

信州大学審査学位論文

フロントローディングを目指した設計基盤構築とこれに用いる  
外部ソフトウェア入出力値群の定義・運用に関する研究

2023年3月

工藤卓史



## 目 次

---

	頁
第 1 章 緒言	1
1. 1 背景	2
1. 1. 1 設計のフロントローディングと 1DCAE	2
1. 1. 2 設計プラットフォーム	5
1. 1. 3 BOM と入出力定義	6
1. 2 研究目的	8
1. 3 論文構成	10
第 2 章 従来手法と課題	12
2. 1 活用技術	13
2. 1. 1 従来手法	13
2. 1. 2 課題	13
2. 2 設計プラットフォームのトレンド追従	15
第 3 章 課題解決の手法	17
3. 1 設計プラットフォーム要素の再構築	18
3. 2 環境技術としての設計プラットフォーム	20
3. 3 JSM を用いた設計プラットフォームのトレンド追従	25
3. 3. 1 JSM	25
3. 3. 2 機能拡張	27
3. 3. 3 JSM editor	32
3. 3 まとめ	45
第 4 章 発電用水車を対象とする実際の Virtual Engineering 設計環境	46
4. 1 発電機用水車	47
4. 1. 1 概要	47
4. 1. 2 比速度	50
4. 1. 3 部品	53

4.2	概要	54
4.2.1	フロー図	54
4.2.2	Basic Calculation	55
4.2.3	Model Selection	56
4.2.4	Dimension Modify	56
4.2.5	情報の BOM への格納	56
4.2.6	技術計算	57
4.2.7	Iteration	57
4.3	データ管理	58
4.4	データ出力	60
4.5	まとめ	60
第5章	検証・評価	61
5.1	評価内容および条件	62
5.2	JSM および JSM editor の効果	63
5.3	Virtual Engineering 設計環境の有効性	65
5.3.1	処理時間	65
5.3.2	羽根車バリエーション	67
5.3.3	Iteration	68
5.3.4	機能開発技術の変更	69
5.3.5	環境運用技術・機能開発技術の更新・追加	70
第6章	結言	72
6.1	結論	73
6.2	今後の課題	74
6.2.1	1DCAE Virtual Engineering 環境	74
6.2.2	その他機械製品システムへの展開	75
参考文献		76
謝辞		83
本研究に関する論文発表		84
他社所有商標などの表示		85

付章	3D モデル, 計算書	86
付.1	3D モデル	87
付.1.1	構成部品バラエティ	87
付.1.2	3D-CAD データバラエティ	88
付.2	計算書	90
付.2.1	種類	90
付.2.2	設計ルール	91
付.2.3	計算書例	92
付.2.4	3D-CAD の計算書としての活用	96

(全 97 頁)

# 第 1 章

## 緒 言

## 第 1 章 緒言

### 1.1 背景

#### 1.1.1 設計のフロントローディングと 1DCAE

性能に影響を与える設計仕様を複数有する機械の設計において、要求性能を満足する設計仕様の組み合わせを選び出すことは困難な作業である。これは、詳しくは 4.1 節にて述べるが、本稿で対象とする製品である発電用水車を例に説明すると、理想とする発電用水車は高出力・低重量であるが、実際の関係は高出力・高重量または低出力・低重量であり、性能がトレードオフとなっていることに起因する。加えて発電用水車には、安全性、効率、コスト等、多くの性能が要求される。また、発電用水車は、水車本体の他、発電機、軸受け潤滑装置、回転数制御装置、オペレーション制御装置等複数の補器が必要であり、これら一式で一つのシステムを形成している（以下製品システムと称す）。このことから、発電用水車においては、前記補器類との組み合わせも性能に影響を与える要因となる。また、一つの性能に限った場合においても、その性能を決定する設計仕様は複数存在することが多い。例えば、発電用水車の効率の場合、発電用水車の回転数、羽根車直径、羽枚数等の設計仕様が存在する。また、これらの仕様は発電用水車の他の性能に影響を与え、ひいては補器の性能に影響を与える。これらは複雑に影響しあっており、前述の通りトレードオフの関係になっている場合がほとんどである。近年、性能の高効率化を代表として、環境、コスト、納期などの要求が高度化してきており、これにより製品システムの複雑化し、設計仕様の組み合わせを選び出す作業はますます煩雑化し困難になってきている。

製品システムの複雑化は、企業における製品設計の分業化を余儀なくしている。機械設計とそれに付属する電気、制御設計の分業はもとより、機械設計においてもユニット、部品単位で分業化されている例が多い。この傾向は、今後ますます強まるものと考えられる。製品設計の分業化が進むことにより、詳細設計後のユニット間、部品間の仕様（取合い、専有面積、重量等）に不整合が生じ設計をやり直す、いわゆる「手戻り」が発生しやすくなる。このため、初期設計の段階に

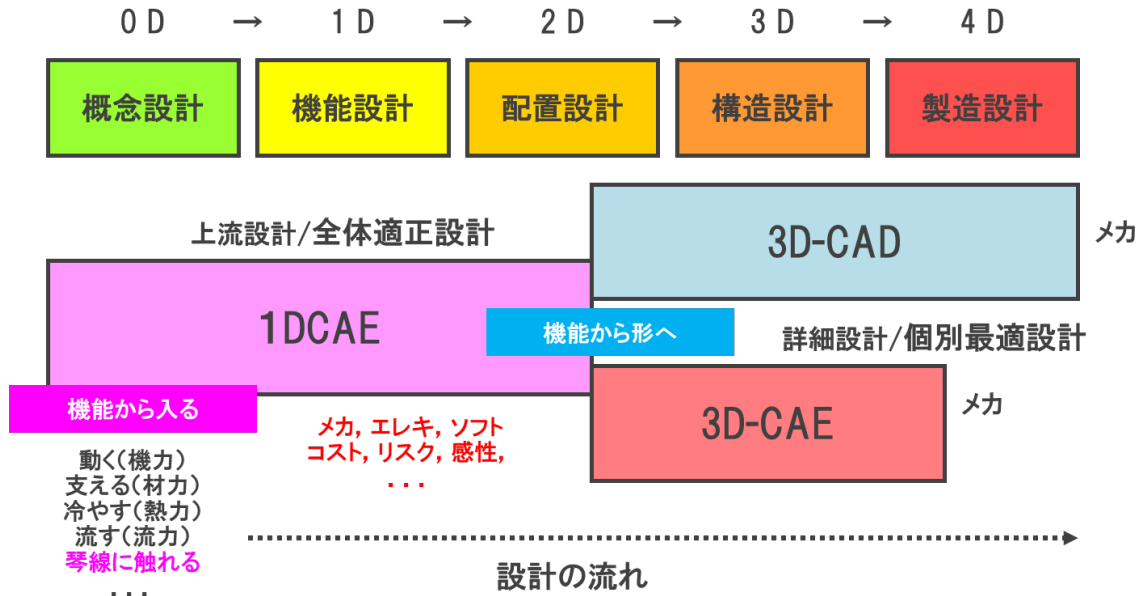
において、製品を構成する設計仕様の組み合わせを適切に決定し、ユニット間、部品間の仕様やインタフェースを決めておくことが望ましい。これは従来詳細設計時に行っていた作業を初期設計の段階に行うもので、設計のフロントローディングと呼ばれる。

前述の通り、設計のフロントローディングの目的は「初期設計の段階で要求性能を満足する設計仕様の組み合わせを選び出す」こと、および「ユニット間、部品間の仕様やインタフェースを決めておくことで手戻りを無くす」ことである。これに加えて、製品システム製造にかかるコストを精度よく推定できるメリットもある。このため、設計のフロントローディングは製品受注前の段階で行うことが望ましい。ここで、詳しくは 2.1.2 項にて説明するが、従来の設計は、まず製品システムの主要部を設計し、この結果を利用して細部の設計に展開していく「積上げ設計」と呼ばれる方法である。積上げ設計では、前述の通り「手戻り」が発生しやすい問題に加え、上流側の設計が終了しないと下流側の設計に着手できない問題があり設計に時間を要する。発電用水車の場合、受注前に設計にあてられる期間はおおむね 2～4 週間程度であり、正確に積上げ設計を行う時間の確保は難しい。したがって、設計のフロントローディングの効果を十分に発揮するためには、製品システムにおける機械設計とそれに付属する電気、制御設計、さらには機械設計におけるユニット、部品単位の設計を並行して実施できる技術が求められている。

この問題に対して大富らにより、「1DCAE」という概念が提案されている<sup>(1)-(6)</sup>。1DCAE とは、製品設計を行うにあたり、形を作る前に機能ベースで対象とするシステム全体を漏れなく表現して個々の機能を解析・評価し、詳細設計前に全体最適設計を行う設計概念である。また、全体最適設計の結果を持ってユニット、部品の個別最適設計を実施し、さらに個別最適設計の結果を全体適正設計に戻し、システム全体の検証を行うことも提案されている。Fig 1.1 に示すように、製品システム設計の流れは、上流側から 0D から 4D の 5 つのフェイズに分けることができる。これらの内容を以下(1)から(5)に示す。1DCAE は主に (1) の概念設計から (3) の配置設計のフェイズに適用される。なお、1DCAE においては、製品シス



テム全体の機能を考えることが重要であり，構造は明確になっている必要は無いとされている。



**Fig. 1. 1 Areas covered by 1DCAE in product design, reproduced with permission from ref. 2 with permission from The Japan Society of Mechanical Engineering**

- (1) 0D: 概念設計  
製品全体の仕様設計であり，機能，重量，コスト，納期等の計画を含む。
- (2) 1D: 機能設計  
(1) で決定した仕様等を機能として部品またはモジュールへと分解し個々の部品またはモジュールの機能を決定する。意匠設計もここに含まれる。
- (3) 2D: 配置設計  
部品またはモジュールのレイアウト設計および部品またはモジュール間のインタフェース設計。
- (4) 3D: 構造設計  
部品またはモジュール内の詳細設計。

## (5) 4D: 製造設計

### 製造方法の決定

以上本項では、製品システムにおいては、「要求性能を満足する設計仕様の組み合わせを選び出すことは困難な作業である」ことおよびこの作業を「設計のフロントローディング」として短時間で行うことが求められていることを述べた。さらにこれを解決する手法として 1DCAE の概要を説明した。

#### 1.1.2 設計プラットフォーム

前述の通り、1DCAE は設計概念であり、実際の製品システムの設計においては 1DCAE の概念を取り入れた設計環境、つまり設計プラットフォームが必要である。近年、1DCAE を実現する技術として、3D-CAD を用いた設計/開発技術が進歩しており<sup>(7)-(9)</sup>、欧米では CAD/CAM/CAE/DMU (Digital Mock-Up) らの垣根を廃してこれらを同一の場で活用する Virtual Engineering の環境が整いつつある<sup>(10)-(15)</sup>。しかしながら、日本においては、AEM (Assimilability Evaluation Method) 等一部で活用が見られるものの、その可能性が過小評価されており、従来の技術や手法に固執することで、新しいトレンドについて行けなくなる現象がさまざまな業界で起こっていると言われている<sup>(16)</sup>。また、上記有益な設計/開発ツールを使用している場面においても、前後の作業との連携が取れておらず、限定的な適用にとどまっているケースも見られ、上流設計段階で十分に活用されているとは言い難い状況にある。

ここで、Virtual Engineering に係わる技術は、以下の 3 つに分類される。

- 環境運用技術 (CAD, CAE 等のソフトウェア)
- 機能開発技術 (分析や設計妥当性評価等の手法)
- 活用技術 (実運用設計プラットフォーム)

環境運用技術の開発・推進に関しては、主にソフトウェアメーカーを中心としたデジタルの専門家によって行われており、機能開発技術の開発・推進は主に研究

機関や大学によって行われている。しかしながら、活用技術に関しては、中心となって推進していく仕組みがなく、このことがトレンドについていけない状況をつくりだしていると考えられる。

実際の製品設計においては、設計プラットフォームは対象となる製品システム別に異なることが一般的であるが、本研究では、詳細は 3.1 節で述べるが、適用できる製品システムに汎用性を持たせる目的で 3D-CAD を中心とする設計プラットフォームを提案する。また、3D-CAD は一般に以下に示す特徴を有していることも理由である。

- (1) 2D-CAD データと比較して、形状に曖昧さがない。
- (2) データに物理特性を付加することが可能であり、質量や慣性モーメント等の物理情報を容易に得ることが可能である。
- (3) ほぼすべての市販ソフトウェアにおいて、データ共有が前提で設計されており、複数のユーザ間で情報を共有しやすい。
- (4) 2D-CAD, CAM, CAE 等へのデータ移行が容易である。

特に (2), (3) 項記載の特徴は、1DCAE, つまり概念設計, 機能設計, 配置設計のフェイズにおいて、その過程および結果を 3D モデルの中に蓄えることができ、かつ、これらの共有・流用を容易とすることから有効に働くと考えられる。なお、1DCAE においては、機能ベースで対象製品システム全体を表現することが重要であるため、3D モデルは簡易なものでかまわないとされる。

### 1.1.3 BOM と入出力定義

前記の通り 1DCAE は概念であるが、研究開発においては、MapleSim, SimulationX 等, 汎用的な環境としての設計プラットフォームも具体的に提案され始めている。しかしながら、1DCAE を始めとする設計のフロントローディング概念・手法を実際の設計・製造に活用しようとする時、既存のアプリケーション、つまり外部ソフトウェアも利用することから、すべての処理が単一のソフトウェア内で完結することは少なく、複数の異なる外部ソフトウェア間にわたっての処

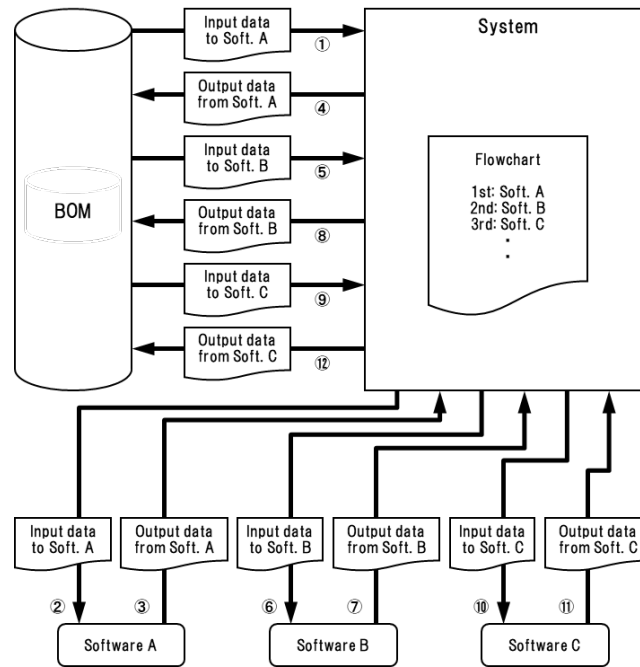
理を行う必要がある。したがって、実際に業務を行う設計プラットフォームは異なる外部ソフトウェアの機能を連携できることが望ましい。

これら外部ソフトウェアは個々に異なる入出力値群を必要とする。イメージを Fig. 1. 2 に示す。Fig. 1. 2 は設計プラットフォーム全体を示しており、設計システム (System, 詳細は第 3 章で説明する) を中枢とする。また、Fig. 1. 2 における BOM (Bill of Material: 数値等を保管する DB (Data Base)) とは設計分野においては「部品表」を示すものであるが、近年ではデータベースとして管理されており、製品システムの構成のみならず顧客仕様、性能計算結果、各種部品の寸法などが格納されている。設計システムはフローチャートに従い、外部ソフトウェアを A, B, C, … の順に実行する。設計システムは外部ソフトウェア A を実行するにあたり、BOM より外部ソフトウェア A が必要とする適切な入力値群を取得する。次いで設計システムはこの入力値群を外部ソフトウェア A へ送る。外部ソフトウェア A は入力値群を処理し出力値群をシステムに返す。設計システムは外部ソフトウェア A から受け取った出力値群を BOM の適切な位置に格納する。したがって設計システムには、外部ソフトウェアの実行フローおよびそれぞれの外部ソフトウェアへの入力として BOM 中のどの値を渡すか、また、外部ソフトウェアから出力された値を BOM 中のどこに格納するかを明確に定義しておく必要がある。

実際の設計環境においては、実行フローの図示化・可視化が必要であり、G-RD (Global Relations Diagram of Function and Demarcation) 等の手法が提案されている<sup>(17)</sup>。また、実行フローを実施する手法として、SOA (Service-Oriented Architecture) 等が提案されている<sup>(18)</sup>。SOA では異なる外部ソフトウェア実装における通信プロトコル標準化<sup>(19)</sup>に遅れが見られるものの、外部ソフトウェアプラットフォームの概念は確立されつつあり<sup>(20)</sup>、また、実際の業務に即した外部ソフトウェアの実行フローを実用的な時間・労力で構築する機能も提供されてきている<sup>(21)</sup>。

これに対して、BOM と入出力値群の関係定義は、現状、設計システム上で個々の値に対して手作業で設定が行われている。近年の設計・製造環境においては、入出力値群の肥大化や使用する外部ソフトウェアの増加に伴い、手作業による入

出力値群の定義は，多大な作業時間を必要とするとともにミス発生の要因となっている。また，これらの定義は一度の作業で終わるものではなく，業務の変化による外部ソフトウェアの追加・削除，作業手順の変更に従う必要もある。したがって，個々の外部ソフトウェアが必要とする入出力値群を俯瞰して定義・管理できる手法が重要である。



**Fig. 1. 2 Image of data translation**

## 1. 2 研究目的

1.1 節では，製品システムの設計において，「要求性能を満足する設計仕様の組み合わせを選び出すことは困難な作業である」こと，およびこの作業を「設計のフロントローディング」として短時間で行うことが求められていることを述べた。さらに，設計のフロントローディングに有効な手段として 1DCAE という概念が提案されていること，1DCAE を活用するにあたり，実際の製品システムに汎用的に適用できる設計プラットフォームが望まれていること，加えて，設計プラットフォームにおいては，外部ソフトウェアが必要とする入出力値群を俯瞰(ふかん)して定義・管理できる手法が重要であることを述べた。上記の要求を実現することを目的として，本研究では，製品システムの設計において，手戻りなどの問題

点を含む従来の設計手法から脱却するための、異なる製品システムの設計にも適用できる 1DCAE およびフロントローディング設計プラットフォームを提案、開発する。具体的には、「活用技術」と「設計プラットフォームのトレンド追従」の 2 点に分け、以下に示す各問題点に対する解決方法を提案して開発するものである。

- 活用技術

- ◆ 問題点

- 1.1.1 節で述べた通り、現在の 1DCAE は概念であり、また、1.1.3 節で述べた通り、研究開発の場等を除き、複雑な製品システムに適用できる汎用的な設計プラットフォームは提案されていない。これは以下の理由による。

(A) 製品システムごとに異なった環境を構築する必要がある。

(B) 使用する環境運用技術・機能開発技術の数が多く、1.1.3 節記載の BOM と入出力値群の関係定義が煩雑である。

- ◆ 解決方法

- フロントローディング設計に有効な 1DCAE に対応する Virtual Engineering 設計環境として以下の技術を開発・実装する。

(A) 環境運用技術・機能開発技術を活用技術から独立し、環境運用技術・機能開発技術を入れ替えることで異なる製品システムに対応する。  
このとき、

- ◆ 従来の Engineering System (設計プラットフォームシステム本体) に含まれる要素を分解し、Engineering System の仕事は外部ソフトウェア実行手順に沿った BOM-外部ソフトウェア間の入出力値群の受け渡しのみとする。
- ◆ 変更・修正のインターフェースは 3D-CAD とする。

(B) BOM と入出力値群の関係を俯瞰して定義・管理できる手法を開発する。なお、本内容は次の「設計プラットフォームのトレンド追従」に含む

## ● 設計プラットフォームのトレンド追従

### ◆ 問題点

- 一度構築した設計プラットフォームの寿命は限定的である。これは、環境運用技術および機能開発技術が日々進歩しているからである。このため、設計プラットフォームを運用し続けるためには、必要に応じて、1.1.3 節で述べた通り、外部ソフトウェアの追加・削除、作業手順の変更を行い続けなければならない。しかしながら、実際の設計現場において、設計プラットフォームに専任のメンテナンス担当者が配置されることはまれであり、設計者自らが通常業務の合間にこれらの変更を行わなければならないのが現実である。なお設計者が通常業務に手を取られメンテナンスがおろそかとなり、その結果、設計プラットフォームが陳腐化し最新の設計に使用できなくなる事例も多い。

### ◆ 解決方法

- Engineering system に手を加えることなく、また C 言語等のプログラミング技術を必要とせず BOM 構成および外部ソフトウェアが必要とする入出力値群を俯瞰して定義・管理できる手法を開発する。本内容には以下を含む
  - (A) 環境運用技術・機能開発技術の追加/削除に伴う外部ソフトウェア実行手順定義
  - (B) 環境運用技術・機能開発技術の追加/削除に伴う BOM-外部ソフトウェア間の入出力値群定義
  - (C) 上記を簡便に運用するためのエディタ開発

なお、問題点の詳細は第 2 章、解決方法の詳細は第 3 章で述べる。また、本論文には適正化/最適化ロジックは含んでいない。また、実際の製品システムとして発電機用水車を用いている。発電機用水車に関しては 4.1 節にて詳しく述べる。

## 1.3 論文構成

本論文は第 2 章以降、各章で以下の内容を記述する。

## 第 2 章：従来手法と課題

活用技術および設計プラットフォームのトレンド追従における従来方法の課題の説明

## 第 3 章：課題解決の手法

第 2 章記載の課題を解決する本研究における開発の説明

活用技術

設計プラットフォームにおける Engineering System と BOM, Execution Flow 等のデータ部を完全に独立した構成とし，BOM 内のデータと各種計算書の間を，JSM (Job vs Spec Matrix) を用いて定義する。

プラットフォームのトレンド追従

JSM を改良した表記方法を開発し，個々の外部ソフトウェアが必要とする入出力値群を俯瞰して定義・管理できる表記方法を実現する。また，本表記方法を活用するための専用エディタを作成する。

## 第 4 章：発電用水車を対象とする実際の Virtual Engineering 設計環境

第 3 章にて開発した手法の有効性検証を目的とする Virtual Engineering 設計環境の制作および説明

## 第 5 章：検証・評価

第 3 章にて開発した手法および第 4 章にて作成した Virtual Engineering 設計プラットフォームの検証・評価

## 第 6 章：結言

付章：3D モデル，計算書

第 3 章，活用技術にて独立させたデータの付属説明



## 第 2 章

### 従来手法と課題

## 第 2 章 従来手法と課題

第 1 章にて、手戻り等の不具合を解決するために、設計のフロントローディングが重要であることを述べた。本章では設計のフロントローディングを行う上での課題を活用技術と設計プラットフォームのトレンド追従の面から説明する。

### 2.1 活用技術

#### 2.1.1 従来手法

従来手法として既存設計システムについて説明する。既存設計システムは古典的手法である積上げ設計，すなわち，まず設計対象であるシステムの主要部を詳細に設計し形をつくり，次いで付属・補器部品の詳細設計へと展開していく手法をベースとしており，これを手作業から 2D-CAD, 3D-CAD, CAE と言ったデジタル手法に置き換えたものである。ここで，特に従来の手作業時には存在しなかった 3D-CAD によるモデリングや CAE によるシミュレーションなどは Virtual Engineering と呼ばれる。デジタル手法が取り入れられ始めたころは，例えば部品 A 設計，部品 B 設計など個々の設計作業・結果が自動でつながっておらず，前作業の結果を手動で次の作業に入力する必要があったが，近年では設計作業・結果を一つのシステム内で管理することも増えてきており，市販設計統合環境ソフトウェア製品も存在する。

#### 2.1.2 課題

従来手法には以下の課題が存在する

- (A) 積上げ設計においては，設計の初期段階においてシステム全体の完成形が見えにくく，また，先行する部位の設計が終了しないと次の部位の設計に取りかかることができない。また，後発の部位において設計が成立しない場合，先発の部位に戻って設計をやり直す「手戻り」が発生することがある。これを避けるために先発の設計に余裕を持たせることも多く，製

