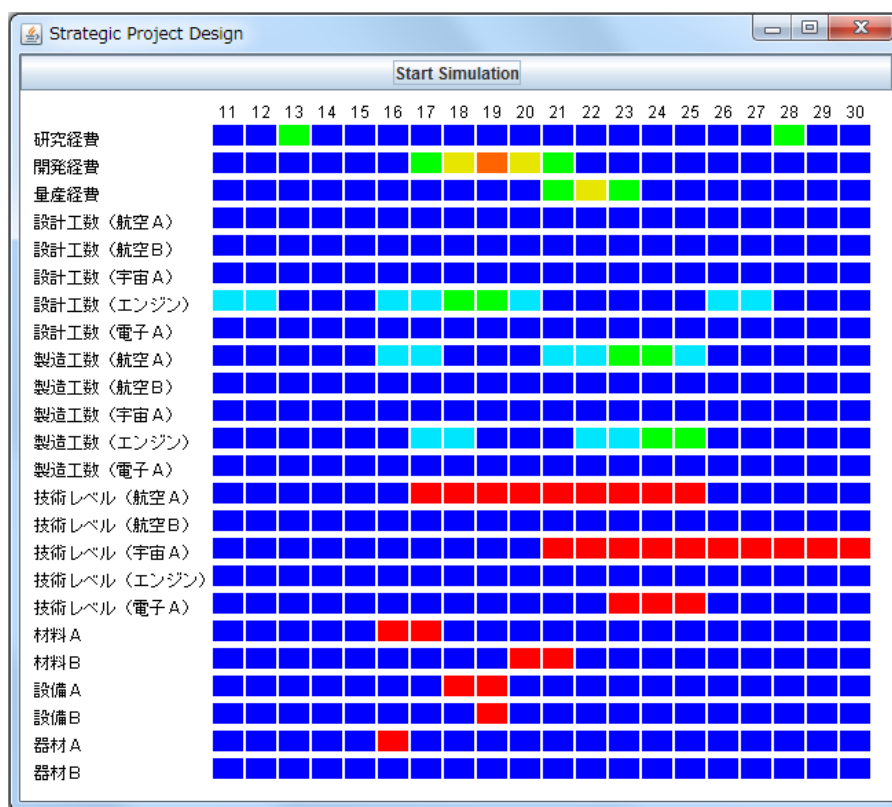


図 11.7 SPD ツールの画面例 (カラー)

ウィンドウ上部にあるボタンが [ Start Simulation ] のボタンである。縦軸は戦略目標データの項目と制約条件データの項目、横軸は西暦下二桁の年度を表している。画面中央の四角のマトリックスの色が戦略目標データ及び制約条件データとプロジェクト群との整合性を表している。マトリックスの色とリスクケースの対応を表 11.2 に示す。

表 11.2 マトリックスの色とリスクケースの対応

マトリックスの色	リスクケース
青色	すべてのリスクケースで整合性が取れている。
水色	1つのリスクケースで整合性が取れていない。
緑色	2つのリスクケースで整合性が取れていない。
黄色	3つのリスクケースで整合性が取れていない。
橙色	4つのリスクケースで整合性が取れていない。
赤色	すべてのリスクケースで整合性が取れていない。

●データベースのファイルの読み込み

データベースのファイルについては、Java の workspace 内の SPD.csv ファイルを読み込むように設定してある。別のファイルを読み込むためには、ソースコードの設定を修正する必要がある。また、シミュレーション条件データ、戦略目標データ、制約条件データ、リスクケースデータ及びプロジェクトデータは、それぞれ別のファイルを読み込むことも可能であるが、設定を変更するためにはソースコードの設定を変更する必要がある。