品システムが肥大化する恐れがある。

- (B) 従来設計システムでは主として以下のものがシステム内に直接コーディングされており，これらを変更するために IT 技術者を必要とする。
- 環境運用技術・機能開発技術の登録（各々のシステム上での実行ファイル位置）
  - 設計フロー（環境運用技術・機能開発技術の実行順序）
  - 環境運用技術・機能開発技術の入出力定義（BOM 内のどの値をどの環境運用技術・機能開発技術の入力とし，また，それぞれの環境運用技術・機能開発技術の出力を BOM のどこに格納するか の定義）
- (C) また，市販設計統合環境ソフトウェア製品を使用している場合，変更内容によってはユーザ側で変更することができない仕様となっており，メーカー側に変更を依頼する必要がある。このため環境運用技術や機能開発技術の変更に従従することが容易でない。
- (D) 設計システムには，設計者が設計の変更・修正を行うインタフェースを備える。従来の設計システムでは，これらは設計システムに付属する専用のインタフェースから行われる。このため，設計時に寸法などの値を変更・修正を行う個所は設計システム構築時にあらかじめ決めておく必要がある。また，変更・修正を行う個所は製品システムごとに異なるため，インタフェースはそれぞれの製品システムごとに異なるものとなる。これを変更する場合は上記 B 項記載の問題があり容易ではない。

第 1 章で述べた通り，実製品設計に適用し得る 1DCAE Virtual Engineering 環境においては，環境運用技術および機能開発技術の実装や更新を速やかに実行できることが重要である。ここで，設計システムの寿命が短くてもよい研究開発においては MapleSim, SimulationX 等，汎用的な環境も具体的に提案され始めている。しかしながら，設計システムの寿命が長いことが求められる実製品設計においては，上記課題により，日々進化する環境運用技術および機能開発技術に従従することが容易でなく，一度構築された設計システムが適用できなくなるという状況が起こりやすい。

## 2.2 設計プラットフォームのトレンド追従

設計プラットフォームをトレンドに追従させるためには、1.1.3節にて説明したBOMと外部ソフトウェアとの入出力定義を容易に行えることが必要である。

入出力群定義に関する方法は元山らにより提案されており<sup>(22)-(28)</sup>、DSM (Design Structure Matrix) と CRUD 図に大別できる。Fig. 2.1 に DSM および CRUD 図の例を示す。

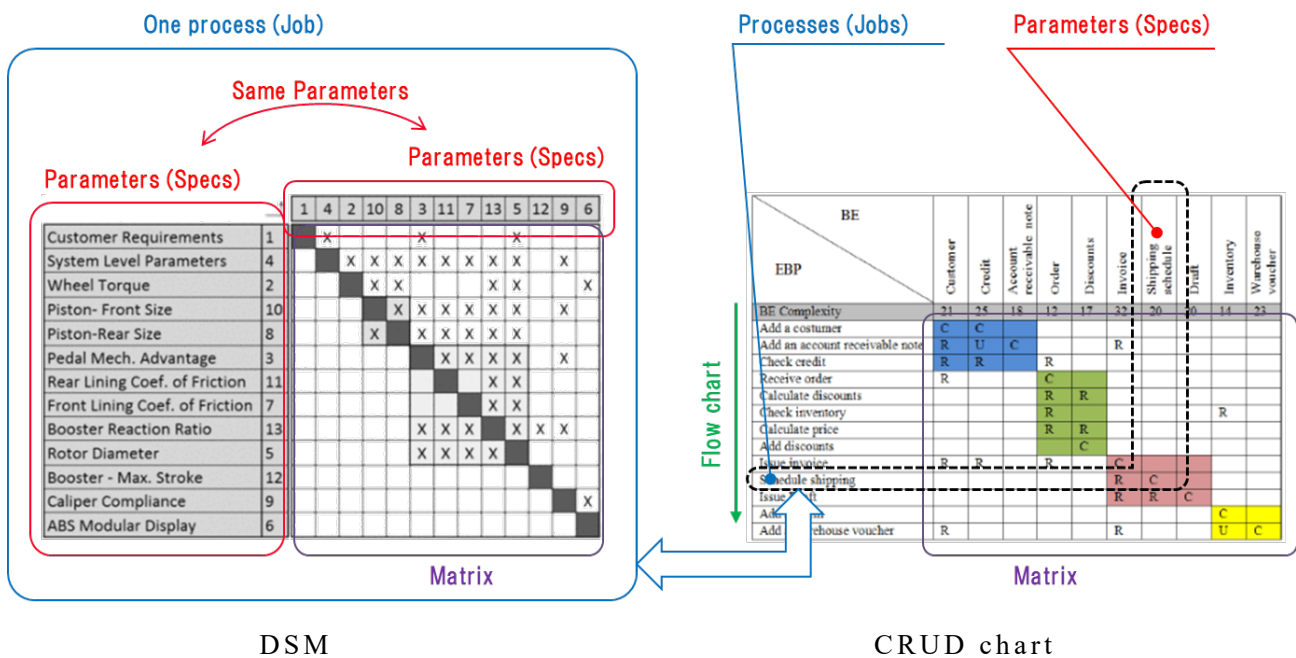


Fig. 2.1 DSM and CRUD chart Examples

### (A) DSM

製品開発や組織開発などを構成する入出力値 (Parameter) 間依存関係を縦横同じ要素のマトリックス形式で表現したもの。製品システム全体からアセンブリ、個々の部品へと入出力値の関係性を多面的に表現することが可能である<sup>(29)</sup>。また、入出力値群または入出力値単体の要素間依存関係がビジュアル的に把握しやすく、主に製品開発段階での重要な要素やインタフェースの把握と言ったスクリーニングに用いられる<sup>(30)</sup>。

## (B) CRUD 図

主に IT システム開発に用いられ、縦軸に個々のソフトウェアがデータに対してどのような処理をするか (Flow chart) を、横軸に入出力値群 (Parameters) をマトリックス形式で表現したもの。マトリックス内に関係性をデータベース管理システム (DBMS, Database Management System) でデータを操作する際の操作種別である Create (作成), Read (参照), Update (更新), および Delete (削除) で示す。データと機能 (画面) における機能の不足, 処理の集中や抜け漏れを確認するために使用される。

DSM は、個々の値を個別に表記し、関連を可視化できることが特徴である。しかしながら DSM は、製品システムまたはそれを構成するアセンブリ、部品単体と言った工程単位で表記するものであり、すべての工程を俯瞰して表記することができず、複数の外部ソフトウェアに渡る入出力値の関係性を表記するには不向きである。また、CRUD 図は、複数のソフトウェア間における入出力関係を俯瞰して表記できることが特徴であるが、入出力値をまとめて一つの群として扱うため、個々の入出力値を管理することができない。現実の設計・製造業における実行フローに記載する外部ソフトウェア数は、一つのソフトウェアの機能が複数回使用されることも考慮すると、数十～数百におよび、また取り扱う入出力値数は数千～数十万に及ぶ。したがって、入出力値を俯瞰して定義・管理するためには、DSM と CRUD 図双方の機能、すなわち、入出力値群の値を個々の値として取り扱うことおよびすべての工程を俯瞰して表記できることが必要であると考えられる。

## 第 3 章

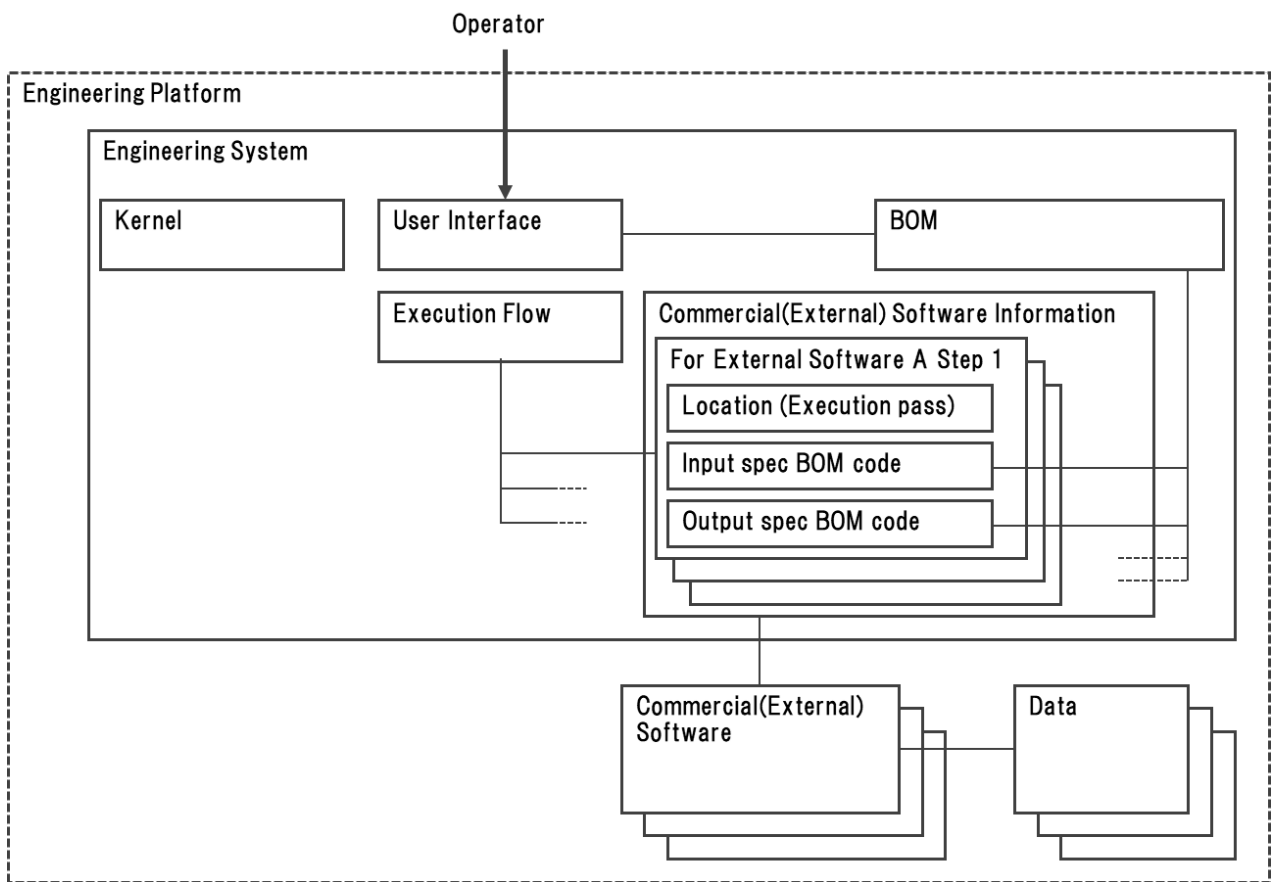
### 課題解決の手法

### 第 3 章 課題解決の手法

本章では，第 2 章で説明した課題の解決方法として，本研究で開発した内容を説明する。

#### 3.1 設計プラットフォーム要素の再構築

Fig. 3.1 に一般的な設計プラットフォームの模式図を示す。



**Fig. 3.1 Typical Engineering Platform Diagram**

Fig. 3.1 に示すように一般的な設計プラットフォームにおいては外部ソフトウェアおよび外部ソフトウェアのデータを除くすべての要素が設計システム内に埋め込まれている。ここで，それぞれの要素を機能別に分類すると以下の通りとなる。

- Kernel: 処理を実行する
- User Interface: 情報を入力（変更を含む）・確認する
- Data: 情報を格納する
  - BOM
  - Execution Flow
  - Commercial (External) Software Information
  - Location (Execution pass)
  - Input spec BOM Code
  - Output Spec BOM Code
  - Commercial (External) Software Data
- Commercial (External) Software: 情報を処理する

ここで、設計システム内に最低限必要な Kernel を除き、ほかの要素をすべて設計システムの外部に置くことを考えた。ここで、以下の 2 点が問題となる。

- User Interface
- 以下の Data の管理
  - Execution Flow
  - Commercial (External) Software Information
  - Location (Execution pass)
  - Input spec BOM Code
  - Output Spec BOM Code

上記に対して、まず User Interface に関して、2.1.2 項で説明した通り、従来の設計プラットフォームにおいては、対象とする製品システムに特化したものであり、環境運用技術・機能開発技術の変更等が生じる場合は、User Interface の変更により IT 技術者を必要とするものであった。本研究では機械設計において、「すべての計算・検討結果は機械部品の寸法に現れ、また、機械部品の寸法から変更でき



る」との発想から初期値の入力・計算結果の閲覧を除き，User Interface は 3D-CAD を用いることとした。このようにすることで，専用の User Interface を用意する必要がなくなる。またデータの管理に関しては，詳細は 3.3.1 項で説明するが，JSM (Job vs Spec Matrix) を用いる手法を開発した。Fig. 3.2 に再構築後の設計プラットフォームの模式図を示す。Fig. 3.2 では環境運用技術・機能開発技術と活用技術 (Engineering System) およびこれらをつなぐ要素が独立しており，環境運用技術・機能開発技術の追加/削除を Data の修正のみで対応可能である。

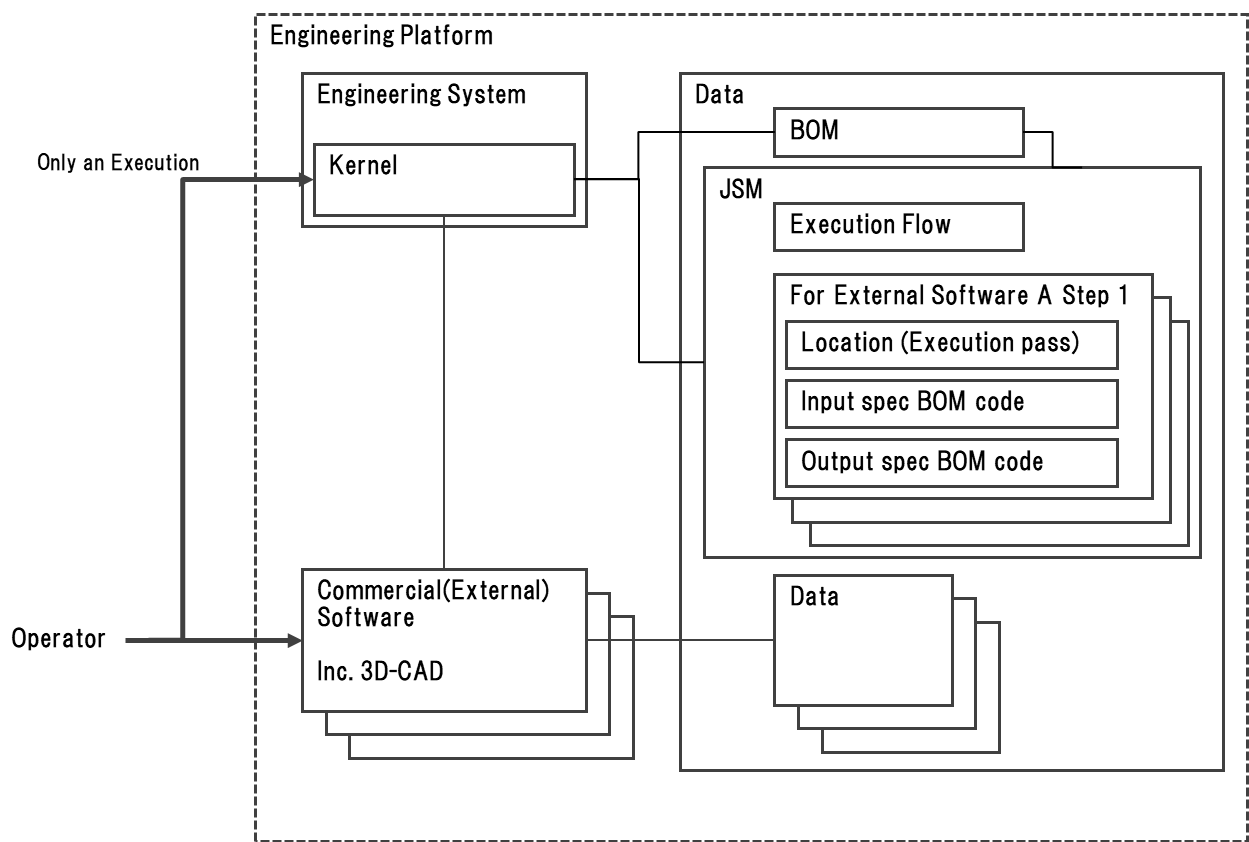
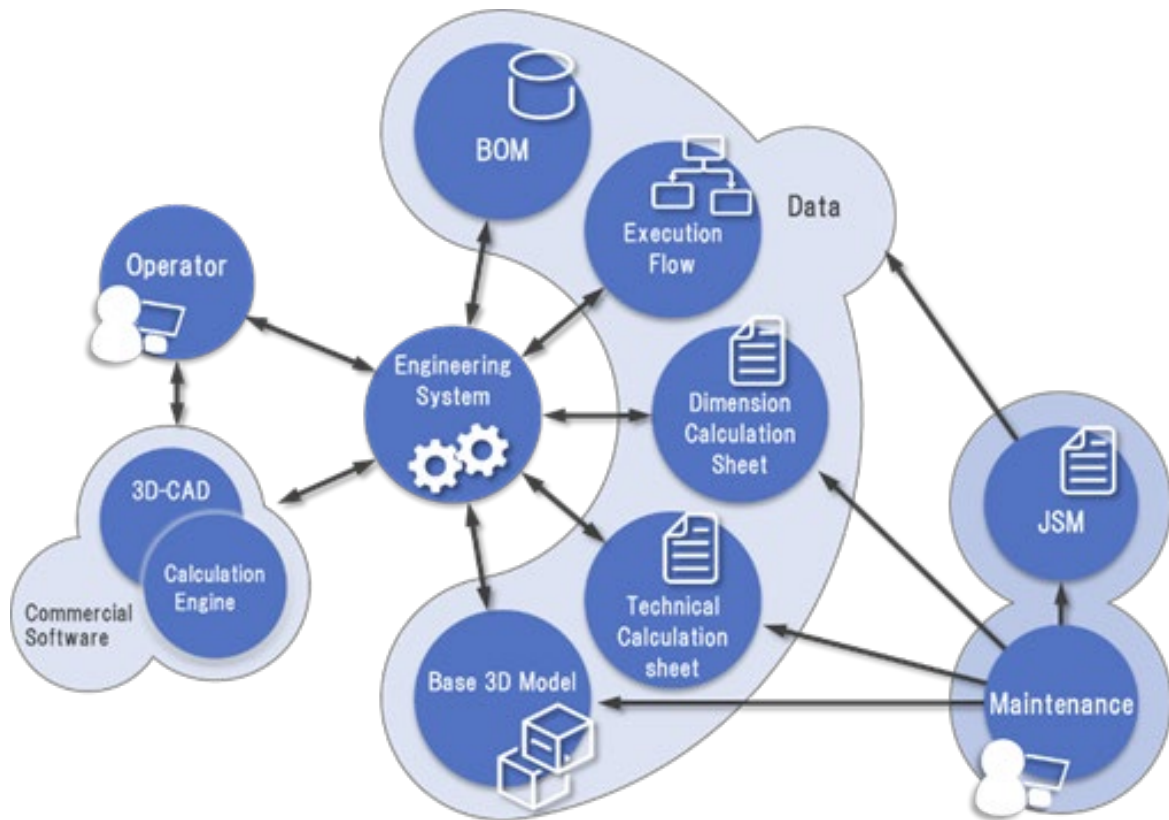


Fig. 3.2 Restructured Engineering Platform Diagram

### 3.2 環境技術としての設計プラットフォーム

Fig. 3.3 は Fig. 3.2 を元として，本研究が提案する 3D-CAD を中心とする 1DCAE Virtual Engineering 環境の概念を示した図である。



**Fig. 3. 3 Conceptual Diagram of Design Platform for 1DCAE**

Fig. 3. 3 中の語句について以下説明する。

- ・ **Engineering System**  
Virtual Engineering 設計環境の中心であり， Operator からの指示を受け処理を実行する。なお， 処理は後述の Execution Flow にしたがって行われる。また， BOM を含む各種 Data 間のやり取りを行う。
- ・ **Operator**  
Virtual Engineering 設計環境のユーザであり， Engineering System および Commercial Software （外部ソフトウェア）を操作し実設計業務を行う。
- ・ **Data**  
以下のデータから構成される。
  - BOM

顧客データや 3D モデルの寸法，各種計算結果等，すべてのデータを保管する DB (Data Base)。

- Execution Flow  
各種計算の実行順序が定義されるフロー。
  - Dimension Calculation Sheet  
機械製品システムの寸法計算ルールが定義される計算書。詳細は 4.2.1 項にて説明する。
  - Technical Calculation Sheet  
機械的強度や性能等をバックチェックするための技術計算書。詳細は 4.2.1 項にて説明する。
  - Base 3D Model  
基本 3D モデル(3D-CAD ソフトの部品またはアセンブリファイル)。詳細は 4.1 節にて説明する。
- ・ Commercial Software  
市販外部ソフトウェアであり，3D-CAD ソフトウェアや数値演算・解析を行うソフトウェア
  - ・ JSM  
BOM 内のデータと各種計算書の関係を定義するファイルおよびソフトウェア。実設計業務とは切り離されている。詳細は 3.3.1 項で説明する。
  - ・ Maintenance  
データの更新や新規計算書追加等のメンテナンス作業。実設計業務とは切り離されている。

本環境は大きく以下の 3 つの機能およびデータ群で構成している。

- 環境運用技術
  - 3D-CAD
  - Calculation Engine (数値演算や表計算を行うソフトウェア)
- 機能開発技術

- Dimension Calculation Sheet（機械寸法を決定する式を記載したシート）
- Technical Calculation Sheet（機械強度などの技術計算を行う式を記載したシート）
- 活用技術
  - Engineering System
- データ
  - BOM
  - Execution Flow（実行フロー）
  - Base 3D Model（基本 3D モデル）
  - JSM

本研究の目的は、汎用的な設計システムとしての 1DCAE Virtual Engineering 環境を提案することにある。2 章で述べた課題から、本環境は以下の概念で構築している。

#### A) 積上げ設計の排除

製品システムが要求する顧客仕様を入力として、製品システム全体に対して最低限成立する構造・構成を自動で提案・3D モデル化する。提案された 3D モデルに対しての設計者（Operator）からの設計変更は製品システムのどの部位（寸法、材質等）に対しても優先順位なく実施可能とする。

#### B) 構成要素の独立

すべての環境運用技術、機能開発技術およびデータは、1DCAE Virtual Engineering 環境内でそれぞれが独立しており、例えば環境運用技術実行時のファイルパスも Engineering System 内にはコーディングしない。ここで、「環境運用技術・機能開発技術の登録」、「設計フロー」、「環境運用技術・機能開発技術の入出力定義」は JSM を用いて管理する。

### C) インタフェースの統一

A 項にて実施する設計変更は 3D-CAD を通して行う。すなわち，設計変更のインタフェースは 3D-CAD のインタフェースとなる。

A 項を実現することで，優先順位なく詳細設計を進めることが可能であり，また，製品システム全体を見ながら評価・修正を行えるため，設計者が直感的に機械構成のバランスを取りやすくなる。これは 1DCAE の重要な考え方のひとつである，「個別最適化設計の結果を全体適正設計に戻す」ことに相当する。B 項を実現することで，IT 技術者でなくとも環境運用技術・機能開発技術の更新が可能であり，構築した 1DCAE Virtual Engineering 環境を長期間にわたって運用することが可能である。C 項を実現することで，インタフェースが統一され，環境運用技術・機能開発技術変更の際に JSM の定義を除き 1DCAE Virtual Engineering 環境に変更を加える必要がない。また，環境運用技術・機能開発技術を入れ替えることで，異なる製品システムの設計にも使用することが可能となる。

ここで，インタフェースとして 3D-CAD を選択した理由は，3D-CAD は，多くの市販ソフトウェアにおいてパラメトリック機能を有しており，設計変更が容易に行えるからである。また，3D model 内に材料比重等の物理特性を付加することができるため質量や慣性モーメント等の物理情報を手間なく得られること，さらに 2D-CAD，CAM，CAE 等へのデータ移行が容易であることも理由であり，これらの特徴から 1DCAE において有効に働くと考える。なお，3D-CAD と比較して 2D-CAD は図面情報を一画面もしくは紙面にて一括で確認できるという特徴がある。したがって，1DCAE Virtual Engineering 環境内では 3D-CAD を中心として設計を行い，最終出力を 2D-CAD の図面とする考えである。

### 3.3 JSM を用いた設計プラットフォームのトレンド追従

#### 3.3.1 JSM

2.2項で述べた通り，入出力値群を俯瞰して定義・管理し，プラットフォームのトレンド追従を実現するためには，DSMとCRUD図双方の機能が必要であると考えられる。本研究では，JSMの機能を拡張することでこれを実現できると考え，外部ソフトウェアに対する入出力値群の関係定義を行うマトリックススタイルの表記方法を考案した。JSMの考案は日野ら<sup>(31)</sup>によるもので，IPO（Input-Process-Output）関係表とも呼ばれる。本来のJSMは，モジュラーデザイン（MD

：Modular Design）<sup>(32)</sup>、<sup>(33)</sup>において，新規のBOMを構築する際にBOMに格納すべき値を漏れなく抽出する手法であり，外部ソフトウェアに対する入出力値群の関係定義を行うものではない。また，日野らによっては，前記抽出作業以外の活用方法は言及されていない。JSMの例をFig. 3.4に示す。

JSMにおいて，Jobとは設計における手作業を含むあらゆる作業・工程であり，JSMの上辺に列挙する。なお，Jobの並び順は作業・工程の順番である必要はない。Specは個々のJobが必要とする入力値および個々のJobからの出力値であり，JSMの左辺に列挙する。なお，SpecはJobに対する入力値群・出力値群ではなく，個々の値単位で記載する。SpecもJobと同様に並び順を考慮する必要はない。Matrix領域には，個々のJobとJobに入力するSpecの交差位置に右上向矢印，個々のJobとJobより出力されるSpecの交差位置に左下向矢印をそれぞれ記載する。

JSMの作成は既知のJobおよびSpecを上辺および左辺に列挙することから始める。次いでMatrix領域に矢印を記載し，個々のJobに必要な入出力値がSpecとして記載されていることを確認する。また同時に，個々のSpecに関して，そのSpecを出力するJobが存在することも確認する。このときJobやSpecが不足している場合は追加を行う。上記の作業を行い，JobおよびSpecの不足がなくなった時点で左辺に列挙したSpecはBOMに格納すべき値として漏れなく抽出できたことになる。

Spec			Job							
Name	Symbol	Unit	Job1	Job2	Job3	Job4	Job5	Job6	Job7	Job8
Spec01	a1	mm	✓	✓		✓				
Spec02	a2	mm	✓	✓		✓				✓
Spec03	b1	mm			✓					
Spec04	b2	deg.		✓	✓	✓				
Spec05	c1	mm					✓	✓	✓	
Spec06	c2	mm					✓			
Spec07	c3	mm					✓			
Spec08	d1	mm				✓	✓	✓	✓	
Spec09	d2	deg.				✓	✓			
Spec10	e1	mm							✓	
Spec11	e2	mm								
Spec12	e3	mm								
Spec13	e4	mm								
Spec14	f1	mm						✓		
Spec15	f2	bool						✓		

Matrix

Fig. 3. 4 Sample of JSM

上記のように、JSMは、外部ソフトウェアに対する入出力値群の関係定義を行うものではないものの、実行フロー上のJobの位置関係に関係なく、個々のJobへの入出力関係を記載することができる。また、複数のJobにおける入出力関係を表現することが可能であり、設計工程全体を俯瞰して表記することが可能である。さらに入出力値を入出力値群ではなく個々の値で取り扱うことが可能である。したがってJSMは外部ソフトウェアに対する入出力値群の関係定義に有効な表記方法として適用できる可能性がある。しかしながら、JSMはBOMに格納すべき値として漏れなく抽出することのみを目的としており、例えばJobやSpecの重複を検出する手法などは提案されていない。また、その目的からBOMに格納す

べき値を抽出した後の活用は考慮されておらず，システムとの連携も検討が必要となる。

### 3.3.2 機能拡張

JSM の機能拡張は第一に，システム，BOM および外部ソフトウェアとの連携を取るための表記上の拡張，次いで記載内容の不備を確認するチェックルール構築の順に行った。以下 (1) にて表記上の拡張について，(2) にてチェックルール構築について説明する。

#### (1) 表記上の拡張

JSM を表記方法として活用するにあたり，以下 (A) から (F) に示す項目を拡張をした。また，拡張した JSM の例を Fig.3.5 に示す。

#### (A) Job の並びを業務順に定義

- Job の並びを業務順に定義することで，Job の並び順から実行フローを作成できるようになり，SOA など別の手法で業務フローを作成する手間を省くことができる。

#### (B) JSM マージ作業のための Job グループ定義欄追加

- 近年は業務担当者の専任化が進んでいることから，実際の JSM 作成においては，複数人による分担作業が行われるものと考えられる。したがって JSM の作成時には複数の JSM をマージする作業が発生する。そこで，例えば"部品 A 設計部分"等，個々の Job が属するグループを定義する欄を追加した。3.3.3 項で説明する本稿にて開発した専用のエディタは複数の JSM を一つにマージする機能を有している。本拡張によりエディタは job をグループ単位で取り扱うことができるようになりマージ作業が容易となる。

#### (C) Job に "A (Auto) ", "M (Manual) "のステータスを追加

- 個々の Job がシステム上で完全に自動で実行できるか人間の介入を必要とするかを明確にするものである。もちろん理想はすべての Job が完全に自動で実行できることである。



(D) Matrix 内の表記を以下のように変更

- Job への入力：右上向矢印→"I"
  - 下記"I/O"および"R"の追加に伴う表現の変更
- Job への出力：左下向矢印→"O"
  - 下記"I/O"および"R"の追加に伴う表現の変更
- Job への入出力："I/O"（新規追加）
  - 入力値を Job の出力値によって上書きすることを意味する。システム側の対応も必要であるが，これを利用することで，イタレーション機能を有さない外部ソフトウェアを用いて自動でイタレーションを行う機能を実現する。また，不要に BOM に管理する入出力値が増えることも抑止できる。
- Job への参照入力："R"（新規追加）
  - 一般に，機械設計における安全率，材料物性，JIS の規定値などは固定値として外部ソフトウェア内にコーディングすることが多い。しかしながら設計基準の変更によりこれらの値を変更する必要がある場合，もれなくコードを修正することは難しい。したがって，このような値は参照値ファイルを用意しておき，そこから"R"を使用して参照する仕組みとし，明確に表記を分けることとした。

(E) Spec への BOM コード（アドレス）追加

- 本アドレスはシステム実装時には個々の Spec に対して指定する必要があり，JSM 上で一括管理することを目的に追加した。

(F) Spec への外部ソフトウェアにおける内部アドレス記載欄追加

- 本アドレスはシステム実装時には個々の Spec に対して指定する必要があり，JSM 上で一括管理することを目的に追加した。

(2) チェックルール構築

記載内容の不備を確認するチェックルールとして以下の (A) から (E) のルールを構築した。本ルールに従うことで Job, Spec 共に必要な項目を過

不足なく列挙でき，かつすべての入出力が確実に連結していることが確認できる。また，(A) から (E) は全て独立したルールである。

- (A) 単一の Spec は一つ以上の Job への入力となっていなければならない。Job への入力となっていない Spec は不要であり削除する。

記載不備例

	Job 01	Job 02	Job 03	Job 04	Job 05	Job 06
Spec 01		0				

後 Job で使用しない値を計算する必要はない→不要な業務の削除

- (B) 単一の Spec は一つの Job からの出力を受け取っていないてはならない。Job からの出力を受け取っていない場合は，その Spec を決定する Job がないとみなし，新たな Job を作成する。

記載不備例

	Job 01	Job 02	Job 03	Job 04	Job 05	Job 06
Spec 01		1			1	

Job 02 より前に Spec 01 を決定する Job が必要

- (C) 単一の Spec は出力を複数の Job から受け取ってはならない。複数の Job からの出力は重複作業を意味し，Job 内容を精査の上，一つに限定する。

記載不備例

	Job 01	Job 02	Job 03	Job 04	Job 05	Job 06
Spec 01		0		0		

Job 02 または Job 04 を削除→重複作業の削除

- (D) 単一の Job には一つ以上の Spec 入力がないてはならない。Spec 入力なしに出力を決定できる Job は固定値を出力しているだけであり，不要と考え削除する。

記載不備例

	Job 01
Spec 01	
Spec 02	
Spec 03	
Spec 04	
Spec 05	0
Spec 06	0

Input なしで計算できる Spec は固定値→不要な業務の削除

- (E) 単一の Job は一つ以上の Spec への出力がないてはならない。Spec への出力がない Job は不要な Job と考え削除する。

記載不備例

	Job 01
Spec 01	
Spec 02	
Spec 03	
Spec 04	
Spec 05	
Spec 06	

Output がない→不要な業務の削除

The diagram shows a table with columns for Structure, BOM Code, Address, Name, Symbol, Unit, and Job groups (Group A and Group B). The BOM Code column is divided into three sub-columns (0, 1, 2). The Address column is divided into three sub-columns (File, Feature, Dimension). The Job group columns are divided into two sub-columns (Group A and Group B). The table contains 17 rows of data, with some cells containing values like 'a1', 'aC', 'b1', 'bC', 'c1', 'c2', 'c3', 'c4', 'e1', 'eC', 'eB', 'e4', 'f1', 'f2', 'f3', 'f4'.

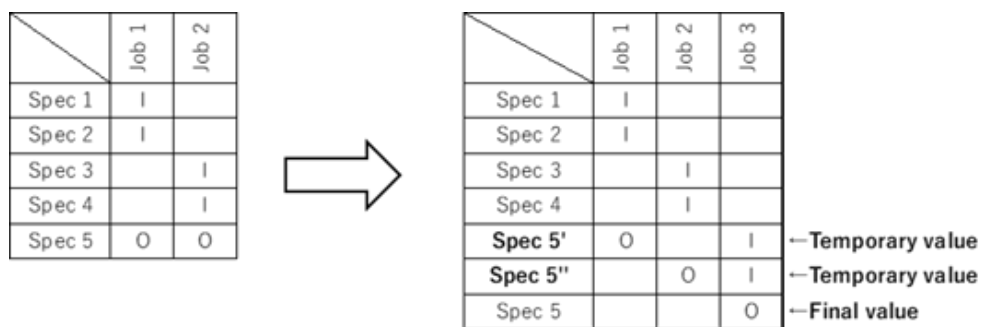
Structure	BOM Code	Address			Name	Symbol	Unit	Group A		Group B				
		File	Feature	Dimension				Job 1	Job 2	Job 3	Job 4			
Unit A	B00001	Assi A**	Fea. A	Dim. A	Spec 1	a1	mm							
	B00002	Assi A**	Fea. B	Dim. A	Spec 2	aC	mm	I	I	I				
	B00003	Assi AA**	Fea. A	Dim. C	Spec 3	b1	mm		O					
	B00004	Assi AA**	Fea. B	Dim. D	Spec 4	bC	deg.	I/O	I	I				
	Subunit AA	Part AA1	B00005	Part AA1**	Fea. A	Dim. A	Spec 5	c1	mm			O	I	I
			B00006	Part AA1**	Fea. B	Dim. A	Spec 6	cC	mm			O		
			B00007	Part AA1**	Fea. C	Dim. A	Spec 7	c3	mm			O		
	Part AA1	B00008	Part AAC**	Fea. B	Dim. B	Spec 8	c1	mm		O	I	I	I	
		B00009	Part AAC**	Fea. C	Dim. A	Spec 9	cC	deg.		O	I			
	Subunit AB	Part AB1	B00010	Part AB1**	Fea. C	Dim. B	Spec 10	e1	mm					O
			B00011	Part ABC**	Fea. A	Dim. A	Spec 11	eC	mm					
	Part AC	Part ABC	B00012	Part ABC**	Fea. B	Dim. B	Spec 12	eB	mm					
			B00013	Part ABC**	Fea. C	Dim. C	Spec 13	e4	mm					
			B00014	Part AC**	Fea. A	Dim. B	Spec 14	f1	mm					O
			B00015	Part AC**	Fea. B	Dim. A	Spec 15	f2	bool					O
			B00016	Part AC**	Fea. C	Dim. A	Spec 16	f3	mm					O
			B00017	Part AC**	Fea. C	Dim. B	Spec 17	f4	mm					O

Fig. 3. 5 Sample of Extended JSM

ここで、(C) 項に関して追記する。実際の設計工程においては、複数の Job からの出力値を比較・調整し、最終的な値を決定する実行フローが良く見られる。

(C) 項記載チェックルールはこれを否定、または JSM 上での表記を不可能とするものではない。Fig. 3. 6 を用いてこれを説明する。Fig. 3. 6 左図は単一の Spec である Spec 5 が複数の Job である Job1 および Job2 からの出力を受け取っている表記例である。

この表記は (C) 項記載のチェックルールに反し、また、仮にこれをシステム上で実行した場合、システムにも依存するが、恐らく Spec5 の値は Job2 の出力で上書きされる。これは比較・調整するという作業を一つの Job として追加することで解決できる。Fig. 3.5 右図は比較・調整を行う Job である Job3 を追加した例である。Job1 および Job2 の出力をそれぞれテンポラリーな Spec である Spec5' および Spec5'' に出力しており、次いでこれらを Job3 に入力して最終的な値である Spec5 を決定、出力している。このように比較・調整作業を個別の Job として追加することで、(C) 項記載のチェックルールに反することなく比較・調整作業を JSM 上に表記することが可能である。



**Fig. 3. 6 Sample of Value Comparison Expression**

### 3. 3. 3 JSM editor

現実の設計・製造業における実行フローに記載する外部ソフトウェア数、つまり JSM に記載する Job 数は数十～数百におよび、また取り扱う入出力値数、つまり Spec 数は数千～数十万に及ぶ。これだけの値を取り扱うにあたって、3. 2. 2 項記載の確認をもれなく行うこと、また、作成した JSM を設計システムに手動で組込むことは現実的ではない。したがって、本研究では、JSM を単なる表記方法にとどめず、専用のエディタを開発し、エディタ上で取り扱うことで利便性を高めた。

JSM editor 開発にあたり、コーディング（プログラミング）手法として以下の 3 方式を比較した。要件定義と各手法の比較結果を Table 3.1 に示す。

- (1) Microsoft C# WPF (Windows Presentation Foundation)
- (2) Microsoft Excel GUI (Graphical User Interface) + C# アドイン  
+ xlsx ファイル
- (3) Microsoft Excel GUI (Graphical User Interface) + C# アドイン  
+ jsme ファイル

比較の結果、製作時間が短縮でき、かつ必要な要件を実現できる見通しを得た (3) の方式を採用した。

Table 3.1 JSM editor Requirement definition

項	機能	JSM editor作成ツール		
		C# WPF	Excel GUI+C# アドイン+xmlsmファイル	Excel GUI+C# アドイン+jsmeファイル
1	新規, 読み込み, 閉じる, 保存	○(自作)	○ (Excel依存, 自作してテーブル構築が必要)	○(自作)
2	インポート, エクスポート	○(自作)	○(自作)	○(自作)
3	印刷	△(自作, 製作時間大, 要調査)	○ (Excel既存, Excelで可能なことに限定)	○ (Excel既存, Excelで可能なことに限定)
4	マージ	○(自作)	○(自作)	○(自作)
5	オプション	○(自作)	○(自作)	○(自作)
6	マニュアル, バージョン, 連絡先	○(自作)	○(自作)	○(自作)
7	Exit	○(自作)	× (Excelが閉じてしまう)	× (Excelが閉じてしまう)
8	切り取り, コピー, 貼り付け, 削除	○(自作)	○ (Excel既存, Excelで可能なことに限定)	○ (Excel既存, Excelで可能なことに限定)
9	Undo, Redo	△ (自作, 製作時間大, 要調査, 検討: 製作時間かけてもExcelと同じレベルまで引き上げられるか不明)	○ (Excel既存, Excelで可能なことに限定)	○ (Excel既存, Excelで可能なことに限定)
10	フィルター(行方向)	△(自作, 製作時間大) フィルタ状態は破棄される	○ (Excel既存, Excelで可能なことに限定) あらかじめシートに設定しておく フィルタ情報も保持可能	○ (Excel既存, Excelで可能なことに限定) ファイル読み込み時にアドインからシートにフィルタを設定可能か要検証 フィルタ状態は破棄される
11	Visible/Hidden(列方向)	○(自作)	○ (Excel既存, Excelで可能なことに限定)	○ (Excel既存, Excelで可能なことに限定)
12	検索, 置換	△(自作, 製作時間大)	○ (Excel既存, Excelで可能なことに限定)	○ (Excel既存, Excelで可能なことに限定)
13	エラーチェック	○(自作)	○(自作)	○(自作)
14	計算書出力	○(自作)	○(自作)	○(自作)
15	マウスクリックイベント (十字ハイライト)	○	xmlsmファイルなら, ブックマクロに記載して, アドインに通知可能	通常, 左クリックイベントはキャッチ不可能 実現可能か要検証
16	許可したエリアのみ編集	○	シートロックで可能か要検証	シートロックで可能か要検証
17	セル入力値チェック(I, 0のみなど)	○	× (Excelのペーストコマンドをフックできることが必要)	× (Excelのペーストコマンドをフックできることが必要)
18	リボン	○	○	○
19	WPFコントロール	○	○	○
20	マトリックスGUI	○ (Gridに縦書きできるか要検証。3つに分けるのでレイアウト考慮)	○ (マトリックスを生成したファイル)	△(ファイル読み込み時にアドインからシートを設定できるか要検証)
21	マトリックスGridの拡大縮小 (画面右下のカーボタン)	(要検証, 自作, 製作時間大)	○ (Excel既存, Excelで可能なことに限定)	○ (Excel既存, Excelで可能なことに限定)
22	アドイン自動読み込み	—	(要検証)	(要検証)
23	タブとグリッドの連動	○	(シートイベントとアドイン連携を要検証。)	(シートイベントとアドイン連携を要検証。)
24	初期設定の手間	exe, dllが同じ場所にあればサーバ上で可能	アドインを配布する必要あり	アドインを配布する必要あり
25	更新版の配布	ファイル置き換えのみ。サーバで使った場合, 誰かが使っているときは更新できない	各ユーザーにアドインDLLを配布する必要あり	各ユーザーにアドインDLLを配布する必要あり
26	ファイルダブルクリックで開く	○ (関連付けは手作業)	○ (Excelが開く)	× (Excelに関連付けてもOpen不可能)
27	複数ドキュメント	×(SDI, 別exeを立ち上げたときのクリップボードの挙動を要調査)	○(MDIなのでコピーアンドペースト可能)	○(MDIなのでコピーアンドペースト可能)
28	起動時間	exeの起動時間	Excelモジュール起動時間+アドインロード時間	ファイルOpen時間+アドインロード時間
29	開発環境	VS2015以降(リボン, zip対応のため) Expressで可能。	VS2015以降(リボン対応のため) ExpressはNG(アドイン作れない)	VS2015以降(リボン, zip対応のため) ExpressはNG(アドイン作れない)
30	ファイルサイズ	CSVファイルを圧縮するので, ファイルサイズは小さくできる	データをすべてExcelファイルとして持たせるので, ファイルサイズが大きくなる可能性有	CSVファイルを圧縮するので, ファイルサイズは小さくできる
31	Windows, Office	.Netがあれば, Win7でもWin10でも可能	Office2010用のアドインとして開発するので, 新バージョンで必ず動作するかは要検証	Office2010用のアドインとして開発するので, 新バージョンで必ず動作するかは要検証
32	評価	製作時間が必要であるが可能。 Excelの操作をそのまま模倣すること不可能(セルドラッグで移動など)	見た目はExcelなので, 比較的自由にできる Excelまかせにできる部分があるため製作時間は削減できる ただし, Excelがやってしまうこと(貼り付けなど)に対して, 追加のチェック機能などを入れられない	見た目はExcelなので, 比較的自由にできる Excelまかせにできる部分があるため製作時間は削減できる ただし, Excelがやってしまうこと(貼り付けなど)に対して, 追加のチェック機能などを入れられない

本研究で開発した JSM editor のサンプル画面を Fig. 3.7 に示す。また、リボン部の拡大図を Fig. 3.8 に示す。なお、動作推奨環境は以下である。

- ・ OS: Microsoft Windows 7
- ・ メモリ: 4[G byte]以上
- ・ CPU: 2.0[GHz]以上
- ・ モニタサイズ: 1920×1080 以上
- ・ ディスク容量: 2.0[Mbyte]以上の空き容量

(セットアップには 10[Mbyte]上の空き容量が必要)

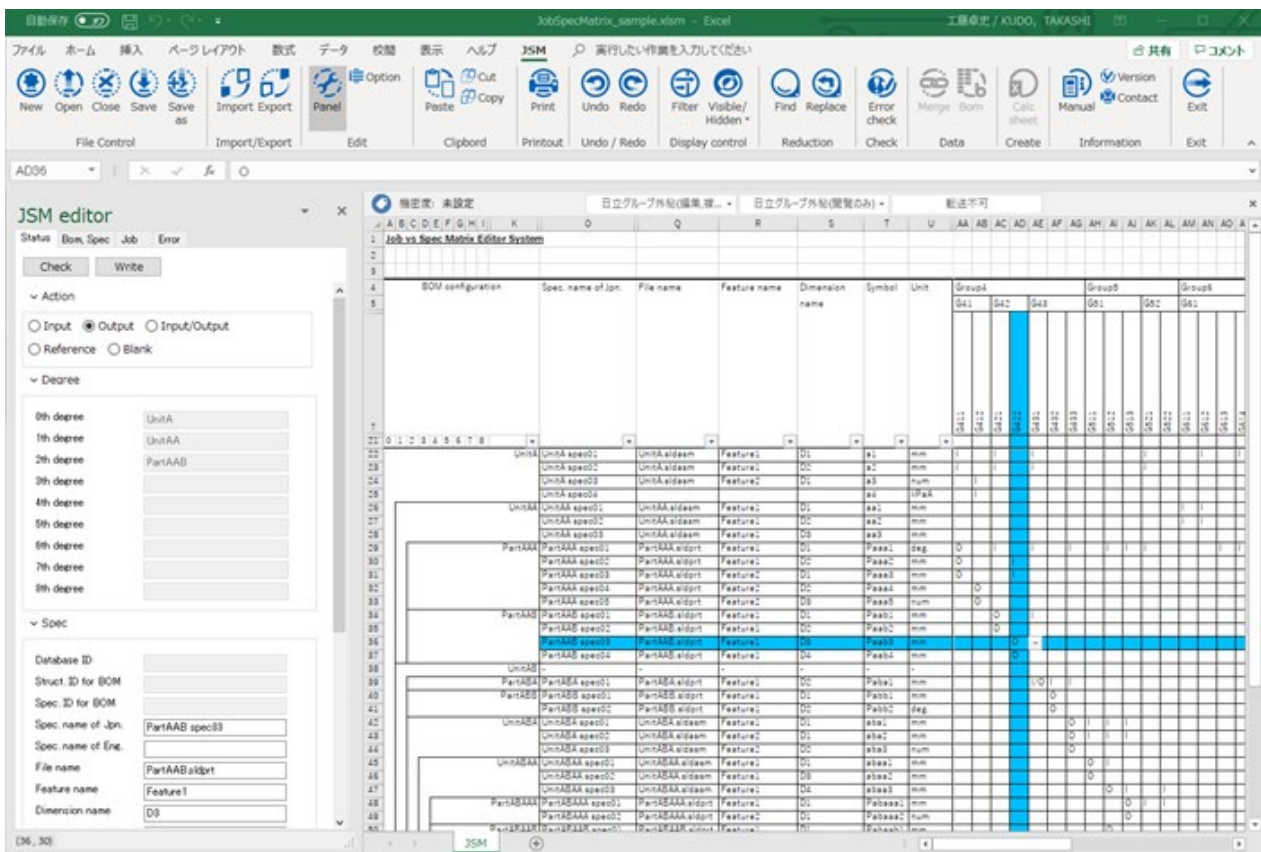
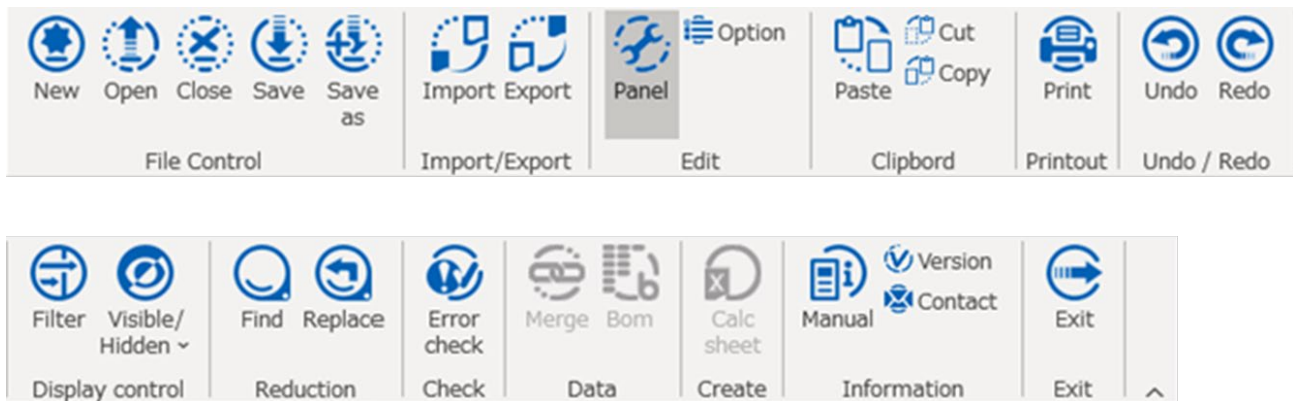