## 11.2 SPD ツールのプログラム

```
/*
 * Strategic Project Design Program
 *
 * Copyright (C) 2011 Tokukazu Ohmine All Rights Reserved.
 */

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.StringTokenizer;

class MainFrame extends JFrame {
    private JPanel p;
    public MainFrame() {
        setTitle("Strategic Project Design");
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        p=new JPanel();
        p.setBackground(Color.WHITE);

        // シミュレーションプログラム
        JButton b1 = new JButton("Start Simulation");
        this.add(b1, BorderLayout.NORTH);
        b1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ev) {
                boolean debug = true;           // デバッグモード制御 : true:通常モード, false:デバッグモード
                boolean spiralFlag = true;      // プログラムモード選択 : true:スパイラル2, false:スパイラル1
                int display = 0;                // 画面出力形式
                int i=0, j=0, k=0, n=0, r=0;
                int yearStart=0;                // シミュレーション開始年度
                int yearTime=0;                 // シミュレーション実施期間
                int itemStrategic=0;            // 戦略データの項目数
                int itemLimit=0;                // 制約条件データの項目数
                int itemAll=0;                  // 戦略と制約条件の項目数の合計
                int itemExp=0;                  // 経験データの項目数
                int projectNo=0;                // プロジェクト数
                float [][] strategicData;       // 戦略データ
                float [][] limitData;           // 制約条件データ
                float [][] [] projectData;      // プロジェクトデータ
                float [][] [] projectGroupData; // プロジェクトグループデータ
                String [][] expData;            // 経験データ
                int [][] evaluationResult;       // 評価結果
                String [] itemName;             // 項目名
                String [] itemCalcType;         // 項目別プロジェクト群計算方法
                String [] itemCompType;         // 項目別評価方法
                String dataType;
                String basicDataFileName;       // 基本データの読み込みファイル名
                String strategicDataFileNme;    // 戦略データの読み込みファイル名
                String limitDataFileName;       // 制約条件データの読み込みファイル名
                String experienceDataFileName;  // 経験データの読み込みファイル名
                String projectDataFileName;     // プロジェクトデータの読み込みファイル名
            }
        });
    }
}
```

```

// データ読み込みファイル名設定（各データのファイルは別々に設定することも可能）
basicDataFileName = "SPD.csv";
strategicDataFlieName = "SPD.csv";
limitDataFileName = "SPD.csv";
experienceDataFileName = "SPD.csv";
projectDataFileName = "SPD.csv";

// シミュレーション条件データ読み込み
try {
    File csv = new File(basicDataFileName);
    BufferedReader br = new BufferedReader(new FileReader(csv));
    String line = "";
    while((line = br.readLine())!=null) {
        if (line.startsWith("BD")) {
            break;
        }
    }
}
// データ読み込み
StringTokenizer st = new StringTokenizer(line, ",");
dataType = st.nextToken();
display = Integer.parseInt(st.nextToken());
yearStart = Integer.parseInt(st.nextToken());
yearTime = Integer.parseInt(st.nextToken());
itemStrategic = Integer.parseInt(st.nextToken());
itemLimit = Integer.parseInt(st.nextToken());
itemExp = Integer.parseInt(st.nextToken());
projectNo = Integer.parseInt(st.nextToken());
br.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

// 配列初期設定
itemAll = itemStrategic+itemLimit;
itemName = new String[itemAll];
itemCalcType = new String[itemAll];
itemCompType = new String[itemAll];
projectData = new float[projectNo][itemAll][yearTime][5]; // リスクケースは5段階評価
projectGroupData = new float[itemAll][yearTime][5]; // リスクケースは5段階評価
evaluationResult = new int[itemAll][yearTime];

// 戦略データ読み込み
strategicData = new float[itemStrategic][yearTime];
try {
    File csv = new File(strategicDataFlieName);
    BufferedReader br = new BufferedReader(new FileReader(csv));
    String line = "";
    for (i=0;i<itemStrategic;i++) {
        while((line = br.readLine())!=null) {
            if (line.startsWith("SD")) {
                break;
            }
        }
    }
    StringTokenizer st = new StringTokenizer(line, ",");
    dataType = st.nextToken();
    itemName[i] = st.nextToken();
    itemCalcType[i] = st.nextToken();
    itemCompType[i] = st.nextToken();
    for (j=0;j<yearTime;j++) {
        strategicData[i][j] = Float.parseFloat(st.nextToken());
    }
}

```



```

    }
}
br.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

// 制約条件データ読み込み
limitData = new float[itemLimit][yearTime];
try {
    File csv = new File(limitDataFileName);
    BufferedReader br = new BufferedReader(new FileReader(csv));
    String line = "";
    for (i=0;i<itemLimit;i++) {
        while((line = br.readLine())!=null) {
            if (line.startsWith("LD")) {
                break;
            }
        }
        StringTokenizer st = new StringTokenizer(line, ",");
        dataType = st.nextToken();
        itemName[itemStrategic+i] = st.nextToken();
        itemCalcType[itemStrategic+i] = st.nextToken();
        itemCompType[itemStrategic+i] = st.nextToken();
        for (j=0;j<yearTime;j++) {
            limitData[i][j] = Float.parseFloat(st.nextToken());
        }
    }
    br.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

// リスクケースデータ読み込み
expData = new String[itemExp][8];
try {
    File csv = new File(experienceDataFileName);
    BufferedReader br = new BufferedReader(new FileReader(csv));
    String line = "";
    for (i=0;i<itemExp;i++) {
        while((line = br.readLine())!=null) {
            if (line.startsWith("RD")) {
                break;
            }
        }
        StringTokenizer st = new StringTokenizer(line, ",");
        dataType = st.nextToken();
        for (j=0;j<8;j++) {
            expData[i][j] = st.nextToken();
        }
    }
    br.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