Fig. 3.7 Sample shot of JSM editor



**Fig. 3. 8 Ribbon Menu in JSM editor**

以下に JSM editor は以下の特徴的な機能を有す。

( 1 ) Error check

- ・ 3. 3. 3 項記載の記載内容不備を自動チェック
- ・ Spec 名称の重複チェック
- ・ BOM コードの重複チェック

( 2 ) Bom ( 管理者のみ実行可能 )

- ・ BOM コードの自動付番
- ・ システムコードの自動付番 ( システムコードは内部管理用コードのため、実設計者 ( オペレータ ) は意識する必要はない。 )

( 3 ) Merge ( 管理者のみ実行可能 )

- ・ 複数 JSM を一つのファイルにマージ

( 4 ) Export

- ・ 設計システム用業務フロー自動生成出力
- ・ 設計システム用 BOM 構造の自動生成出力
- ・ 設計システム用 JSM マトリクス部データ自動生成出力



なお、出力されるファイルは以下となる。

No.	固定ファイル名	内容
1	t_bom_master csv	BOM 構造定義
2	t_property_master csv	BOM 構造用付属情報
3	t_calc_sheet_master csv	業務フロー
4	t_data csv	JSM マトリクス部データ

( 5 ) Import

- ・ 設計システムから JSM に関するデータを読み込む
- ・ 読み込むファイルは ( 4 ) と同一である。

( 6 ) Calc sheet

- ・ 外部ソフトウェア用入出力シート自動生成出力  
詳細は 4. 2. 3 項で説明する。

( 7 ) 3D-CAD 連携 ( Fig. 3.7 Ribbon Menu にはアイコン未表示 )

- ・ 指定した 3D-CAD ファイル内のパラメータおよびそのアドレス情報を取得し、JSM の Spec にインポートする。
  - インポートされるパラメータ
    - ・ すべての長さ寸法
    - ・ すべての角度寸法
    - ・ すべての数量値
  - 各パラメータのアドレス情報
    - ・ ファイル名
    - ・ フィーチャー名
    - ・ 寸法名

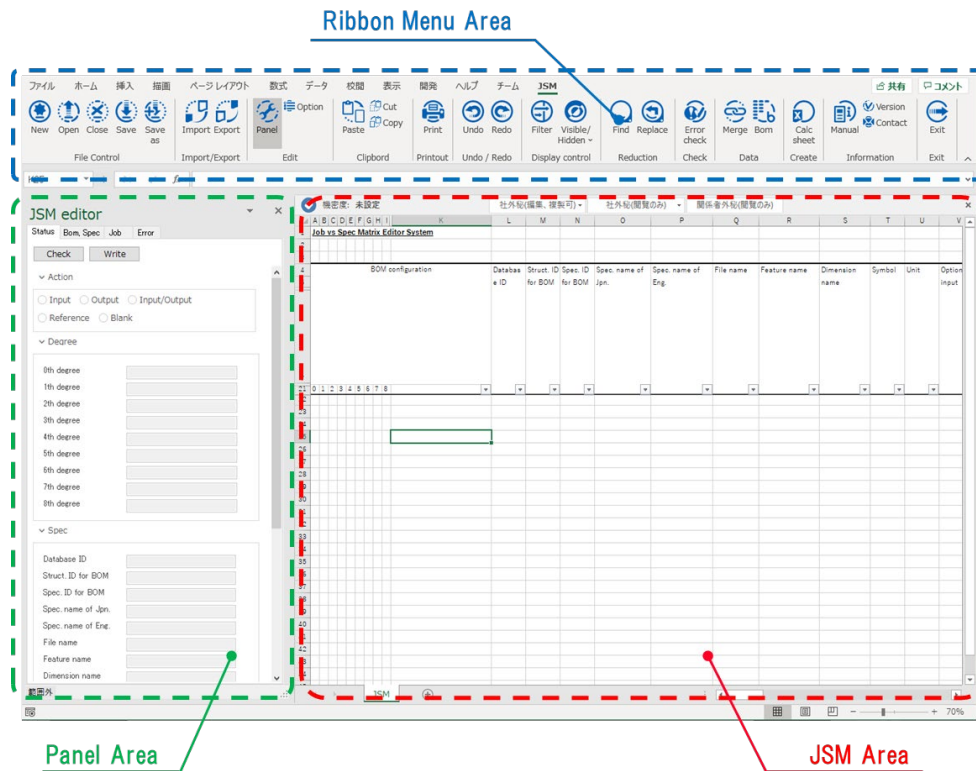
備考:

- ・ 指定した 3D-CAD ファイルにアセンブリファイルを指定した場合は、アセンブリファイルに含まれるすべてのパーツファイル情報に加え、アセンブリファイル本体に含む情報も取得する。
- ・ 現時点で本機能は 3D-CAD として SolidWorks バージョン 2014～2017 のみ対応している。

その他下記一般的な機能も有す。

- ・ ファイル操作
  - 新規作成
  - オープン/クローズ
  - 上書保存/指定保存
- ・ コピー/カット/ペースト
- ・ アンドゥ/リドゥ
- ・ フィルタ
- ・ 表示/非表示
- ・ 検索/置換
- ・ プリントアウト

次に JSM editor のユーザインタフェースについて説明する。JSM editor のユーザインタフェースは大きく、リボンメニュー部、JSM 部およびパネル部の 3 つに分けられる。Fig. 3.9 に各部の配置を示す。



**Fig. 3. 9 User Control Area Layout of JSM editor**

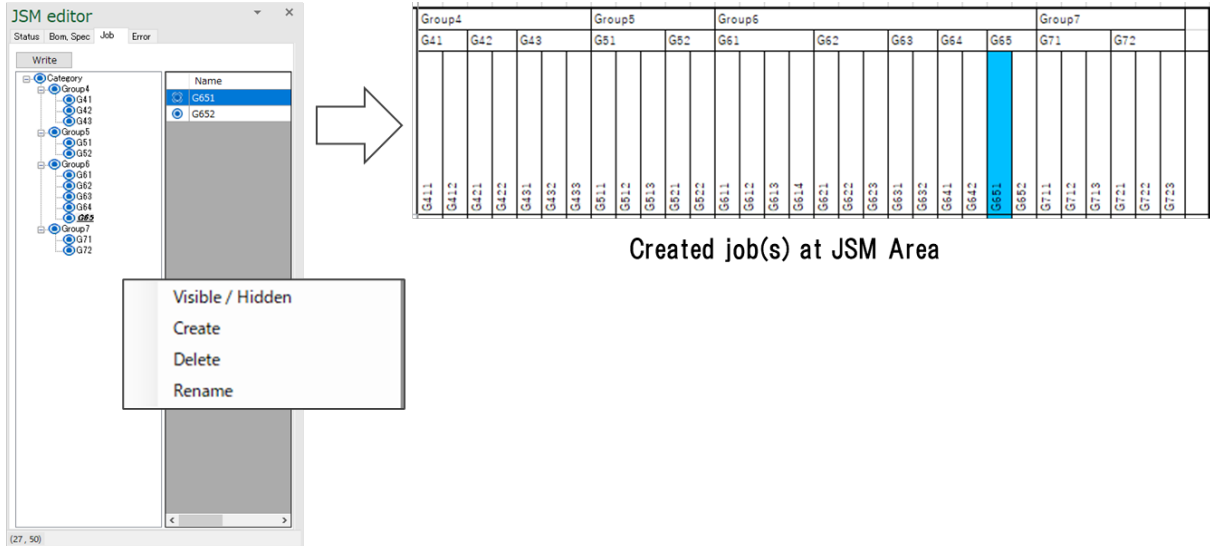
リボンメニュー部は Fig. 3. 8 で説明した機能を提供するアイコン（ボタン）が配置されている。JSM 部は行列の表形式となっており、3. 3. 2 節で説明した JSM を直接ここに記載することが可能である。パネル部は JSM 部への記載をサポートする。以下、パネル部について説明する。

パネル部には以下に示す 4 つのタブが設けてあり、それぞれ個別の機能を提供する。

- (A) Job パネル
- (B) Bom, Spec パネル
- (C) Status パネル
- (D) Error パネル

(A) Job パネル

Job パネルでは Job の構成をまとめて作成することが可能である。Fig. 3.10 に例を示す。



Job(s) Configuration on Job Panel

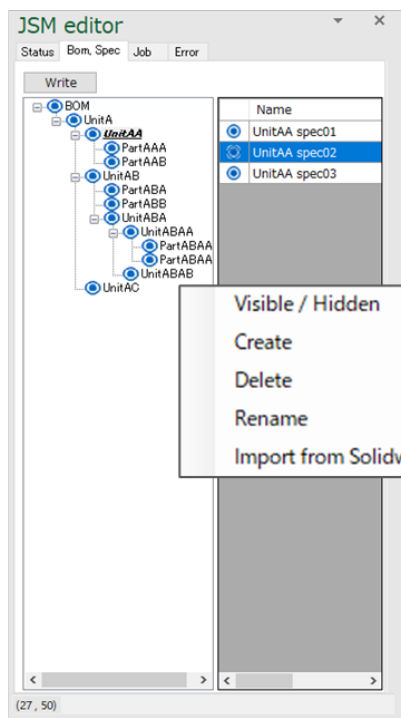
Fig. 3.10 Example of Creation of Job Configuration

Fig. 3.10 に示すように、Job は 2 段までのグループで構成することができる。また、ドラッグ・アンド・ドロップでグループをまたいで移動することも可能である。また、個々の Job に対して以下の操作が可能である。

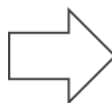
- ・ 表示/非表示切り替え (Visible/Hidden)
- ・ 新規作成 (Create)
- ・ 削除 (Delete)
- ・ 名称変更 (Rename)

(B) Bom, Spec パネル

Job パネルでは Spec の構成をまとめて作成することが可能である。Fig. 3.11 に例を示す。



Spec(s) Configuration on Bom, Spec Panel



UnitA	000001	UnitA spec01
	000002	UnitA spec02
	000003	UnitA spec03
	000004	UnitA spec04
UnitAA	000005	UnitAA spec01
	000006	UnitAA spec02
	000007	UnitAA spec03
PartAAA	000008	PartAAA spec01
	000009	PartAAA spec02
	000010	PartAAA spec03
	000011	PartAAA spec04
	000012	PartAAA spec05
PartAAB	000013	PartAAB spec01
	000014	PartAAB spec02
	000015	PartAAB spec03
	000016	PartAAB spec04
UnitAB	-	-
PartABA	000017	PartABA spec01
PartABB	000018	PartABB spec01
	000019	PartABB spec02
UnitABA	000020	UnitABA spec01
	000021	UnitABA spec02
	000022	UnitABA spec03
UnitABAA	000023	UnitABAA spec01
	000024	UnitABAA spec02
	000025	UnitABAA spec03
PartABAAA	000026	PartABAAA spec01
	000027	PartABAAA spec02
PartABAAB	000028	PartABAAB spec01
	000029	PartABAAB spec02
	000030	PartABAAB spec03
UnitABAB	000031	UnitABAB spec01
	000032	UnitABAB spec02
UnitAC	-	-

Created Spec(s) at JSM Area

Fig. 3. 11 Example of Creation of Spec Configuration

Spec は機械製品システムのユニット（アセンブリ）構成を考慮して 9 段までのグループで構成することができる。また、ドラッグ・アンド・ドロップでグループを跨いで移動することも可能である。また、個々の Spec に対して以下の操作が可能である。

- ・ 表示/非表示切り替え（Visible/Hidden）
- ・ 新規作成（Create）
- ・ 削除（Delete）
- ・ 名称変更（Rename）
- ・ Import from Solidworks

なお、本節冒頭で説明したとおり、一つの機械製品システムが取り扱う Spec 数は数千～数十万におよび、これらは 3D モデルの寸法値がほとんどである。これらの値を個々に 3D モデルから拾い出しマニュアルで Spec として記載することは

現実的ではない。そこで、本研究では、ExcelのVBA（Visual Basic Applications）より3D-CADのAPI（Application Programming Interface）を介して3Dモデルから全寸法値を取得する方法を採用した。なお、本機能はJSM editorの機能ではなく、Specを作成する際の前準備として行うものである。Fig. 3.12にExcelの画面を示す。また、Fig. 3.13に取得した寸法値のリスト画面を示す。寸法値のリストにはJSMで使用しないデータも含まれているがこれは目視確認に用いているだけである。

JSM editorのBom, Specパネル画面では、Bom, Specパネル上に本リストを読み込む機能を有しており、数千の3Dモデル寸法のSpecを1min.程度で作成することができる。

	A	B	C	D	E	F	G
1					関数名	処理内容	入出力シート
2	Method	SetDimensionAssemPart			GetDimensionAssemPart	アセンブリに含まれるパーツの全寸法リストを取得する	DimAssemf
3	Sheet	DimAssemPart			SetDimensionAssemPart	アセンブリに含まれるパーツの寸法リストを設定する	DimAssem
4	File	C:\Users\XXXXXXXX\Desktop\保管データ\200_データツール\200_Tools\200_SolidworksTools\As			GetDimensionAssem	アセンブリの寸法を取得する	DimAssem
5	input1				SetDimensionAssem	アセンブリの寸法を設定する	DimAssem
6	input2				GetDimensionPart	パーツの寸法を取得する	DimPart
7	input3				SetDimensionPart	パーツの寸法を設定する	DimPart
8	input4				GetWeightPart	パーツの重量を取得する	WeightPar
9	input5				GetSketchAssem	アセンブリのスケッチを取得する	SketchAsse
10	Status	Finish			GetSketchPart	パーツのスケッチを取得する	SketchPart
11					OpenFile	ファイルを開く	
12		Exec			MateSketch	スケッチ合致を実行(Assem-Partのみ)	SketchList
13					UnSuppressPart	指定されたアセンブリ内のパーツを表示(抑制解除)する	PartList
14					MakeMaterialTable	材料テーブルを作成する	MaterialTab
15					SetConfiguration	表示するコンフィグレーションを指定する	Configuratic
16					SetCompConfigProperties	アセンブリ指定で配下のパーツなどのコンフィグレーションを指定	Configuratic
17					SetHoleWizard	穴ワイザード	HoleWiz
18							
19							

Fig. 3. 12 Interface of 3D model Dimension Get Program

#	A	B	C	D	E	F	G	H	I	J	
1	FilePath	FullName	DimName	FeatureName	FileName	ParamType	ParamTypeS	DrivenState	ReadOnly	SystemValue	Conver
2	Part1.SLDFRT	Dr@Center@Part1.Part	Dr	Center	Part1.Part	0	Linear	Driving	FALSE	2.5	
3	Part1.SLDFRT	D2@Center@Part1.Part	D2	Center	Part1.Part	0	Linear	Driving	FALSE	4	
4	Part1.SLDFRT	ang@Center@Part1.Part	ang	Center	Part1.Part	1	Angular	Driving	FALSE	2.79E+00	
5	Part1.SLDFRT	rs@Center@Part1.Part	rs	Center	Part1.Part	0	Linear	Driven	FALSE	0.555407289	
6	Part1.SLDFRT	D7@Center@Part1.Part	D7	Center	Part1.Part	0	Linear	Driven	FALSE	3.939231012	
7	Part1.SLDFRT	D1@Center@Part1.Part	D1	Center	Part1.Part	1	Angular	Driven	FALSE	1.570796327	
8	Part1.SLDFRT	D4@Center@Part1.Part	D4	Center	Part1.Part	1	Angular	Driven	FALSE	1.570796327	
9	Part1.SLDFRT	rl@Center@Part1.Part	rl	Center	Part1.Part	0	Linear	Driving	FALSE	0.75	
10	Part1.SLDFRT	D3@Center@Part1.Part	D3	Center	Part1.Part	0	Linear	Driving	FALSE	1.5	
11	Part1.SLDFRT	D1@Moved@Part1.Part	D1	Moved	Part1.Part	1	Angular	Driving	FALSE	0.104719755	
12	Part1.SLDFRT	ang@Moved@Part1.Part	ang	Moved	Part1.Part	1	Angular	Driving	FALSE	2.792526803	
13	Part1.SLDFRT	st@Moved@Part1.Part	st	Moved	Part1.Part	0	Linear	Driven	FALSE	0.176557885	
14	Part1.SLDFRT	φ@Moved@Part1.Part	φ	Moved	Part1.Part	1	Angular	Driven	FALSE	0.237459706	
15	Part1.SLDFRT	sl@bell-crank_pin_section@Part1.Part	sl	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.215	
16	Part1.SLDFRT	gl@bell-crank_pin_section@Part1.Part	gl	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.0305	
17	Part1.SLDFRT	dl@bell-crank_pin_section@Part1.Part	dl	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.52	
18	Part1.SLDFRT	ll@bell-crank_pin_section@Part1.Part	ll	bell-crank_pin_sec	Part1.Part	0	Linear	Driven	FALSE	3.05E-01	
19	Part1.SLDFRT	h@bell-crank_pin_section@Part1.Part	h	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.465	
20	Part1.SLDFRT	d@bell-crank_pin_section@Part1.Part	d	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.22	
21	Part1.SLDFRT	l2@bell-crank_pin_section@Part1.Part	l2	bell-crank_pin_sec	Part1.Part	0	Linear	Driven	FALSE	0.4895	
22	Part1.SLDFRT	m@bell-crank_pin_section@Part1.Part	m	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.04	
23	Part1.SLDFRT	n@bell-crank_pin_section@Part1.Part	n	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.06	
24	Part1.SLDFRT	d1@bell-crank_pin_section@Part1.Part	d1	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	2.00E-01	
25	Part1.SLDFRT	d2@bell-crank_pin_section@Part1.Part	d2	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.28	
26	Part1.SLDFRT	o@bell-crank_pin_section@Part1.Part	o	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.53	
27	Part1.SLDFRT	D3@bell-crank_pin_section@Part1.Part	D3	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.58	
28	Part1.SLDFRT	D4@bell-crank_pin_section@Part1.Part	D4	bell-crank_pin_sec	Part1.Part	0	Linear	Driving	FALSE	0.35	
29	Part1.SLDFRT	d2@guide-link_rod_pin_section@Part1.F	d2	guide-link_rod_pin	Part1.Part	0	Linear	Driving	FALSE	0.09	
30	Part1.SLDFRT	h2@guide-link_rod_pin_section@Part1.F	h2	guide-link_rod_pin	Part1.Part	0	Linear	Driving	FALSE	0.0905	
31	Part1.SLDFRT	t@guide-link_rod_pin_section@Part1.Pa	t	guide-link_rod_pin	Part1.Part	0	Linear	Driving	FALSE	0.08	
32	Part1.SLDFRT	d1@servomotor_rod_pin_section@Part1.d	d1	servomotor_rod_p	Part1.Part	0	Linear	Driving	FALSE	0.1	
33	Part1.SLDFRT	h1@servomotor_rod_pin_section@Part1.h	h1	servomotor_rod_p	Part1.Part	0	Linear	Driving	FALSE	0.0905	
34	Part1.SLDFRT	t@servomotor_rod_pin_section@Part1.it	t	servomotor_rod_p	Part1.Part	0	Linear	Driving	FALSE	0.08	
35	Part1.SLDFRT	D1@lever_motor_side@Part1.Part	D1	lever_motor_side	Part1.Part	0	Linear	Driving	FALSE	0.2	
36	Part1.SLDFRT	L@lever_motor_side@Part1.Part	L	lever_motor_side	Part1.Part	0	Linear	Driven	FALSE	0.575	
37	Part1.SLDFRT	W2@lever_motor_side@Part1.Part	W2	lever_motor_side	Part1.Part	0	Linear	Driven	FALSE	0.316586912	
38	Part1.SLDFRT	D1@lever_ring_side@Part1.Part	D1	lever_ring_side	Part1.Part	0	Linear	Driving	FALSE	0.18	
39	Part1.SLDFRT	L@lever_ring_side@Part1.Part	L	lever_ring_side	Part1.Part	0	Linear	Driven	FALSE	0.380407289	
40	Part1.SLDFRT	W2@lever_rino_side@Part1.Part	W2	lever_rino_side	Part1.Part	0	Linear	Driven	FALSE	0.299969385	

Fig. 3. 13 Example of 3D model Dimension List

(C) Status パネル

Job パネルでは Spec の構成をまとめて作成・確認・編集することが可能である。Fig. 3.13 は Job 部において，Spec “000008” が Job “G421” の入力と指定している部位を選択している例である。なお，Fig. 3.14 において，Spec “000008” の File name, Feature name および Dimension name は，画面上により多くの Matrix 部を表示するためにユーザの意思で非表示としており，値を画面上で確認することができない。このようにいくつかの情報を非表示とすることは，Spec, Job が多い場合よく行われる作業である。

BOM configuration	Spec. ID for BOM	Spec. name of Jpn.	Group4			Group5		G61
			G41	G42	G43	G51	G52	
UnitA	000001	UnitA spec01						
	000002	UnitA spec02						
	000003	UnitA spec03						
	000004	UnitA spec04						
UnitAA	000005	UnitAA spec01						
	000006	UnitAA spec02						
	000007	UnitAA spec03						
PartAAA	000008	PartAAA spec01	O					
	000009	PartAAA spec02	O					
	000010	PartAAA spec03	O					
	000011	PartAAA spec04	O					
	000012	PartAAA spec05	O					
PartAAB	000013	PartAAB spec01		O				
	000014	PartAAB spec02		O				
	000015	PartAAB spec03		O				
	000016	PartAAB spec04		O				
UnitAB	-	-						
PartABA	000017	PartABA spec01			I/O			
PartABB	000018	PartABB spec01			O			
	000019	PartABB spec02			O			
UnitABA	000020	UnitABA spec01			O			
	000021	UnitABA spec02			O			
	000022	UnitABA spec03			O			
UnitABAA	000023	UnitABAA spec01				O		
	000024	UnitABAA spec02				O		
	000025	UnitABAA spec03				O		

Fig. 3. 14 Example of matrix data selection at Job area

この時の Status パネルの表示を Fig. 3. 15 に示す。Status パネルには、Fig. 3. 14 では非表示となっていた Spec “000008” の File name, Feature name および Dimension name を含め、選択された部位のすべてのデータが表示されており、Job 部では非表示であった情報を即座に確認することができる。また、機械構成を表す Degree および Bom code, 計算書の ID を除く項目は本パネルにて編集することが可能である。



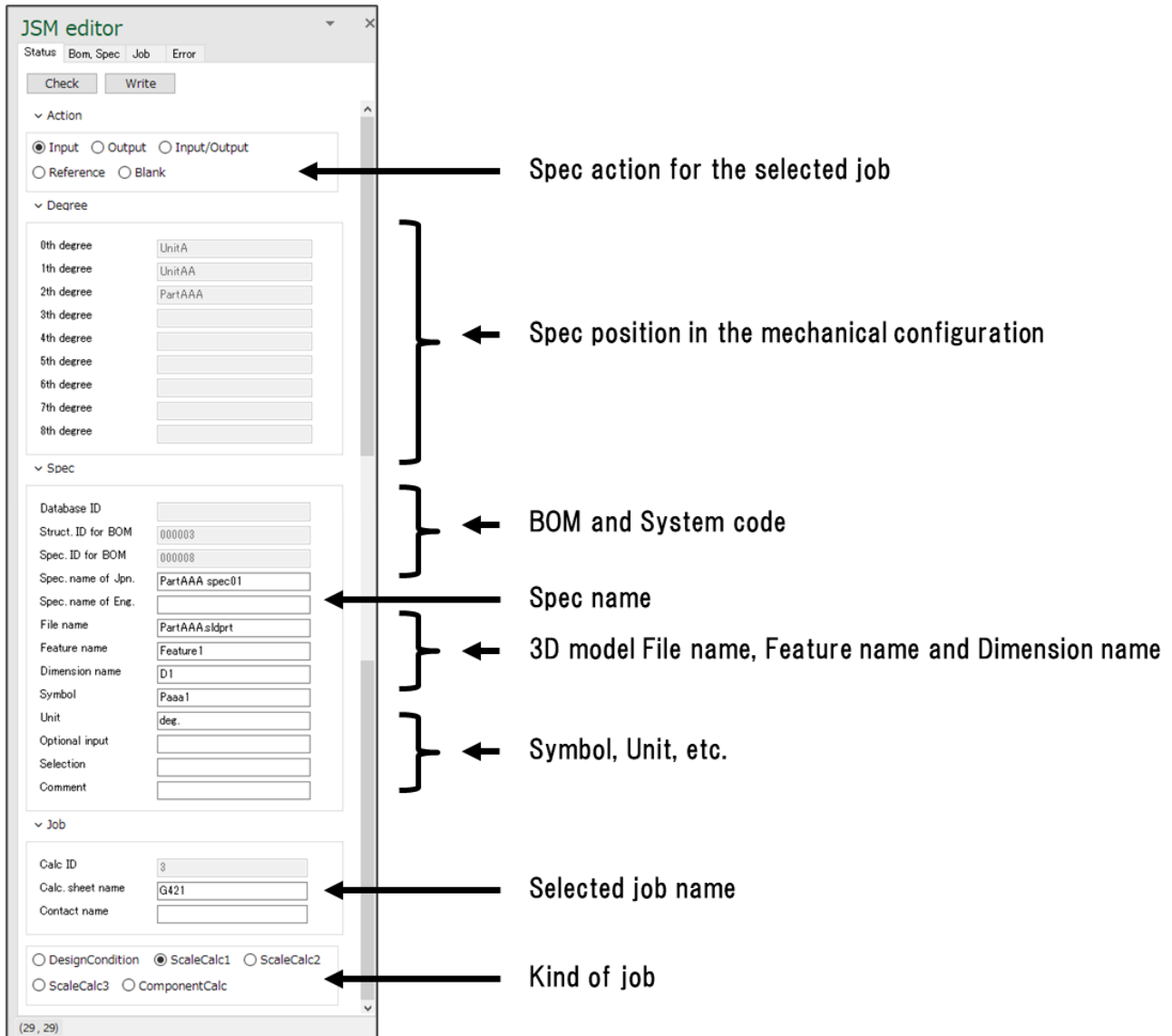


Fig. 3. 15 Example of status panel

(D) Error パネル

Error パネルでは JSM 内の記載不備を自動で検出し確認することができる。記載不備は 3.3.2 項で説明した内容の他、Spec name や Job name がいない場合、重複している場合なども含む。

Fig. 3. 16 にエラーを検出した例を示す。

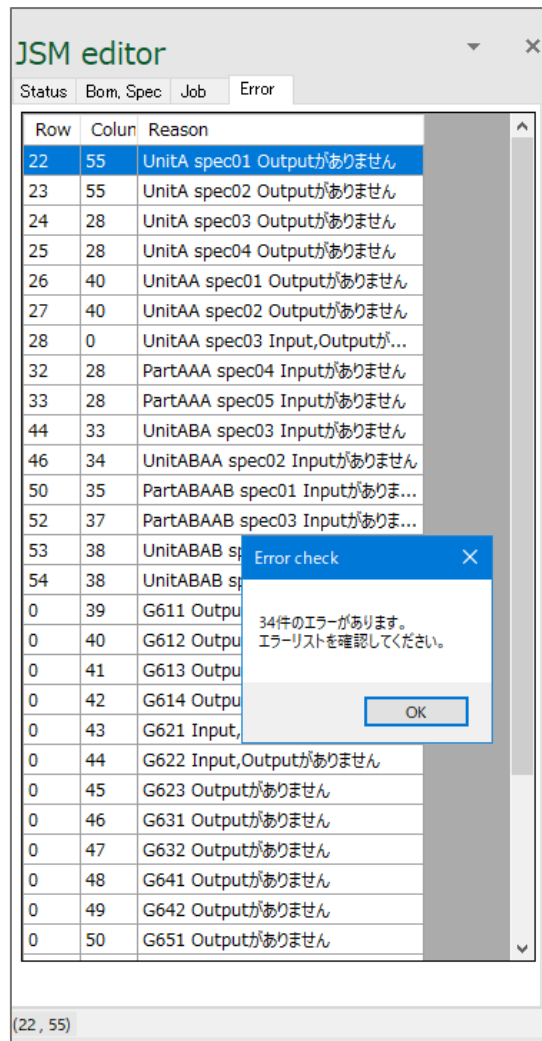


Fig. 3. 16 Example of Error list

### 3. 3 まとめ

第 3 章では，第 2 章で説明した課題の解決方法として，汎用的な設計システムとしての 1DCAE Virtual Engineering 環境においては「積上げ設計の排除」，「構成要素の独立」，「インタフェースの統一」が必要であると考え，その設計概念を示した。また，プラットフォームのトレンド追従においては，入出力値群を俯瞰して定義・管理する手法として JSM を考案した。なお，JSM では，チェックルールを構築するとともに JSM を作成・管理を容易とする JSM editor を制作し多くの Job，Spec を取り扱うことを可能とした。

## 第 4 章

発電用水車を対象とする実際の Virtual Engineering 設計環境

## 第 4 章 発電用水車を対象とする実際の Virtual Engineering 設計環境

本章では，まず製品システムとしての発電用水車の説明を行い，次いで第 3 章にて開発した内容を取り込んだ実際の Virtual Engineering プラットフォームについて説明する。

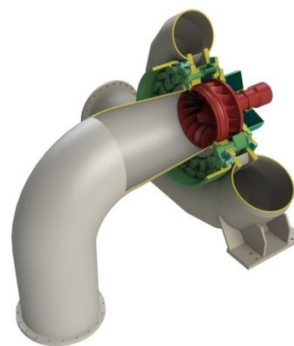
### 4.1 発電機用水車

#### 4.1.1 概要

本研究が対象とする機械製品システムは発電機用水車である。水車とは，原理的に水のエネルギーを機械エネルギーに変換する原動機である。水のエネルギーには位置エネルギー，速度エネルギーおよび圧力エネルギーがあり，それらを水柱高さで表示すると位置水頭（Potential Head），速度水頭（Velocity Head）および圧力水頭（Pressure Head）となる。水車が利用するエネルギーはこれらの組み合わせであり，それぞれをどのような状態で活用するかによって水車は異なる型式となる。本研究が対象とする水車はフランシス水車と呼ばれる主に位置水頭を機械エネルギーに変換する水車である。出力は数十～数百 kW のクラスをターゲットとしている。また，Fig. 4.1 に示すように，フランシス水車には立型（縦型）と横型があるが，本研究がターゲットとする出力クラスでは設置面積が少なくなることから立型が多く用いられるため，横型はターゲットとしていない。



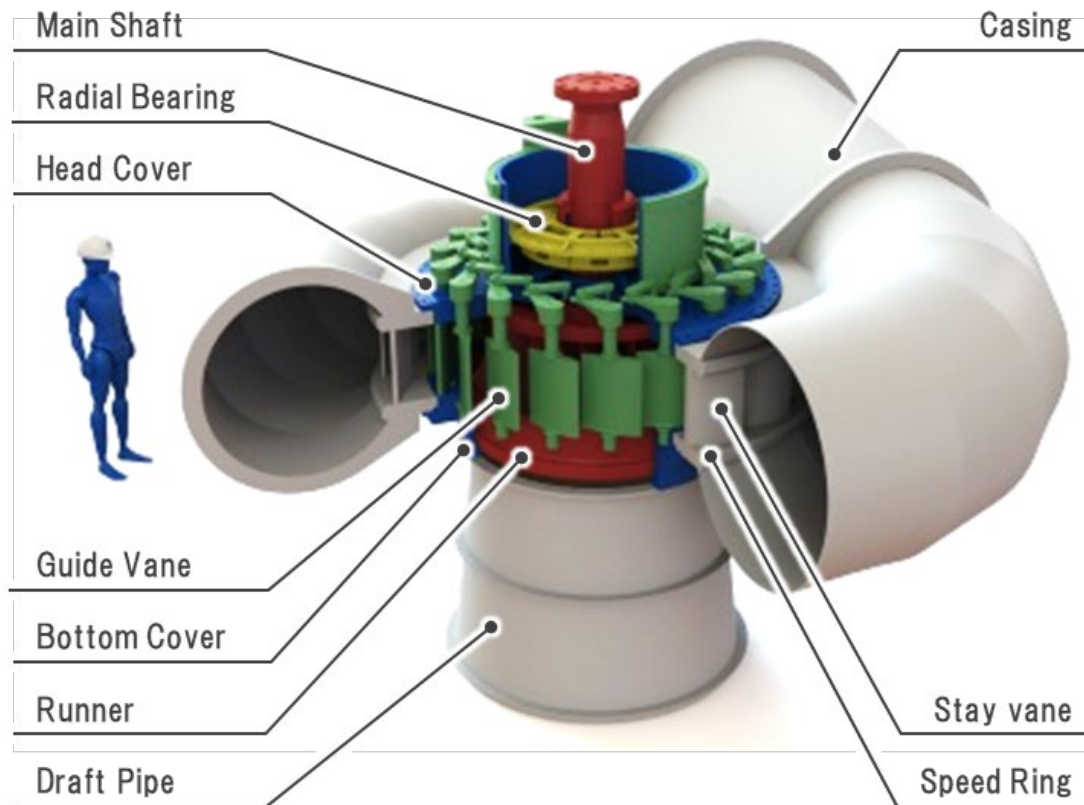
Vertical Type



Horizontal Type

Fig. 4.1 Layout Type of Francis Turbine (3D Cut Model)

立型フランシス水車の構造を Fig. 4. 2 に示す。



**Fig. 4. 2 Typical Outline Drawing of Francis Turbine (3D Cut Model)**

立型フランシス水車の主要な構成部品の説明を以下に示す。なお、主軸の上部には発電機が接続されるが、Fig. 4. 2 においては省略している。

(1) ケーシング (Casing)

渦巻形の構造で、その入口で取水した水を内周部から均等に羽根車に導く。

(2) ステイベーン (Stay Vane)

ケーシングより導かれた水を整流する。

(3) スピードリング (Speed Ring)

ケーシングと上下カバーを接続すると共にステイベーン上下の流路を形成する。

(4) ガイドベーン (Guide Vane)

羽根車に流入する水の量を調整する。

(5) 羽根車 (Runner)

ガイドベーンからの流水をうけ回転し、その回転エネルギーを主軸に伝える。

(6) 主軸 (Main Shaft)

羽根車の回転エネルギーを発電機に伝える。

(7) 軸受 (Radial Bearing)

主軸のラジアル荷重を支える。なお、水車側の軸受は本軸受のみであり、本軸受に加えて、発電機側にラジアル荷重を支える軸受2機およびスラスト荷重を支える軸受1機が設置されることが通常である。

(8) 上カバー (Head Cover)

羽根車上側のカバー。ガイドベーンやその動作機構、軸受等多くの機構部品が搭載されるベースを兼ねる。

(9) 下カバー (Bottom Cover)

羽根車下側のカバー。

(10) 吸出し管 (Draft Pipe)

羽根車を通過し、エネルギーを失った水を水車外部へ導く。

ここで、(4)のガイドベーンの役割を補足する。発電機用水車においては、主軸と発電機の軸は直接結合されており、羽根車の回転数はそのまま発電される電気の周波数となる。発電機の負荷は電力の使用量によって逐次変化する。このとき羽根車に流入する水の量が一定のままでは羽根車の回転数が変動し発電される電気の周波数も変動してしまう。そこで、発電機用水車では、ガイドベーンの開度にフィードバックをかけ、羽根車に流入する水の量を調整することで羽根車の回転数が一定となるように制御を行っている。

#### 4.1.2 比速度

機能開発技術の一つである設計ルールの例として、発電機用水車の基本寸法の計算式を示す。

発電機用水車の基本寸法は落差  $H$  (Head, 単位: m), 出力  $P$  (Power, 単位: kW), 回転速度  $N$  (Speed, 単位:  $\text{min}^{-1}$ ) から決定され, 基本的に式 (4.1) に示す比速度 (Specific Speed) :  $N_s$  が同じとなる水車においては, 羽根車の流水面形状はサイズこそ異なるが相似となる。

$$N_s = N \frac{\sqrt{P}}{H^4} \dots\dots\dots (4.1)$$

$N_s$ : Specific speed, 単位: m-kW

$N$ : Speed, 単位:  $\text{min}^{-1}$

$P$ : Power, 単位: kW

$H$ : Head, 単位: m

なお, 式 (4.1) 記載比速度の単位は無次元数ではなく, 正しくは式 (4.2) となる。しかしながら, 複雑であることから省略し, 式 (4.1) で”m-kW”と示したように, 計算に用いた落差と出力の単位を併記することが慣例となっている<sup>(34)</sup>。

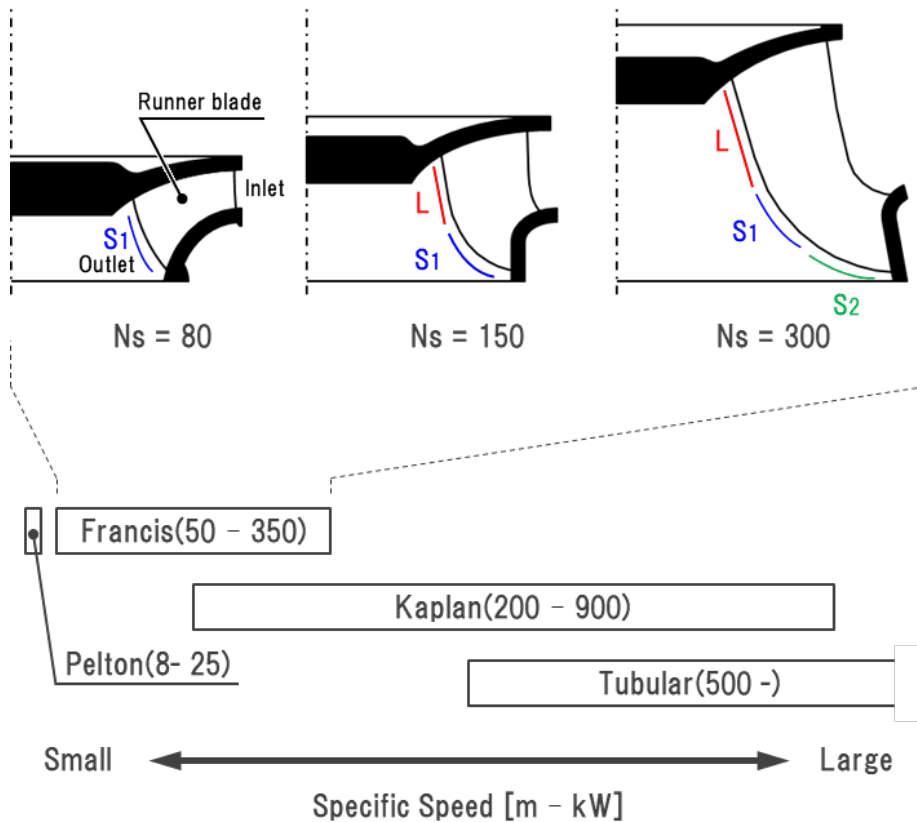
$$\frac{\sqrt{10}}{6} \text{kg}^{\frac{1}{2}} \text{m}^{-\frac{1}{4}} \text{s}^{-\frac{5}{2}} \dots\dots\dots (4.2)$$

ここで, 4.1.1 項で説明したとおり, 羽根車の回転速度はそのまま発電される電気の周波数となるため, 回転数 ( $N$ ) は Table 4.1 に示す値を使用することとなる。発電機のポール数 (極数) は発電機用水車の出力から決まるため, 回転速度は出力から一位的に決まると考えてよい。

Fig. 4.3 に比速度と水車の主要部品である羽根車の断面形状の関係を示す。

**Table 4. 1 Rotate Speed of Water Turbine**

Electric Generator Pole Number [-]	Turbine Rotate Speed [min. <sup>-1</sup> ]	
	for 50[Hz]	for 60[Hz]
6	1,000	1,200
8	750	900
10	600	720
12	500	600
14	429	514
16	375	450
18	333	400
20	300	360
24	250	300
28	214	257
32	188	225
36	167	200
40	150	180
48	125	150
56	107	129
64	94	113
72	83	100
80	75	90
88	68	82



**Fig. 4. 3 Water Turbine Type and Runner Profile of Francis on Specific Speed**



Fig. 4. 3 左側に図示する低比速度の羽根車は高落差地点に適用され羽根車は偏平な形をしている。対して右側に図示する高比速度の羽根車は低落差地点に適用され、羽根車の入り口側の高さ寸法である呑口高さが高い。このため、高比速度の羽根車においては、羽根車外周に設けられる案内羽根（Guide Vane）と固定羽根（Stay Vane）の背丈も高い。また、高落差で水圧が高い水車においては、それを構成する部品、特に羽根車が回転する羽根車室の上カバー（Head Cover）は概して厚く剛性が高い。このように、フランシス水車の中でも比速度によって部品形状・構成に大きな違い、つまりバラエティがある。

以下、Fig. 4. 3 図示の羽根車の羽形状を例に用いて 3D Model のバラエティについて説明する。本システムは 3D-CAD を中心としていることから、必然的に使用している 3D-CAD ソフトウェアのパラメトリック機能が単一の部品ファイル内で幾何学的形状を表現できるかがバラエティ分類の要否を決める。Fig. 4.3 において、左側に図示する低比速度羽根車の羽（Runner blade）出口側（outlet）の平面への投影形状は一つの円弧（S1）で表すことができる。対して、中央および右側図示の中比速度および高比速度羽根車は、それぞれ一つの直線（L）と一つの円弧（S1）、一つの直線（L）と二つの円弧（S1, S2）で表される。本稿で使用した 3D-CAD を含め、多くの 3D-CAD では寸法値に「0（ゼロ）」を入力することはできないため、例えば直線（L）の有無をパラメトリック機能で表現することはできない。したがって、このような場合はバラエティとして複数の 3D Model を用意する必要がある。なお、本稿で使用した 3D-CAD ではパラメトリック機能とは別の機能を用いてこれらを単一の 3D Model で表現することも可能であったが、特に実際の設計者に不慣れな機能であることを理由に使用を避けた。

Fig. 4. 3 に示した通り、フランシス水車羽根車は比速度 50 – 350 [m-kW] の広い範囲を有する。450 種類の過去実績を調査した結果、羽根車の羽形状は、寸法値を含めると、一つとして同じ形状はなかった。しかし、パラメトリック機能を用いることから寸法値を考慮せず幾何学的形状の構成要素のみで整理すると、わずか 12 種類のバリエーションで表現可能であった。

また，水車全体のサイズは，羽根車羽根外径： $D_1$  に依存する。 $D_1$  は，比速度に伴って増加するランナの周速係数： $\phi$  より導くことができる。式（4.3）に周速係数の式を示す。

$$\phi = \frac{\pi D_1 N}{60\sqrt{2gH}} \dots\dots\dots (4.3)$$

$g$ : Gravitational acceleration [m/s<sup>2</sup>]

### 4.1.3 部品

Table 4.2 に立型フランシス水車部品構成の一部を示す。

**Table 4.2 Typical Vertical Francis Turbine Components**

Turbine type	Unit	Sub Unit	Parts	Remarks
Vertical Francis Turbine	Rotate unit	Runner unit	Runner	-
			Runner Corn	-
			Runner Key	-
		Main Shaft Unit	Main Shaft	-
			Flange Liner	-
			Shaft Sleeve	-
			Water protect ring	-
			Bearing Liner	-
		Stay Unit	Casing Unit	Casing
	Head Cover Unit		Head Cover	-
			Head Cover Liner	-
			Upper Sheet Liner	-
			Upper Protect Liner	-
			Main Shaft Liner	-
	Sheeling Box	Include Seal		
...	...	...		

本研究がターゲットとする立型フランシス水車の主要部品数（Table 4.2 における Parts の種類）は近年の設計実勢を精査した結果，最大公約数で 51 種類となった。ただしこの数値はネジ，ナット等の鋳螺類に代表される小物部品は含まれていない。これは 1.1.1 項で述べた通り，1DCAE の段階においては機能ベースで対象システム全体を表現することを目的としているため，3D モデルを簡易なものにとどめていることによる。

## 4.2 概要

### 4.2.1 フロー図

Fig. 4.4 に筆者らが作成した発電用水車 1DCAE Virtual Engineering 環境フロー図を示す。なお、Engineering System は図が煩雑になるため Fig. 4.4 中に記載していない。

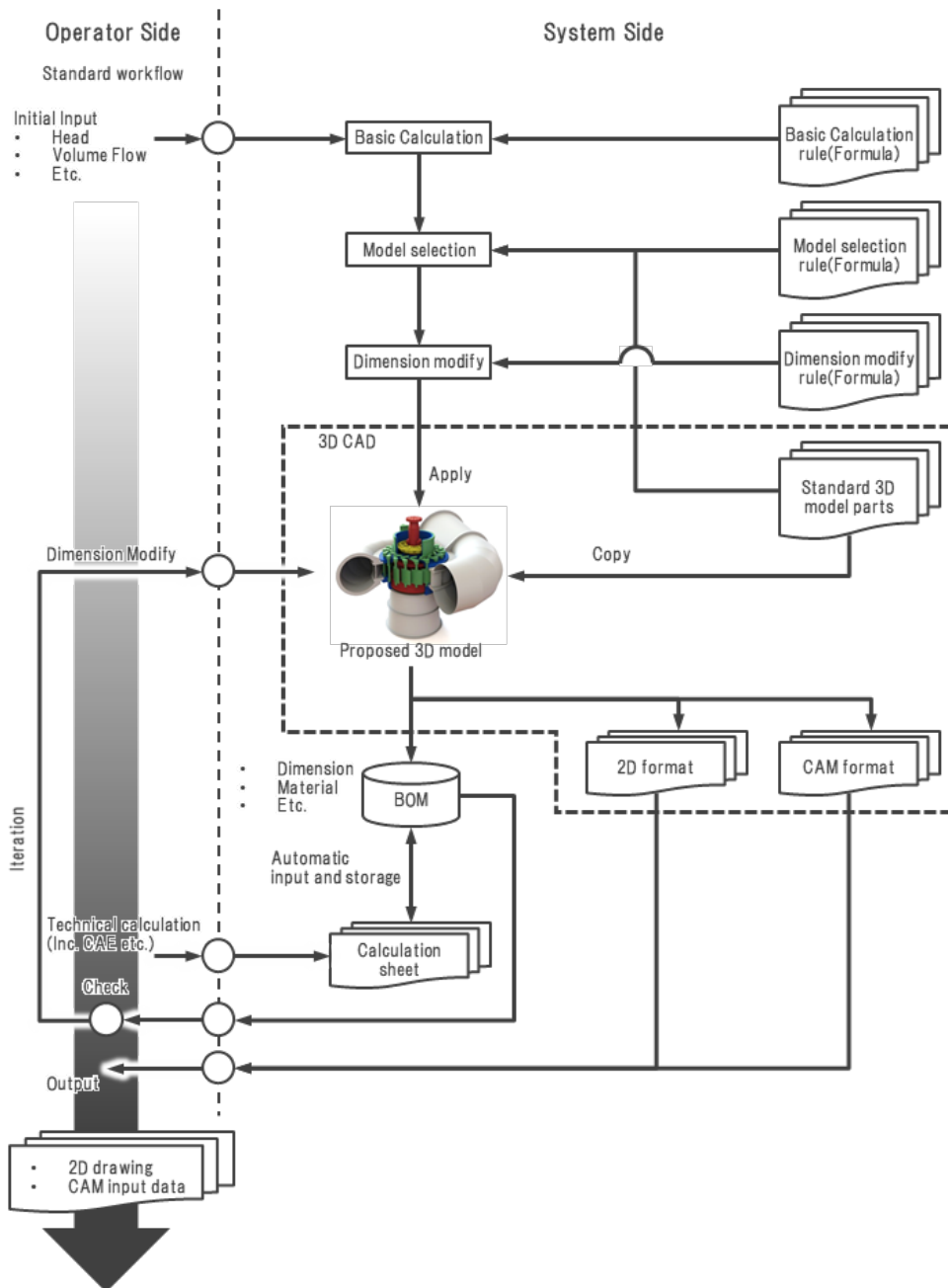


Fig. 4.4 Workflow Diagram on 1DCAE Virtual Engineering System

本環境は、1.1.2項で説明したとおり3D-CADを中心とし、CAEやCAMとの連携を可能とする機能の他、以下の機能を有する。なお、(A)～(D)は完全自動で実施され、結果は3Dモデル(Proposed 3D Model)とBOMに格納される。