```

```

// プロジェクトデータ読み込み
try {
    File csv = new File(projectDataFileName);
    BufferedReader br = new BufferedReader(new FileReader(csv));
    String line = "";
    String projectType = "";
    String itemNameProject = "";

    k=-1;
    while((line = br.readLine())!=null) {
        if (line.startsWith("PN")) {
            k=k+1;
            i=0;
            continue;
        }
        else if (line.startsWith("PT")) {
            StringTokenizer st = new StringTokenizer(line, ",");
            dataType = st.nextToken();
            projectType = st.nextToken();
            continue;
        }
        else if (line.startsWith("PD")) {
            StringTokenizer st = new StringTokenizer(line, ",");
            dataType = st.nextToken();
            itemNameProject = st.nextToken();
            // 戦略&制約条件項目とプロジェクト項目の一致をサーチ
            while(!(itemName[i].equals(itemNameProject))) {
                if (i+1<itemAll){
                    i=i+1;
                }
                else {
                    i=0;
                }
            }
            for (j=0;j<yearTime;j++) {
                projectData[k][i][j][0] = Float.parseFloat(st.nextToken());
                for (n=1;n<5;n++) {
                    projectData[k][i][j][n] = projectData[k][i][j][0];
                }
            }
            // 経験データ該当時の処理(スパイラル2用処理)
            if (spiralFlag) {
                for (r=0;r<itemExp;r++) {
                    if (expData[r][0].equals(projectType)&&expData[r][1].equals(itemNameProject)) {
                        if (expData[r][2].equals("乗算")) {
                            for (j=0;j<yearTime;j++) {
                                for (n=0;n<5;n++) {
                                    projectData[k][i][j][n]=projectData[k][i][j][0]*Float.parseFloat(expData[r][n+3]);
                                }
                            }
                        }
                        else if (expData[r][2].equals("加算")) {
                            for (j=0;j<yearTime;j++) {
                                for (n=0;n<5;n++) {
                                    projectData[k][i][j][n]=projectData[k][i][j][0]+Float.parseFloat(expData[r][n+3]);
                                }
                            }
                        }
                        else if (expData[r][2].equals("減算")) {
                            for (j=0;j<yearTime;j++) {
                                for (n=0;n<5;n++) {
                                    projectData[k][i][j][n]=projectData[k][i][j][0]-Float.parseFloat(expData[r][n+3]);
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    }
    // リスクケースの処理を追加する場合はここに追加
    else {
        System.out.println("リスクケース処理エラー：読込補正方法→"+expData[r][2]);
    }
    break;
}
}
}
}
br.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

// プロジェクト群データの作成
for (i=0;i<itemAll;i++) {
    if (itemCalcType[i].equals("合計")) {
        for (j=0;j<yearTime;j++) {
            for (k=0;k<projectNo;k++){
                for (n=0;n<5;n++) {
                    projectGroupData[i][j][n] = projectGroupData[i][j][n] + projectData[k][i][j][n];
                }
            }
        }
    }
    else if (itemCalcType[i].equals("最大")) {
        for (j=0;j<yearTime;j++) {
            for (k=0;k<projectNo;k++){
                for (n=0;n<5;n++) {
                    if (projectGroupData[i][j][n] < projectData[k][i][j][n]) {
                        projectGroupData[i][j][n] = projectData[k][i][j][n];
                    }
                }
            }
        }
    }
    else {
        System.out.println("プロジェクト群データ作成エラー：読込計算方法→"+itemCalcType[i]);
        System.out.println(itemCalcType[i]);
    }
}

// データの比較
// 戦略データとの比較
for (i=0;i<itemStrategic;i++) {
    for (j=0;j<yearTime;j++) {
        for (n=0;n<5;n++) {
            if (itemCompType[i].equals("以下")) {
                if (strategicData[i][j]>=projectGroupData[i][j][n]) {
                    evaluationResult[i][j]=evaluationResult[i][j]+10;
                }
                else {
                    evaluationResult[i][j]=evaluationResult[i][j]+0;
                }
            }
            else if (itemCompType[i].equals("以上")) {