- (A) 比速度など、発電用水車基本値の計算 (Basic Calculation)
- (B) (A)に対する構成部品のバラエティ選定 (Model Selection)
- (C) (B)に対する実寸法の計算・決定 (Dimension Modify)
- (D) 情報のBOMへの格納
- (E) 技術計算の自動化
- (F) 任意変更 (Dimension Modify, Iteration)

以下、上記(A)～(F)を個別に説明する。

#### 4.2.2 Basic Calculation

Basic Calculationは、初期入力を元に比速度など発電機用水車の基本となる値を計算する部分である。なお、初期入力には以下を基本とし、オプションで主要部品の材質を指定することもできる。

- ・ 落差 H (Head[m])
- ・ 出力 P (Power[kW])
- ・ 流量 V (Volume Flow[m<sup>3</sup>/hr])

計算は3.1節で説明したDimension Calculation Sheetで行われる。以下にBasic Calculationにて計算する値の一部を示す。なお、Dimension Calculation Sheetに関する詳細な説明は付.2.3項に示す。

- ・ 比速度
- ・ 回転数
- ・ 羽根車周速係数

- ・ 羽根車外径
- ・ モデル比（基本モデルに対する寸法の比率）

#### 4.2.3 Model Selection

Model Selection は Basic Calculation の結果をもとに，発電機用水車部品のパラエティを決定し，必要な 3D-CAD データを決定する。Basic Calculation と同じく，Dimension Calculation Sheet で行われる。

#### 4.2.4 Dimension Modify

Dimension Modify は 3D-CAD データに設計寸法を入力する。入力は以下の 2 つの処理に分けて行われる。

処理 1: モデル比の適用

- (1) Model Selection で選択した 3D-CAD データ内の寸法値をすべて取得。なお，ここでの寸法値とは長さ寸法のみであり，角度寸法および個数値は含まない。
- (2) 取得した寸法値を Basic Calculation で計算したモデル比倍する
- (3) モデル比倍した寸法値を 3D-CAD データ内のもとの値に上書きする。

処理 2: 調整

- (4) ネジサイズ，板厚などを JIS 規格サイズへ変更
- (5) 部品同士の隙間などを製品設計規格寸法へ変更

なお，処理 2 においては Basic Calculation と同じく，Dimension Calculation Sheet で行われる。

#### 4.2.5 情報の BOM への格納

3D-CAD データ内の値を角度寸法，個数値を含め所得し BOM へ格納する。1.1.2 項で述べた通り，本設計システムは 3D-CAD を中心とした物理情報の一元管理

を特徴としており，BOM と 3D-CAD の値が一致していることは非常に重要である。

#### 4.2.6 技術計算

4.2.1～4.2.5 項にて作成したデータにて，発電機用水車が成立するかを確認するための技術計算を行う部分である。Technical Calculation Sheet で行われる。Technical Calculation Sheet の詳細な説明は付.02.1 項に記載している。4.3 節で例を示すが，計算は一括計算，ユニット単位計算および個別計算が選択できる。また，計算結果は合否判定の他，Virtual Engineering 設計環境のユーザインタフェースから直接計算書呼び出すことで値を確認することも可能である。

#### 4.2.7 Iteration

Iteration は本 Virtual Engineering 設計環境の大きな特徴である。第 1 章にて記載した通り，本稿では，4.2.1 節記載 B)，C) 項で機械寸法を決定する際に寸法の適正化／最適化は行っていない。したがって，初回の C) 項終了時における発電用水車の形状は，Initial Input を最低限満たすものでしかない。発電用水車の形状を適正に近づける作業，例えば羽根車直径の微調整や部品の肉厚変更等はオペレータが判断し Fig. 4.4 記載の Iteration を実施することで行う。また，3.2 節で説明した通り，Iteration 作業のインタフェースは 3D-CAD であり，すべて 3D-CAD を通して行う。なお，4.2.1 節記載 B)，C) 項に適正化／最適化ロジックを組み込むことができれば Iteration 作業を減らすことが可能であるとともに，結果の属人性を低くすることが可能である。しかし現実の製品設計，特に大型の製品システムにおいては，同一の Initial Input であっても，案件によって加工機械，輸送条件，現地施工条件などが異なるため，これらに合わせた設計変更が必要となる。これらの条件をあらかじめ 4.2.1 節記載 B)，C) 項に記載しておくことも技術的には可能であるが，その条件は多岐にわたる。また，条件の使用頻度が数十年に一度というものもあり，これらをすべて事前に数式として用意しておくことは難しく，適正化／最適化ロジックの組み込みは今後の課題である。

Initial Input 作業にて調整が必要な例を以下に示す。

- (1) 4.2.6 項記載の技術計算を行った結果、部品の詳細寸法を変更する必要が生じる場合
- (2) 機械製品システムの輸送に制限があり、特に大型部品の分割を考慮しなければならない場合
- (3) 海外向け製品であるなどの理由により、ボルトやフランジ等の適用規格が異なる場合
- (4) 特殊な材料を要求されている場合

なお、Iteration の結果は、Iteration 後の値を用いて 4.2.6 項の技術計算が自動で行われ変更の妥当性が評価される。これは 1DCAE の本質である、“個別最適化設計の結果を全体適正設計に戻す”ことに相当する。

ここで、上記(1)に対して補足する。技術計算を行った結果、例えば構造物の強度が不足している場合では、設計者は必ず強度を満足するように構造物の寸法見直しを行う。逆に強度が十分すぎるほど満足していた場合、現状これを適正な強度に見直すことはなかなか行われていない。これは、実設計業務においては限られた時間の中ですべてを満足する解を得る必要があるため、技術的に必ずしも必要ではない作業の優先度が低いことによる。しかしながら、不必要な強度を適正に見直すことで、多くの場合、部品の軽量化が図られ、材料調達・製造・輸送などで環境負荷・コスト低減が期待できる。本 Virtual Engineering 設計環境の Iteration 機能は BOM と計算書が連動していることから設計者は 3D モデルの寸法変更に注力でき、その結果を短時間かつ自動で得ることができるため、Iteration を積極的に行うことを推奨している。

### 4.3 データ管理

本 Virtual Engineering 設計環境は、特に BOM においてマルチユーザを想定して設計・構築されている。これは 1DCAE においては複数のユーザがそれぞれの担当部位の詳細検討を同じタイミングで行うことが推定されるからである。したがっ

て、BOMは複数のユーザが同時に同一のデータを参照・更新した場合でも、矛盾なく正常に処理が行われること、つまりトランザクション機能を有している必要がある。

本 Virtual Engineering 設計環境が使用している BOM は Microsoft SQL Server を用いて構築している。Microsoft SQL Server は関係データベース管理システム (RDBMS: Relational Database Management System) でありトランザクション機能を有している。

また、BOMはユーザがアクセス可能なサーバ上に一つ構築されており、すべてのユーザがこの一つの BOM にアクセスすることになる。対して Virtual Engineering 設計環境を提供するソフトウェアはそれぞれのユーザが使用するクライアントにそれぞれインストールされる。

その他のデータは以下の通りである。

- ・ 計算書および 3D モデル
  - マスターデータ  
管理者権限を持つユーザのみがアクセス可能なサーバ上に一つだけ存在する
  - 実設計データ  
すべてのユーザがアクセス可能なサーバ上に実設計単位で一つだけ存在する
- ・ JSM データ
  - JSM editor データ (Excel ブック形式)  
任意の場所に保管・管理が可能である。
  - 設計システム用業務フロー自動生成出力  
SQL Server が管理する Virtual Engineering 設計環境関連ファイル領域に一つだけ存在する。
  - 設計システム用 BOM 構造の自動生成出力  
SQL Server が管理する Virtual Engineering 設計環境関連ファイル領域に一つだけ存在する。



- 設計システム用 JSM マトリクス部データ自動生成出力  
SQL Server が管理する Virtual Engineering 設計環境関連ファイル領域に一つだけ存在する。

#### 4.4 データ出力

本 Virtual Engineering 設計環境から出力可能なデータは以下の通りである。

- ・ 実設計情報リスト
  - ファイルフォーマットは CSV 形式である。
- ・ BOM データ
- ・ 3D モデルファイル
- ・ 計算書
  - Microsoft Excel ブックのみ直接出力可能
  - その他外部ソフトウェアを使用している場合は、外部ソフトウェアの機能を用いての出力を必要とする。

なお、現時点では、2D 図面は出力された 3D モデルデータを 3D-CAD にて読み込み、3D-CAD の機能を用いて作成する手法としている。このため、本 Virtual Engineering 設計環境は 2D 図面を出力する機能は有していない。これは CAM データに関しても同様である。

#### 4.5 まとめ

第 4 章では、製品システムの例として発電用水車をターゲットとし、「発電用水車」の説明、「実際の製品システム設計に適用できる Virtual Engineering 設計環境の構成」の詳細を示した。特に 4.2 節においては本環境上でデータを構築していく手順を詳細に述べた。

## 第 5 章

### 検証・評価

## 第 5 章 検証・評価

本章では、第 5 章に記載した実際の Virtual Engineering プラットフォームにて発電機用水車の設計を行い、従来の方式と比較することで本開発の有効性を検証・評価した結果を説明する。

### 5.1 評価内容および条件

以下の 2 点に対して評価を行った。

- (1) JSM および JSM editor の効果
- (2) Virtual Engineering 設計環境の有効性

以下に評価条件を示す。なお、3.3.3 項で述べた通り、実際の設計で必要とする入出力値数、すなわち JSM における Spec 数は数千～数十万に及ぶが、本評価では Spec 数の少ない簡易モデルを使用した。

- (A) 対象機器
  - ・ 発電用フランシス水車
  - ・ 3D モデルパーツ数:
    - JSM および JSM editor の効果確認用: 49 個
    - Virtual Engineering 設計環境の有効性確認用: 51 個
  - ・ 3D アッセンブリ数: 11 個
- (B) JSM
  - ・ Job 数: 43 個
  - ・ Spec 数 (主に 3D モデルの寸法):
    - JSM および JSM editor の効果確認用:  
869 個
    - Virtual Engineering 設計環境の有効性確認用:  
1149 個

(C) 実行環境

- ・ OS: Windows 10 Pro (64bit)
- ・ 中央演算処理装置 (CPU) :  
Intel (R) Core (TM) i5-6600 CPU@3.30 GHz
- ・ 主記憶装置 (RAM) : 8 GB
- ・ グラフィックボード: NVIDIA GeForce GT 7300

(D) 外部ソフトウェア

- ・ Solidworks 2014
- ・ Microsoft Excel 2010

## 5.2 JSM および JSM editor の効果

評価は以下の (1) ~ (4) ケースを実施し、それぞれにおける JSM の作成やエラーの修正などに掛かった時間を比較することで行った。

- (1) システムを使用しないマニュアル設計
- (2) システムのみ使用
- (3) システム使用および JSM を紙面で運用
- (4) システム使用および JSM editor を使用

なお、(1) ~ (4) の対象機器はすべて同一仕様であるため、(1) ~ (4) の作業を同一人物が行うと作業慣れが発生し正確な評価を行えない可能性がある。また、(1) ~ (4) の作業を個別の人物が行うと、各人の設計スキルの差により正確な評価を行えない可能性がある。そこで本評価では、一名の熟練設計者を指導者とし、設計に必要な Spec 選定や計算書の準備、実行手順作成等を行ってもらった。また、(1) ~ (4) の作業は個別の非熟練者が担当した。したがって非熟練者の作業は JSM の作成やシステムへのデータ登録、計算書にしたがった技術計算等の単純作業のみとなる。また、熟練者から非熟練者に作業を指示する場合において、(1) ~ (4) の評価を順番に実施すると、熟練者の慣れにより指示精度が順次向上し、次の評価結果に影響を与える可能性がある。そこで、熟練者の作業は、

例えば水車の基本仕様計算部分や部品 A, B の詳細計算部分など個々のステップが終わるごとに、非熟練者全員に一斉に展開し (1) ~ (4) の作業を並行して進めることとした。

評価の結果を Table 5.1 に示す。Table 5.1 は、最初の入出力値群定義 (Initial JSM create または Initial translate data create) からこれらの誤記をすべて修正 (Mismatch repair work) し、正しい 3D モデルを得るまでの作業時間を外部ソフトウェアに対する入出力値群定義を行う総作業時間 (Total working time) として評価している。なお、1 行目に前記 (1) のマニュアルで設計を行った場合の作業時間を記載している。なお、1 行目の総作業時間は 72.8 時間であり、前記 (2) に相当する 2 行目の総作業時間の 51.9 % であるが、先に述べた通りマニュアル作業であり、異なる仕様の機器を検討する際に毎々同等の作業時間が発生する。対して、前記 (2) ~ (4) に相当する 2-4 行目は Virtual Engineering 設計環境を用いている。総作業時間が発生するのは設計システム構築時の 1 回だけであり、それ以降は設計システム実行時間 (Basic design calculation ~ 3D model dimension modify) のみしか発生しない。このため、1 行目と 2-4 行目を直接比較はできない。

2 行目に示す通り、従来手法、すなわち JSM を用いず、それぞれの Job への Spec を個別に記載・管理する手法の総作業時間は 140.36 時間であった。この値を基準として、3 行目記載の JSM を使用した場合、および 4 行目記載の JSM editor を使用した場合の総作業時間は 69.98 時間および 26.12 時間であり、それぞれ 50 %、81 % の総作業時間低減が確認できた。

Table 5.1 より総作業時間低減は以下によってなされていると解釈できる。また、本評価には含んでいないが、これらの効果は、3D モデルの変更等により入出力値群の定義を変更する必要が生じた際にも効果があると考えられる。

- (1) JSM を使用することによる誤記発見の容易化
- (2) JSM editor を使用することによる誤記発見の自動化
- (3) (1), (2) に起因するリトライ回数の低下
- (4) JSM および JSM editor を使用することによる設計システムデータアップデート作業の簡易化

Table 5. 1 Result of evaluation

		Manual work (Reference)			System work									Total working time[hr]	Working time ratio[%]
		Basic design calculation[hr]	Detail dimension calculation[hr]	3D model dimension modify[hr]	Initial JSM create[hr]	Initial translate data create[hr]	Total of update and run time[hr]					Mismatch repair work(Total)[hr]			
							System data update[hr/num]	Basic design calculation[hr/num]	Detail dimension calculation[hr/num]	3D model dimension modify[hr/num]	Number of run[num]				
1	Without JSM (Manual work)	3.0	46.5	23.3	-	-	-	-	-	-	-	-	72.8	51.9	
2	Without JSM	-	-	-	-	32.20	82.16					26	26.00	140.36	100.0
							3.16								
							3.00	0.01	0.04	0.11					
3	With JSM, Without JSM editor	-	-	-	30.60	-	20.88					18	18.50	69.98	49.9
							1.16								
							1.00	0.01	0.04	0.11					
4	With JSM editor	-	-	-	19.50	-	4.62					7	2.00	26.12	18.6
							0.66								
							0.50	0.01	0.04	0.11					

### 5. 3 Virtual Engineering 設計環境の有効性

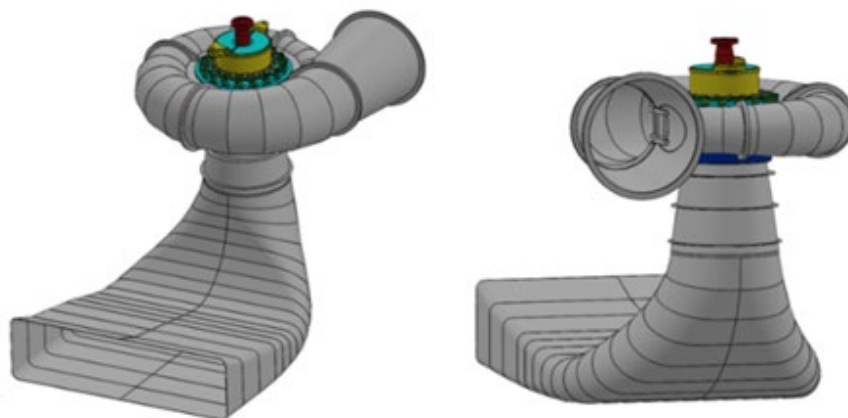
#### 5. 3. 1 処理時間

本 Virtual Engineering 設計環境の実処理時間を以下に示す。本環境は処理速度を追求したものではない。しかしながら、現在、本作業を熟練設計者が 1～数日かけて行っていることと比較すると、十分な処理速度が得られたと判断する。

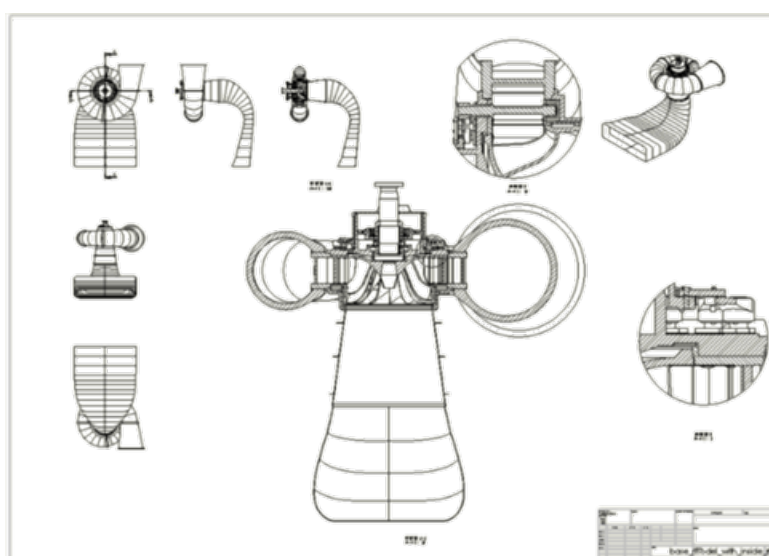
- ・ 起動およびデータのコピー処理： 47 秒

- ・ 基本仕様計算および部品の選定： 36 秒
- ・ 実寸法の計算・決定： 284 秒
- ・ 3D モデルへの寸法反映： 174 秒
- ・ その他の処理： 9 秒
- ・ 合計： 550 秒

Fig. 5. 1 に得られた発電機用水車の 3D モデル全体を示す。また， Fig. 5. 2 に 2D 図面を示す。



**Fig. 5. 1 Example of 3D Model by Virtual Engineering System**



**Fig. 5. 2 Example of 2D Drawing by Virtual Engineering System**

### 5.3.2 羽根車バリエーション

4.1.2 節にて説明した 12 種類の羽根車バラエティの結果を以下に示す。バラエティは主に以下の幾何学的形状の構成要素で分類されている。また、どのバラエティが選択されるかは、4.2.1 節 (B) 項において、Fig. 5.3 を用いて、設計仕様である落差 (Head) および式 (4.2) で得た比速度から一位的に決定できる。

- Blade number (羽根枚数)
- Inlet profile (L or L + S)
- Outlet profile (L or L + S1 or L + S1 + S2)
- Crown offset (With or Without)
- Crown angle (With or Without)

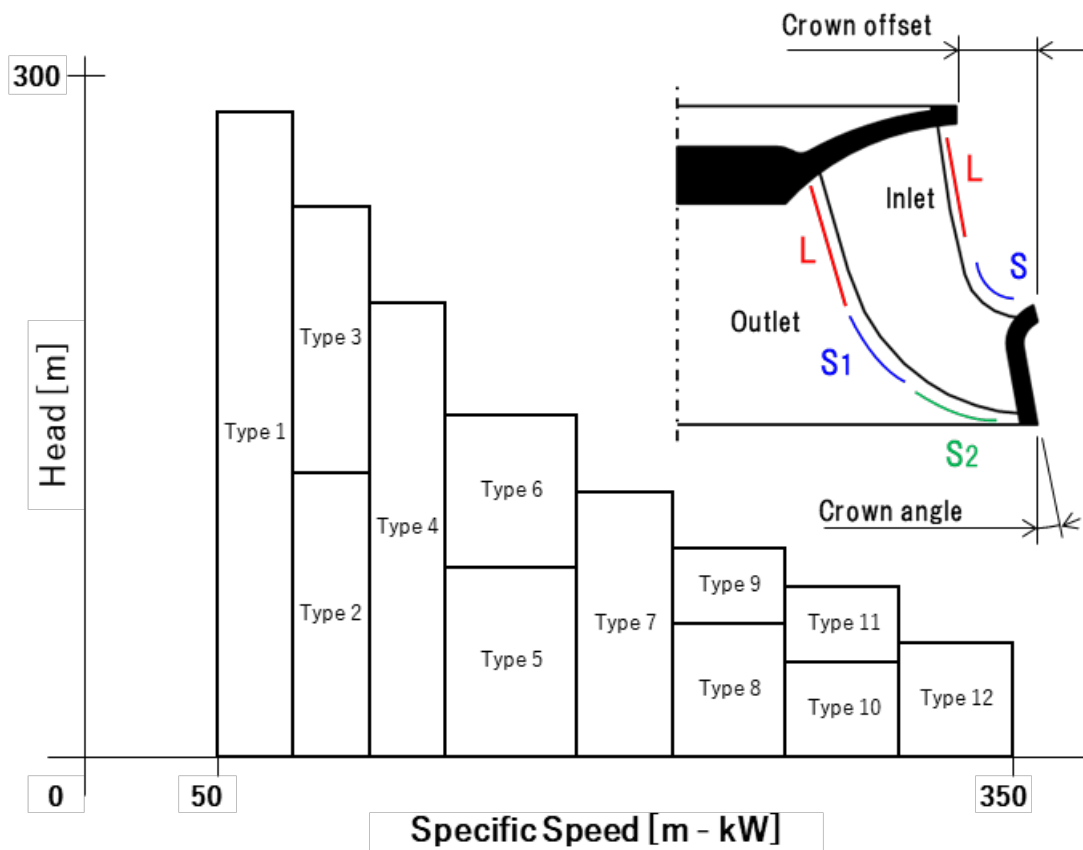


Fig. 5.3 Selection Map of Runner type (Reference Example)



また，上記で選定された羽根車は 4.2.1 節 (C) 項によって寸法を割り当てられ，3D Model 化されることが確認できた。さらに 1 DCAE 環境活用の実例として，4.2.1 節 (F) 項にて，上記で得られた羽根車 3D Model の流体性能や機械的強度評価を 3D Model を CAE ソフトウェアに渡して行えること，および CAE ソフトウェア側で行った羽根車 3D Model 寸法微調整の値をオペレータが元の 3D Model に反映できることを確認できた

### 5.3.3 Iteration

Iteration の例として Dimension Modify を実施した例を示す。Fig. 5.4 は案内羽根 (Guide Vane) 機構部の Technical Calculation が行われた後の画面の一部である。

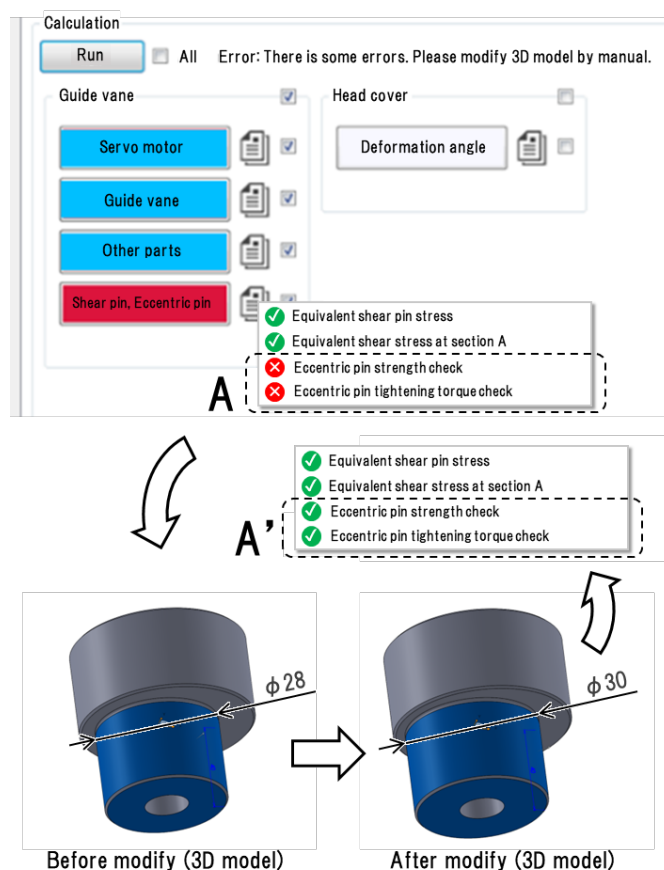
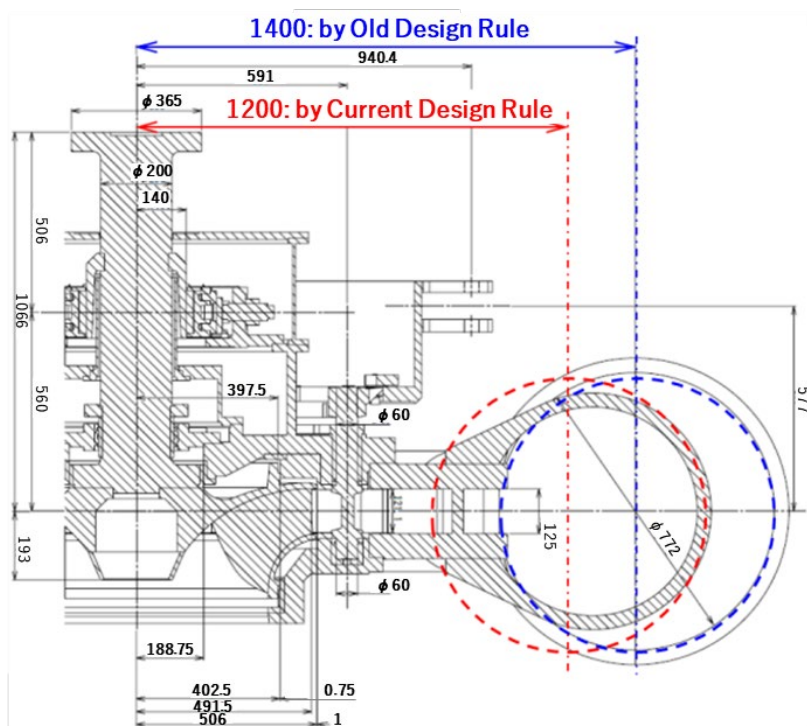


Fig. 5.4 Example of Common Window that Project Status

Fig. 5. 4 中の「A」部「赤丸 X 印」が示すように、案内羽根機構部部品の一つである偏心ピン (Eccentric Pin) の強度計算にて満足な解が得られていないことが確認できる。ここで、3D-CAD 上で偏心ピンの根本直径を変更し、再度 Technical Calculation を実施したところ、「A'」部「緑色 ✓ 印」で示されるように、偏心ピンの強度に問題がないことを確認できた。本環境では、このように、3D-CAD をインタフェースとして用い、「個別最適化設計の結果を全体適正設計に戻す」ことが可能である。

### 5. 3. 4 機能開発技術の変更

本環境構築時に過去に製作実績のある発電用水車の Initial Input を用いて試行を行ったところ、Fig. 5. 5 に示すようにケーシング中心半径が製作実績と大きく異なる結果 (1,400[mm]) となった。



**Fig. 5. 5 Example of Difference Design Point between Virtual Engineering System and Expert Engineer**

これは、ケーシングの中心半径を求める計算式に誤って「ベルマウス形スピードリング」と呼ばれる古い設計形状に用いる数式を適用していたことが原因であった。ここで、Dimension Calculation Sheet 内の数式を現在の設計形状である「平行形スピードリング」に用いる式に変更し、再度実行することで製作実績と同じ 1,200 mm となった。なお第 3 章で説明した通り、本環境は機能開発技術のデータが独立しているため、この際に Engineering System には修正を加える必要はなく、また数式変更作業時間は 10 min.程度であった。

### 5.3.5 環境運用技術・機能開発技術の更新・追加

以下、本環境を 2 年間使用した間に行った環境運用技術・機能開発技術の更新・追加について記載する。

#### A) Initial Input

- 押し込み圧力 (Back pressure) (追加)
- ケーシング分割数 (追加)

#### B) 環境運用技術

- 市販流体解析 CAE への入出力 (追加)
- 市販危険速度計算 CAE への入出力 (追加)
- 市販 3D-CAD ソフトウェアバージョン (更新)

#### C) 機能開発技術

- 水潤滑軸受寸法計算式 (追加: 142 個)
- 水潤滑軸受性能評価計算式 (追加: 23 個)
- ケーシング分割数別寸法計算式 (追加: 40 個)

#### D) データ

- 水潤滑軸受 3D model (追加: 18 個)
- JSM (上記すべてに伴う更新)

これらの環境への実装作業は JSM の更新が主である。また、実装は IT 技術者ではなく発電用水車の設計者が行い、JSM の更新時間は合計で 69.5 時間であっ

た。さらに第 3 章で説明した通り，実装作業において Engineering System に一切の変更を加える必要はなかった。これらのことから，本環境は環境運用技術・機能開発技術の速やかな更新が可能であることが確認できた。

## 第 6 章

### 結言

## 第 6 章 結言

### 6.1 結論

製品システムの設計において、手戻りなどの問題点を含む従来の設計手法から脱却するための、異なる製品システムの設計にも適用できる 1DCAE およびフロントローディング設計環境プラットフォームを提案することを目的として、以下に示す研究を行った。

- ◆ 活用技術
- ◆ 設計プラットフォームのトレンド追従方法

活用技術においては、1DCAE Virtual Engineering 環境のプラットフォームを作成し、実際に発電機用水車向けの環境を構築し設計に適用することで、その効果を評価した。その結果、要求仕様から短時間で発電機用水車の 3D モデルを得ること、3D-CAD のインタフェースとしての有効性、および IT 技術者を必要とすることなく環境運用技術・機能開発技術の入れ替えが可能であることを確認した。設計プラットフォームのトレンド追従方法においては、JSM をベースとした外部ソフトウェア間の入出力値群を定義するマトリックススタイルの表記方法を考案し、JSM 作成および設計システムへの実装をサポートするエディタを開発しその効果を検証した。この結果、4 章に記載した発電機用水車の評価のように多種の内部数値計算が必要であり、かつ 3D CAD などと設計システム連携する場合において 81 %以上の作業時間削減に貢献することを確認した。

本研究で構築した 1DCAE Virtual Engineering 環境の特徴を以下に記載する。

- A) 優先順位なく詳細設計を進めることが可能であり、また、製品システム全体を見ながら評価・修正を行えるため、設計者が直感的に機械構成のバランスを取りやすくなる。これは 1DCAE の本質である、「個別最適化設計の結果を全体適正設計に戻す」ことに相当する。

- B) IT 技術者でなくとも環境運用技術・機能開発技術の更新が可能であり，構築した 1DCAE Virtual Engineering 環境を長期間にわたって運用することが可能である。
- C) インタフェースが統一され，環境運用技術・機能開発技術変更の際に JSM の定義を除き 1DCAE Virtual Engineering 環境に変更を加える必要が無い。また，環境運用技術・機能開発技術を入れ替えることで，異なる製品システムの設計にも使用することが可能となる。

## 6.2 今後の課題

### 6.2.1 1DCAE Virtual Engineering 環境

本研究で開発した 1DCAE Virtual Engineering 環境を多くの製品システムに適用するにあたり，今後，以下の改良を加えていく考えである。

- (1) 本研究で開発した JSM editor のフロー機能はシーケンシャルであり，分岐，繰返しといった機能を有していない。これらは現実の設計・製造業における実行フローでは必須の機能であり，今後，これらの機能実装が必要である。
- (2) 強度解析に使用するメッシュ情報等は Spec 数が多く，JSM の作成やメンテナンスに労力がかかると思われる。これを解決する手法として，例えば複数の Spec を構造化し Job への入出力を C 言語のポインタのように先頭 BOM コードで行う仕組みも考えている。
- (3) 設計形状の決定において，熟練設計者のノウハウが正確に適用できていない部位がある。4.2.2 項で述べたとおり，本研究では過去に製作した発電用水車の実績を基に線形回帰にてルールを作成している。しかしながら，実寸法の中にはステップ状に変化する値などもあり，線形回帰だけでは充分ではなく，熟練設計者が経験的・感覚的に使用しているルールをより正確に記載することは重要である。ルール作成に対する解決策の一つは統計的な手法を用いることであり，応答局面法<sup>(35)</sup>や実験計画法<sup>(36), (37)</sup>を使用した手法が報告されて

いる。また、近年では人工知能（AI: Artificial Intelligence）を使用してルールを自動作成する手法<sup>(38)</sup>も提案されている。しかし、本研究が対象とする発電用水車を代表とする大型機械製品システムは、その設計・製作数が少なく、現時点で必要なルールを自動作成するに足る数の教師データが準備できない。しかしながら、本研究で報告する 1DCAE Virtual Engineering 設計環境により短時間で機械製品システムとして成立する仮想設計データを作成・蓄積することを継続し、多くの教師データを得ることができれば、将来的に人工知能を用いたルール自動作成につなげていけるものと考えられる。また、教師データが少ない場合に対するタグチ T 法<sup>(39)</sup>の適用も検討している。

#### 6.2.2 その他機械製品システムへの展開

発電機用水車はその性能に高い効率を求められるため、機器ごとに個別最適化設計を行う必要があり、多くの外部ソフトウェアを必要とする。JSM は、このような機器の設計環境において有効に働くことが確認できた。また、同様な設計手法の機器として、例えば、以下に示す機器の設計環境においても本手法が有効であり本設計システムの展開を考えている。

- ・ プロセス圧縮機
- ・ 産業用ポンプ
- ・ 産業用大型モータ
- ・ 高圧インバータ



## 参考文献

---

- (1) 大富浩一，山崎美稀，加藤千幸，福江高志：「1DCAEでものづくりを変える」，日本機械学会誌，第120巻，第1188号，pp. 4-7（2017）
- (2) 大富浩一：「1DCAEの背景，考え方，課題，今後」，日本機械学会誌，第120巻，第1188号，pp. 10-15（2017）
- (3) 山崎美稀：「材料選定と1DCAE」，日本機械学会誌，第120巻，第1188号，pp. 24-27（2017）
- (4) 大富浩一：「1DCAEと設計教育」，日本機械学会誌，第120巻，第1188号，pp. 34-35（2017）
- (5) K. Ohtomi, T. Hato: "Design Innovation Applying 1DCAE", Toshiba Review, Vol. 67, No. 7, pp. 7-10（2012）（in Japanese）  
大富浩一，羽藤武宏：「1DCAによるものづくりの革新」，東芝レビュー，Vol. 67, No. 7, pp. 7-10（2012）
- (6) Ohtomi, K., "Importance of Upstream Design in Product Development and Its Methodology", Proceedings of EuroSimE 2005, Berlin, 2005-04, IEEE. 2005, pp.17-18（2005）
- (7) Y. Kusunoki, M. Mimura, S. Sasaki: "Automation of Engineering Design Process by Development of Parametric Design System", Mazda Technical Review, No. 22, pp. 120-125（2004）（in Japanese）  
楠康友，三村光生，佐々木晋：「パラメトリック設計システムの開発～エンジン設計の自動化～」，マツダ技報，No. 22, pp. 120-125（2004）
- (8) N. Takai, S. Yoshizawa, K. Suzuki, Y. Sugawara: "High-speed Automatic Design of Comparator Circuit Using Divided Circuit Optimization", IEEJ Transaction on Electronics, Information and System, Vol. 138, No. 1, pp. 57-64（2018）（in Japanese）  
高井伸和，吉澤慧，鈴木研人，菅原誉士紀：「部分最適化を用いたコンパレータ回路の自動設計の高速化」，電気学会論文誌C，第138巻，No. 1, pp. 57-64（2018）

- (9) A. Ura, M. Sanada, S. Morimoto, Y. Inoue: "Study on low torque ripple IPMSM structure for compressor of air conditioner using automatic design system", The papers of Technical Meeting on Home and Consumer Appliances , IEE Japan, 2018, 26-35, pp. 53-58 (2018) (in Japanese)  
浦聡洋, 真田雅之, 森本茂雄, 井上征則: 「自動設計システムを用いたエアコン圧縮機用低トルクリプル IPMSM 構造の検討」, 電気学会研究会資料. HCA / 家電・民生研究会[編], 2018, 26-35, pp. 53-58 (2018)
- (10) Masahiro Kashiwai, Kazuharu Ishizuka, Akira Yoshida, Hailong Teng, Koichi Hayama, Shin Konomi: "Automated Design Environment for Custom Turbo Machinery -Automated Design of Centrifugal Compressors Using Parametric Design-", Ebara Engineering Review, No. 257, pp. 3-7 (2019)  
柏井正裕, 石塚一晴, 吉田晃, 藤海龍, 葉山耕一, 許斐真: 「ターボ機械のカスタム製品における自動設計化ーパラメトリック設計を用いた遠心コンプレッサの自動設計ー」, エバラ時報, No. 257, pp. 3-7 (2019)
- (11) Jeff A. Estefan: "Survey of Model-Based Systems Engineering (MBSE) Methodologies", INCOSE MBSE Initiative, May 23 (2008)
- (12) Albert Albers, Quentin Lohmeyer: " Advanced Systems Engineering – Towards A Model-Based and Human-Centered Methodology ", Proceedings of TMCE 2012, May 7–11, 2012, PP. 407-416 (2012)
- (13) Meussen, Bernhard: " On The Use of Model Based Systems Engineering And CAD for The Design of Physical Products ", INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED21, 16-20 AUGUST 2021, pp. 2317-2326 (2021)
- (14) K. Orsvarn, T. Axling: "The Tacton View of Configuration Tasks and Engines", AAAI Technical Report, WS-99-05, (1999)
- (15) Miyakawa, S., "Extended Assemblability Evaluation Method", JSME International Journal, Series C, Vol. 45, NO. 2, pp. 567-574 (2002)

- (16) 泉聡志, 伊藤照明, 内田考尚, 木間塚明彦, 木見田康治, 栗崎彰, 土井大介, 長谷川洋, 山本碩徳: 「VE (Virtual Engineering) 時代の設計 / ものづくり」, 日本機械学会誌, 第 120 巻, 第 1184 号, pp. 28-41 (2017)
- (17) T. Saito, K. Mitsukuni: "Comparative Experiment of Information Relationship Description Methods in Business Process Decomposition", IEEJ Transaction on Electronics, Information and System, Vol. 139, No. 8, pp. 944-951 (2019) (in Japanese)  
齋藤哲, 光圀光七郎: 「業務プロセスの分解における情報連携表記方法の比較実験」, 電気学会論文誌 C, 第 139 巻, No. 8, pp. 944-951 (2019)
- (18) T. Makino, K. Kuriyama, A. Hada, S. Fukuchi, T. Tsumaki: "An approach for modeling of application systems based on Service-Oriented Architecture", IPSJ SIG, Technical Report, SE-150 (3) , pp. 17-23 (2005) (in Japanese)  
牧野友紀, 栗山勝宏, 羽田昭裕, 福地修一, 妻木俊彦: 「SOA に基づくアプリケーションシステムの課題と提案」, 情報処理学会, 研究報告, SE-150 (3) , pp. 17-23 (2005)
- (19) K. Oba, M. Hashimoto, S. Fujikura, T. Munehira: "The Status Quo and Challenges of Service-Oriented Architecture (SOA) Based Application Design", IPSJ, Technical Report, SE-149 (10) , pp. 73-80 (2005) (in Japanese)  
大場克哉, 橋本誠, 藤倉成太, 宗平順己: 「サービス指向アーキテクチャ (SOA) に基づくアプリケーション設計の現状と課題」, 情報処理学会, 研究報告, SE-149 (10) , pp. 73-80 (2005)
- (20) A. Esaka, M. Noro, A. Sawada: "Construction of Product Line for SOA Application Platform", IPSJ, Technical Report, SE-183, No. 13, pp. 1-8 (2014) (in Japanese)  
江坂篤侍, 野呂昌満, 沢田篤史: 「SOA アプリケーションプラットフォームのプロダクトライン化」, 情報処理学会, 研究報告, SE-183, No. 13, pp. 1-8 (2014)

- (21) H. Igaki, N. Nakamura, H. Tamada, K. Matsumoto: "Implementing Integrated Services of Networked Home Appliances Using Service-oriented Architecture", IPSJ, Vol 46, No. 2, pp. 314-326 (2005) (in Japanese)  
井垣宏, 中村匡秀, 玉田春昭, 松本健一: 「サービス指向アーキテクチャを用いたネットワーク家電連携サービスの開発」, 情報処理学会誌, Vol 46, No. 2, pp. 314-326 (2005)
- (22) スティーブン・D.エッピングジャー: 「デザイン・ストラクチャー・マトリックス DSM: 複雑なシステムの可視化とマネジメント」, 慶応義塾大学出版会, (2017)
- (23) A. Nishimoto: "Study on product design process modeling with integrated Design Structure Matrix", JSME, The 18th Conference on Design and Systems Division, pp. 58-61 (2008) (in Japanese)  
西本明弘: 「i-DSMによる製品設計プロセスモデリングに関する考察」, 日本機械学会, 第18回設計工学・システム部門講演会, pp. 58-61 (2008)
- (24) 目代武史: 「設計開発マネジメントの分析ツールとしての設計構造マトリクスに関する考察」, 地域経済研究, 第17号, pp. 25-24 (2006)
- (25) 齋藤稔, 岡田伊策, 笈田佳彰, 渡辺郁雄, 松本滋, 稗方和夫: 「設計品質向上のための既存設計ドキュメント活用方法の提案」, 人工知能学会, 第2種研究会資料, pp. 1-5 (2013)
- (26) K. Hiekata, H. Yamato, N. Fukuda, S. Nakamura, I. Okada, M. Saito, Y. Oida, I. Watanabe, S. Matsumoto: "Development of Impact Analysis System for Specification Change of System Using Design Information", JSME, The 24th Conference on Design and Systems Division, No. 14-27, (2014) (in Japanese)  
稗方和夫, 大和裕幸, 深田直人, 中村覚, 岡田伊策, 齋藤稔, 笈田佳彰, 渡辺郁雄, 松本滋: 「システムの仕様変更調査における設計情報を用いた影響分析システムの開発」, 日本機械学会, 第24回設計工学・システム部門講演会論文集, No. 14-27, (2014)

- (27) T. Nagase: "Business Model Analyzing Process BMAPROS", UNISYS TECHNOLOGY REVIEW, Vol. 105, pp. 97-108 (2010) (in Japanese)  
 長瀬隆: 「業務分析手法 BMAPROS」, UNISYS TECHNOLOGY REVIEW, 第 105 号, pp. 97-108 (2010)
- (28) K. Sadamitsu: "Optimization for the Migration Units in Information Systems based on CRUD Relations with Semi-Supervised Graph Neural Networks", The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, pp. 1-4 (2018) (in Japanese)  
 貞光九月: 「半教師有りグラフニューラルネットワークを用いた CRUD 関係に基づくシステム移行単位の最適化」, The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, pp. 1-4 (2018)
- (29) Browning, Tyson R.: "The Many Views of a Process: Toward a Process Architecture Framework for Product Development Processes", Systems Engineering, 12, (1), pp. 69-90 (2009)
- (30) Suh, E. S., O. L. de Weck, and D. Chang, "Flexible Product Platforms: Framework and Case Study", Research in Engineering Design, 18, (2), pp. 67-89 (2007)
- (31) 日野三十四: 「実践 エンジニアリング・チェーン・マネジメント-IoT で設計開発革新」, 日刊工業新聞社, (2017)
- (32) S. Hino: "E-BOM Norm and BOM Data Cooperating System from the Viewpoint of Product Design -Progress Modular Design and Design Knowledge Management-", Japan Society for Production Management, Vol. 14No. 1, pp. 151-154 (2007) (in Japanese)  
 日野三十四: 「E-BOM 規範と BOM データ連携システム—モジュラー・デザインと設計ナレッジ・マネジメントの促進のために—: モジュラー・デザインと設計ナレッジ・マネジメントの促進のために」, 日本生産学会, Vol. 14No. 1, pp. 151-154 (2007)
- (33) 日野三十四: 「実践 モジュラーデザイン【改訂版】工場空洞化時代に勝ち進むために」, 日経 BP 社, (2011)

- (34) 黒川淳一：「ハイドロタービン 新改訂版」，日刊工業出版，pp. 150-160（2007）
- (35) T. Amago: "Sizing Optimization Using Response Surface Method in FOA", R&D Review of Toyota CRDL, Vol. 37, No. 1, pp. 31-36（2002）（in Japanese）  
尼子龍幸：「FOAにおける応答曲面法による寸法最適化」，R&D Review of Toyota CRDL, Vol. 37, No. 1, pp. 31-36（2002）
- (36) M. Tsuchiya, H. Ishii, K. Yamaguchi, M. Takagi: "Optimal Design Support System for Rotor Dynamics Based on the Experimental Design Methodology", Transactions of the Japan society of mechanical engineers. C, Vol. 67, No. 656, pp. 954-961（2001）（in Japanese）  
土屋雅弘，石井博，山口和夫，高木亨之：「実験計画法に基づく軸振動最適設計支援システム」，日本機械学会論文集（C編），67巻，656号，pp. 954-961（2001）
- (37) M. Tsuchiya, H. Yamakawa: "Optimal Rotor-Dynamic Design Based on Experimental Design Methodology", Transactions of the Japan society of mechanical engineers. C, Vol. 67, No. 662, pp. 3086-3092（2001）（in Japanese）  
土屋雅弘，山川宏：「実験計画法による軸振動最適設計」，日本機械学会論文集（C編），67巻，662号，pp. 3086-3092（2001）
- (38) N. Nozaki, E. Konno, M. Sato, M. Sakairi, T. Shibuya, Y. Kanazawa, S. Georgescu: "Application of Artificial Intelligence Technology in Product Design", FUJITSU, 67, 3, pp. 58-65（2016）（in Japanese）  
野崎直行，今野栄一，佐藤満，坂入慎，澁谷利行，金澤裕治：「製品設計における人工知能技術の応用」，FUJITSU, 67, 3, pp. 58-65（2016）
- (39) K. Nishino, A. Suzuki: "Taguchi's T-Method using Median-Median Line Small Sample with Outliers", IEEJ Transaction on Industry Applications, Vol. 138No. 7, pp. 598-604（2018）（in Japanese）

西野佳祐，鈴木新：「外れ値を含んだ小サンプルに対する Median-Median Line を用いたタグチ T 法」，電気学会論文誌 D, Vol. 138, No. 7, pp. 598-604 (2018)

## 謝 辞

---

本研究は、信州大学 繊維学部 小林俊一教授のご指導とご教示を賜って遂行し成しえたものである。ここに深甚なる謝意と敬意を表します。また、当方の闘病による長い休学を挟み、ご迷惑をおかけしながらも本研究を完遂できるようご配慮いただきましたこと、重ねて謝意を表します。

Virtual Engineering 設計環境の対象として発電機用水車をご提案いただき、また、研究環境をご提供いただきました日立三菱水力株式会社 大嶋勝宏専務に心より御礼申し上げます。

発電機用水車の技術指導ならびに設計・評価手法のご指導とご教示を賜り、また、本研究の成果である Virtual Engineering 設計環境を実製品システムに適用・評価いただきました日立三菱水力株式会社 中尾雄次主幹技師に心より御礼申し上げます。

JSM editor の要件定義から開発ならびプログラミング、さらには Virtual Engineering 設計環境のプログラミングを引き受けてくださると共に保守・運営にご尽力いただきました株式会社日立システムズ 荒木創一郎主任技師に心より御礼申し上げます。