```

```

        if (strategicData[i][j]<=projectGroupData[i][j][n]) {
            evaluationResult[i][j]=evaluationResult[i][j]+10;
        }
        else {
            evaluationResult[i][j]=evaluationResult[i][j]+0;
        }
    }
    else {
        System.out.println("戦略データ比較エラー：読込比較方法→"+itemCompType[i]);
        System.out.println(itemCompType[i]);
    }
}
}
}
// 制約条件データとの比較
for (i=0;i<itemLimit;i++) {
    for (j=0;j<yearTime;j++) {
        for (n=0;n<5;n++) {
            if (itemCompType[i+itemStrategic].equals("以下")) {
                if (limitData[i][j]>=projectGroupData[i+itemStrategic][j][n]) {
                    evaluationResult[i+itemStrategic][j]=evaluationResult[i+itemStrategic][j]+10;
                }
                else {
                    evaluationResult[i+itemStrategic][j]=evaluationResult[i+itemStrategic][j]+0;
                }
            }
            else if (itemCompType[i+itemStrategic].equals("以上")) {
                if (limitData[i][j]<=projectGroupData[i+itemStrategic][j][n]) {
                    evaluationResult[i+itemStrategic][j]=evaluationResult[i+itemStrategic][j]+10;
                }
                else {
                    evaluationResult[i+itemStrategic][j]=evaluationResult[i+itemStrategic][j]+0;
                }
            }
            else {
                System.out.println("制約条件データ比較エラー：読込比較方法→"+itemCompType[i+itemStrategic]);
                System.out.println(itemCompType[i+itemStrategic]);
            }
        }
    }
}
}

// シミュレーション結果の表示
Graphics g = p.getGraphics();
Font f = new Font("項目と年数",Font.BOLD,14);

// フォント情報の取得
FontMetrics fm=g.getFontMetrics(f);
int fontAscent = fm.getAscent();

// 表示の大きさ設定
int panelx = 140;
int tileStepx = 24;
int tileSizez = 22;

// 年度の描画
for (i=0;i<yearTime;i++) {
    g.drawString(Integer.toString(yearStart+i),panelx+4+ tileStepx*i,20);
}

// 項目の描画
for (i=0;i<itemAll;i++) {

```

```

    g.drawString(itemName[i], 10, 40+20*i);
}

// シミュレーション結果の描画
if (display==0) {
    for (i=0;i<itemAll;i++){
        for (j=0;j<yearTime;j++){
            if (evaluationResult[i][j]==50) {
                g.setColor(new Color(0, 0, 255));
                g.fillRect(panelx+j*tileStepx, 40-fontAscent+i*20, tileSize, fontAscent);
            }
            else if (evaluationResult[i][j]==40) {
                g.setColor(new Color(0, 230, 255));
                g.fillRect(panelx+j*tileStepx, 40-fontAscent+i*20, tileSize, fontAscent);
            }
            else if (evaluationResult[i][j]==30) {
                g.setColor(new Color(0, 255, 0));
                g.fillRect(panelx+j*tileStepx, 40-fontAscent+i*20, tileSize, fontAscent);
            }
            else if (evaluationResult[i][j]==20) {
                g.setColor(new Color(230, 230, 0));
                g.fillRect(panelx+j*tileStepx, 40-fontAscent+i*20, tileSize, fontAscent);
            }
            else if (evaluationResult[i][j]==10) {
                g.setColor(new Color(255, 100, 0));
                g.fillRect(panelx+j*tileStepx, 40-fontAscent+i*20, tileSize, fontAscent);
            }
            else {
                g.setColor(new Color(255, 0, 0));
                g.fillRect(panelx+j*tileStepx, 40-fontAscent+i*20, tileSize, fontAscent);
            }
        }
    }
}
else {
    g.setColor(new Color(0, 0, 0));
    for (i=0;i<itemAll;i++){
        for (j=0;j<yearTime;j++){
            if (evaluationResult[i][j]==50) {
                g.fillOval(panelx+j*tileStepx+9, 40-fontAscent+i*20+6, tileSize-18, fontAscent-12);
            }
            else if (evaluationResult[i][j]==40) {
                g.fillOval(panelx+j*tileStepx+7, 40-fontAscent+i*20+4, tileSize-14, fontAscent-8);
            }
            else if (evaluationResult[i][j]==30) {
                g.fillOval(panelx+j*tileStepx+5, 40-fontAscent+i*20+3, tileSize-10, fontAscent-6);
            }
            else if (evaluationResult[i][j]==20) {
                g.fillOval(panelx+j*tileStepx+3, 40-fontAscent+i*20+2, tileSize-6, fontAscent-4);
            }
            else if (evaluationResult[i][j]==10) {
                g.fillOval(panelx+j*tileStepx+1, 40-fontAscent+i*20+1, tileSize-2, fontAscent-2);
            }
            else {
                g.fillOval(panelx+j*tileStepx-1, 40-fontAscent+i*20, tileSize+2, fontAscent);
            }
        }
    }
}
g.dispose();
}

```

```
});  
  
Container c=getContentPane();  
c.add(p);  
}  
}  
  
public class StrategicProjectDesign {  
    public static void main(String[] args) {  
        JFrame w = new JFrame();  
        w.show();  
    }  
}
```