そして、業務としての本研究の実施の許可、学位挑戦に関わるご配慮をいただきました株式会社日立インダストリアルプロダクツ 機械システム事業部 気体機システム部 長西克幸部長、岸田真弥主任技師らをはじめとする気体機システム部の皆様に、この場を借りて感謝の意を表します。



## 本研究に関する論文発表

---

- (1) 工藤卓史, 中尾雄次, 小林俊一: 「設計工程における外部ソフトウェア間入出力値群のためのマトリックススタイル表記」, 電気学会 論文誌 C, Vol. 141, No. 11, pp. 1215-1221 (2021)
- (2) 工藤卓史, 中尾雄次, 小林俊一: 「発電用水車設計における 3D CAD を中心とした 1DCAE Virtual Engineering 環境の構築」, 電気学会 論文誌 C, Vol. 143, No. 3, pp. 364-371 (2023)

## 他社所有商標などの表示

---

- ・ Microsoft, Windows, Excel および SQL Server は, 米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- ・ MapleSim は, カナダおよびその他の国における Waterloo Maple Inc. の商標または登録商標です。
- ・ SimulationX は, ドイツおよびその他の国における ESI ITI GmbH の登録商標です。
- ・ Intel は, 米国およびその他の国における Intel Corporation の商標です。
- ・ NVIDIA, GeForce は, 米国およびその他の国における NVIDIA Corporation の商標または登録商標です。
- ・ Solidworks は, 米国およびその他の国の Dassault Systems の商標または登録商標です。

付章

3D モデル，計算書

## 付章 3Dモデル，計算書

### 付.1 3Dモデル

#### 付.1.1 構成部品バラエティ

3.2節に記載したとおり，本研究では，データ内に3D-CADモデルを有する。  
5.1.3項にて説明した通り，本研究が対象とする発電機用水車には51種類の構成部品がある。次いで，本51種類の構成部品それぞれに対してバラエティ，つまり異なる形状を持つ必要があるかを判断する必要がある。

Fig. 付.1に発電機用水車構成部品がバラエティを持つ例として，4.1節で説明した，上カバーの3Dモデルを示す。Fig. 付.1において，左側の図は落差が低い，つまり上カバーに加わる水圧が低い水車用で，ベースとなるプレートは1枚の板で構成されている。これに対して，右側の図は落差が高い，つまり上カバーに加わる水圧が高い水車用で，ベースとなるプレートは2枚となり，またその間はリブで結合され強度が高い構造となっている。

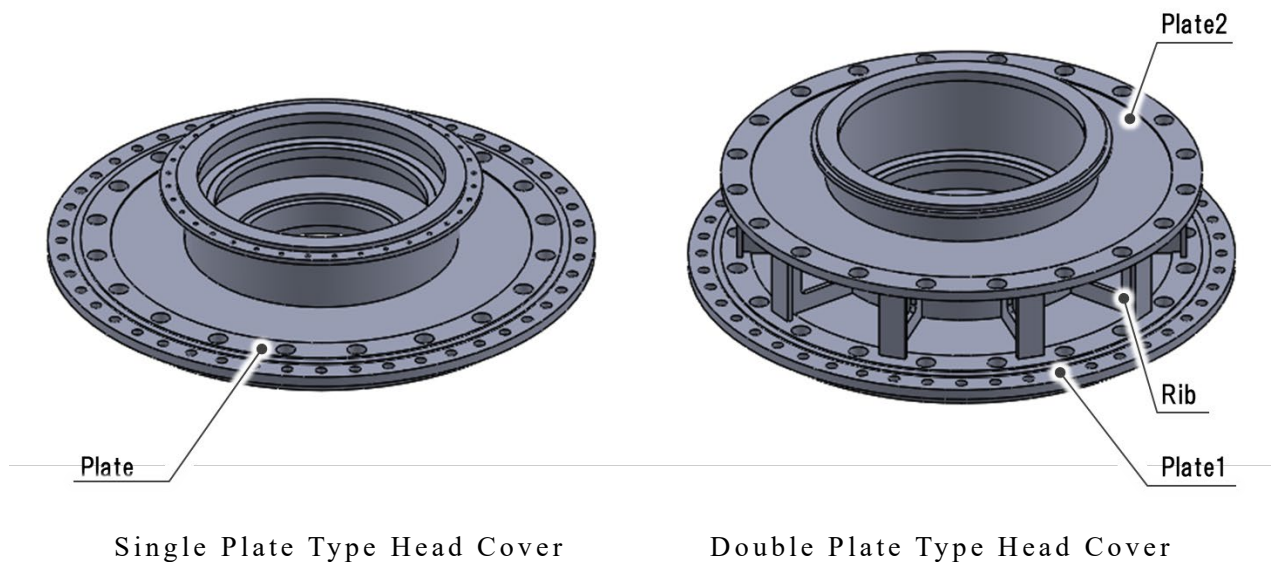


Fig. 付.1 Example of Parts Variety

## 付. 1. 2 3D-CAD データバラエティ

本研究においては、考え得るすべての部品パターンを 3D-CAD データのバラエティとして準備しておく必要はない。本研究の目的の一つは、活用技術としての 3D-CAD を中心とした Virtual Engineering 設計環境構築である。したがって、使用する 3D-CAD ソフトウェアにおいて、以下の機能、すなわちパラメトリックで表現可能な場合はバラエティと見なさず、3D-CAD データをバラエティごとに複数用意しておく必要はない。

### (1) 寸法値の変更

部品全体寸法の比例倍のみならず、1ヶ所以上の寸法を変更することで表現可能な場合も含む

### (2) 数量変更

例えば、ねじ穴の数量など。3D-CAD のパターン化機能で表現可能なものを指す。

### (3) コンフィギュレーション変更

部品内における特定部位の有無（例えば切り欠きの有無）を指す。

### (4) 上記 (1) ~ (3) の組合せ

なお、Fig. 付. 1 に示した部品は、上記の機能を用いることで、3D-CAD データとしては 1 つのバラエティとして作成可能である。

また、これらの機能は使用する 3D-CAD ソフトウェアや 3D モデルの作成方法によって利用できない場合もあるため事前に留意が必要であるとともにモデルの作成方法を考慮する必要がある。3D モデルの作成方法に依存する例を Fig. 付. 2 に示す。

Fig. 付. 2 は一般的な I 型鋼のモデルである。Case 1 は最初に定義した断面形状にフレットを持っておらず、断面形状の押し出し (Extrude) を行った後にフィレットを追加して最終形状としている。対して、Case 2 は最初に定義した断面形状にフィレットを持っており、断面形状を押し出すだけで Case 1 と同一の最終形状が完了する。ここで、バラエティとしてフィレットの無い I 型鋼のモデルが必

要になった場合、Case 1 においてはコンフィギュレーション機能によりフィレットを抑制することで表現が可能である。したがって一つのモデルでフィレットの有無を表現できるため、3D-CAD データのバラエティは一つで良い。対して Case 2 においては、ほとんどの 3D-CAD ソフトウェアにおいてはフィレットに”0（ゼロ）”を定義（入力）することができないため、フィレットの無いモデルを表現することができない。したがって、3D-CAD データには、フィレットの有無で分けた 2 つのバラエティを準備する必要がある。

メンテナンスを考慮した場合、管理する項目が少なくなることからバラエティは少ないほうが良い。しかしながら、Case 1 のモデルと比較して Case 2 のモデルはファイルサイズが大きくなる傾向がある。したがって 3D モデルの作成方法は実運用を考えて慎重に検討すべきである。

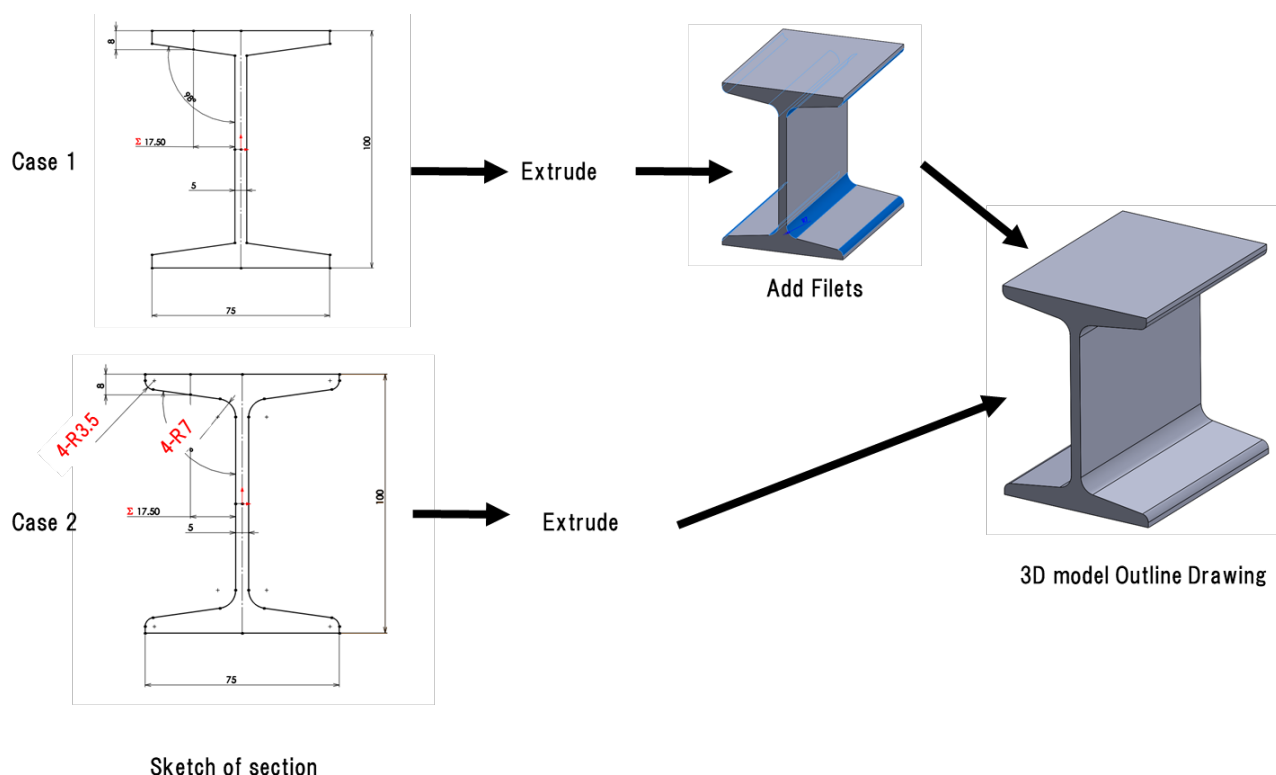


Fig. 付. 2 Example of 3D Model Part Modeling

発電機用水車構成部品のバラエティはモジュラーデザインの手法に従い、下記の手順で行った。

- (1) 製品機能構成
- (2) 機能の分類
- (3) 設計部品構成作成

詳しくは 4.1.2 項で羽根車の例を持って説明したが，発電機用水車の構成部品は，同一の機能部品であっても，サイズのバラエティの他に，幾何学的形状にもバラエティを持つ。したがって，使用する 3D-CAD のパラメトリック機能が単一のパーツファイル内で幾何学的形状を表現できるかが 3D-CAD データとしてのバラエティの要否を決める。羽根車形状を例とすると，フランス水車羽根車は比速度 50 – 350 (m-kW) の広い範囲を有し，さらに実際の製品では，個別最適化を行っているため一つとして同じ形状は無いが，本研究にて使用した 3D-CAD のパラメトリック設計機能では，羽根車の基本的な幾何学的形状である流水面は，わずか十数種類の 3D-CAD データで表現可能であった。

## 付.2 計算書

### 付.2.1 種類

3.2 節で説明したとおり，計算書は以下の 2 種類で構成される。

- Dimension Calculation Sheet
- Technical Calculation Sheet

ここで，Dimension Calculation Sheet は機械製品システムの寸法計算ルールが定義されており，機械製品システムに使用するすべての部品の寸法を決定する計算書である。また，部品のバラエティの決定も本計算書にて行われる。本計算書にて算出した値を BOM に格納し，これらの値を Base 3D Model に当てはめることで新しい機械製品システムの 3D モデルを得ることができる。

Technical Calculation Sheet は得られた 3D モデル，正しくは 3D モデルの寸法に技術的に問題が無いかを確認する計算書である。発電機用水車設計における技術計算の例を以下に示す。

- ・ 構造物強度計算
- ・ 構造物変形量計算
- ・ 起動トルク計算
- ・ ガイドベーン作動トルク計算
- ・ 漏水量計算

なお，4.2.7 項で説明した通り，技術計算の結果，満足な解が得られない，または過剰な余裕がある場合は寸法値を変更する必要がある。

## 付.2.2 設計ルール

Technical Calculation Sheet に記載される数式はすべて理論から導かれる物理式である。対して，Dimension Calculation Sheet に記載される数式は物理式の他に経験式が多く含まれる。経験式の中には，個々の設計者が経験的に使用しているルールや，特に取り決めがなく，設計者の感覚で決めてしまっているルール等が存在する場合があります，この場合は新たにルールを作成し補完する必要がある。

本研究では，これらのルールを作成するにあたり，過去に製作した発電用水車の実績を基に線形回帰にてルールを作成している。

## 付.2.3 計算書例

以下に例を用いて計算書のフォーマットを示す。なお，本例での外部ソフトウェアは Microsoft Excel である。

Fig. 付.3 はポースリエ(またはポーセリエ)の反転器(Peaucellier-Lipkin Linkage)と呼ばれるガイドなしに回転運動を直動運動に変換するリンク機構である。長さ  $a$ ， $b$  および  $c$  のリンクにて構成されており，互いのリンクの交点は回転運動を許容している。また，Fig. 付.3 中に Fixed と示した長さ  $a$  のリンクは固定されている。



る。ここで、 $x$  の値はリンクの長さ  $a$ ,  $b$  および  $c$  によってのみ決まる固定値となり、 $\theta$  の値に依存しない。したがって Y 軸に平行な直動機によって  $\theta$  の値を変化させることが可能である。

本機構は現時点で発電機用水車には使用されていないが、5.1 節で説明するガイドベーンの動作機構として将来の使用を検討しているものである。前記説明の通り  $x$  の値が固定値となるため理論上直動機にラジアル荷重が発生しないため、直動機本体やその固定が簡便となるメリットがあると考えられる。

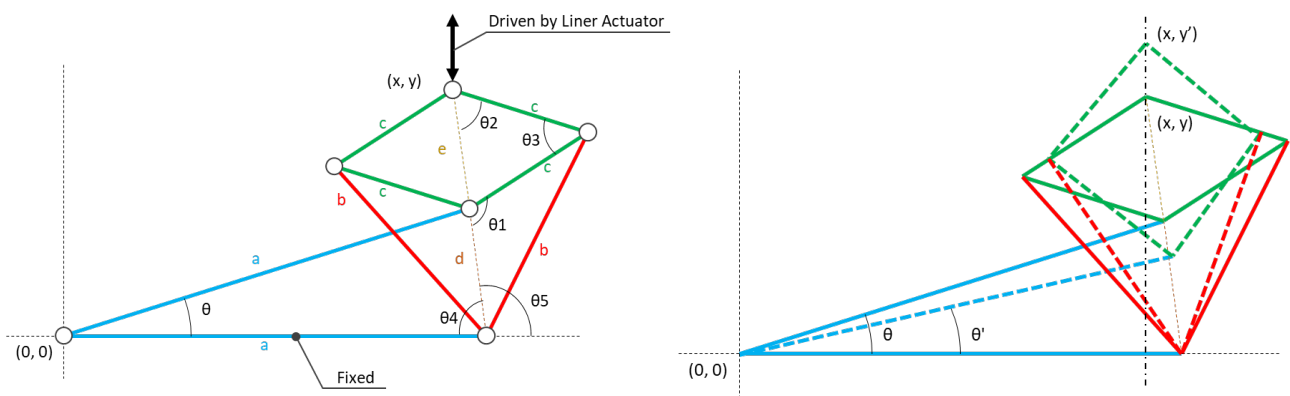


Fig. 付.3 Peaucellier-Lipkin Linkage

Job vs Spec Matrix Editor System					
BOM configuration	Spec. ID for BOM	Spec. name of Jpn.	Symbol	Unit	category1 category2
					Act_Pos
bom1	000001	LinkA_Length	a	mm	
	000002	LinkB_Length	b	mm	
	000003	LinkC_Length	c	mm	
	000004	$\theta$ _position	$\theta$	deg.	
	000005	x_position	x	mm	0
	000006	y_position	y	mm	0

Fig. 付.4 JSM of Peaucellier-Lipkin Linkage

Fig. 付. 4 はリンクの長さ  $a$ ,  $b$  および  $c$  と  $\theta$  を入力として, Job "Act\_Pos"にて直動機の位置  $x, y$  を求めることを記載した JSM である。ここで, 3.3.3 項で説明したとおり, JSM editor は外部ソフトウェア用入出力シート自動生成出力機能を有する。本機能を用いて出力した入出力シートを Fig. 付. 5 に示す。

Date						
仕様種別 Class Code	仕様名 Parameter	仕様 記号 Symbol	単位 Unit	BOM Code	値 Value	
I	LinkA_Length	a	mm	000001		
I	LinkB_Length	b	mm	000002		
I	LinkC_Length	c	mm	000003		
I	$\theta$ _position	$\theta$	deg.	000004		
O	x_position	x	mm	000005		
O	y_position	y	mm	000006		

Fig. 付. 5 I/O sheet of Peaucellier-Lipkin Linkage Calculation

本シートにおける設計システム側が必要とする情報は以下の3点である。これらの情報をもとに設計システムは BOM から本シートに値を入力, 本シートの出力値を BOM に格納する。その他の情報, 例えば仕様名 (Parameter) などは人間が目視確認するためのものであり設計システムは使用していない。

- ・ 仕様種別 (Class Code) :
  - I: (本シートへの入力値がある行を表す)
  - O: (本シートからの出力値がある行を表す)
- ・ BOM Code: DB 内のロケーション
- ・ 値 (Value) : BOM Code が示す領域に実際に格納されている値

なお、JSM editor より本シートを出力したままでは、設計システムとのやり取りは可能であるが、出力値は空（Null）である。ここで、入力値  $a, b, c$  および  $\theta$  を用いて出力値  $x$  および  $y$  を算出する処理を本シートに記載する必要がある。なお、過程は省略するが、 $x, y$  およびこれらを算出する際に使用する中間値は式（付. 1）～（付. 9）のとおりである。

$$d = a\sqrt{2(1 - \cos \theta)} \cdots \cdots \text{（付. 1）}$$

$$\theta_1 = \cos^{-1} \frac{c^2 + d^2 - b^2}{2cd} \cdots \cdots \text{（付. 2）}$$

$$\theta_2 = \pi - \theta_1 \cdots \cdots \text{（付. 3）}$$

$$\theta_3 = \pi - 2\theta_2 \cdots \cdots \text{（付. 4）}$$

$$e = \sqrt{c^2 + c^2 - 2c^2 \cos \theta_3} \cdots \cdots \text{（付. 5）}$$

$$\theta_4 = \frac{\pi - \theta}{2} \cdots \cdots \text{（付. 6）}$$

$$\theta_5 = \pi - \theta_4 \cdots \cdots \text{（付. 7）}$$

$$x = \frac{2a^2 - b^2 + c^2}{2a} \cdots \cdots \text{（付. 8）}$$

$$y = (d + e) \sin \theta_5 \cdots \cdots \text{（付. 9）}$$

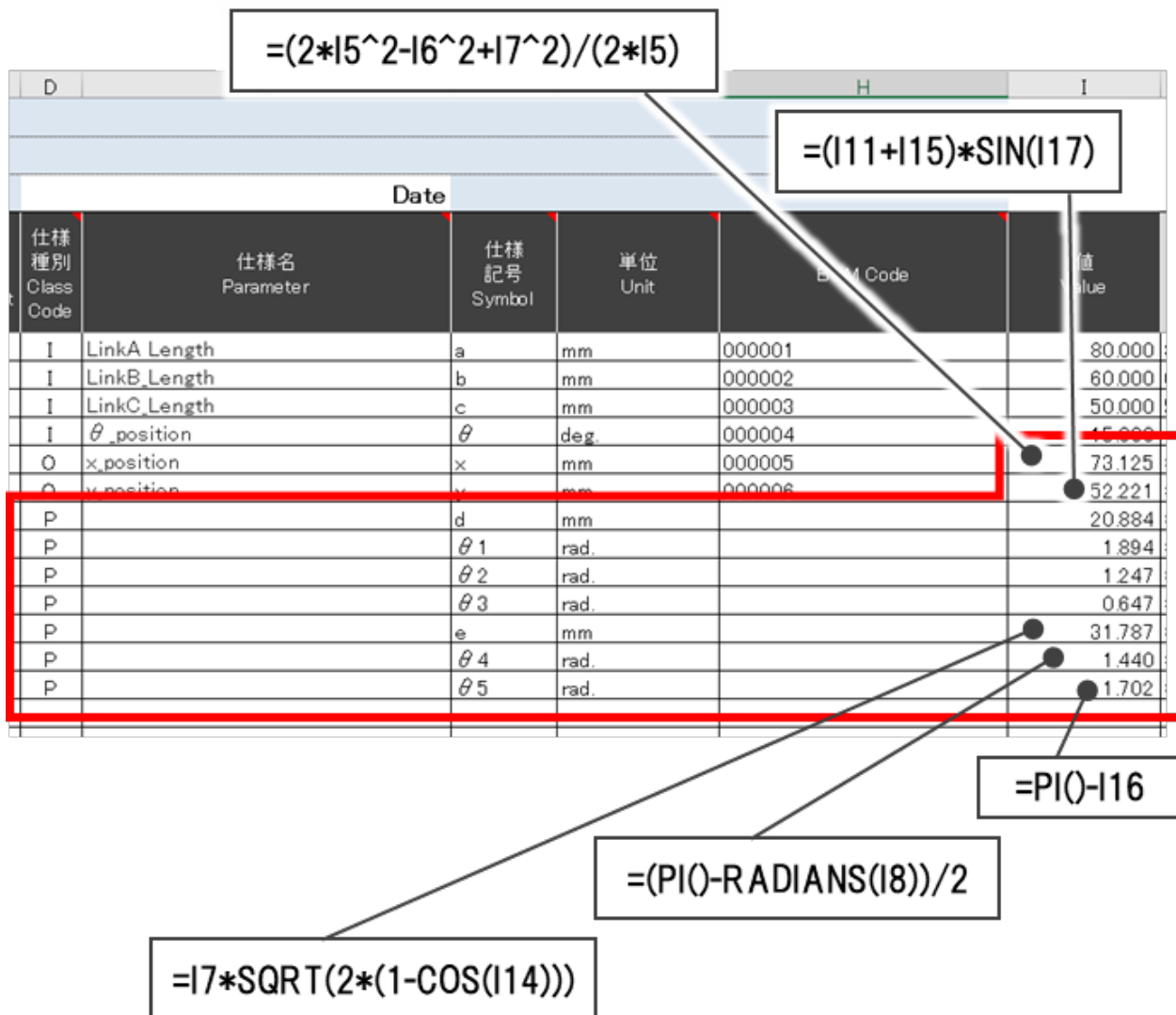


Fig. 付. 6 I/O sheet of Peaucellier-Lipkin Linkage Calculation (with Procedure)

Fig. 付. 5 のシートに処理を追記したものを Fig. 付. 6 に示す。このように必要な処理を追記することで入力値から出力値を算出する計算書が完成する。

なお、本例は入力値 4 ケ、出力値 2 ケであり、また、入力値から出力値を求める処理も簡素であり、JSM editor が出力した外部ソフトウェア用入出力シートに処理を直接記載しているが、処理は同一 Excel Book 上の別シートに記載してもかまわない。

上記に外部ソフトウェアが Microsoft Excel の場合を示した。その他の外ソフトウェアを用いる際は設計システムが外部ソフトウェア用の入出力ファイルを作成する。この動作を以下に記述する。

- ・ 設計システム構築時（メンテナンス者によるマニュアル作業）
  - (1) JSM editor より外部ソフトウェア用入出力シート自動生成出力機能を用いて入出力シート（Microsoft Excel 形式）を出力
  - (2) 入出力シートを設計システムに登録
- ・ 実設計業務時（設計システムによる自動作業）
  - (1) 入出力シートの仕様種別”I”の行に入力値を書込み
  - (2) 外部ソフトウェア入力・出力ファイルをブランクで作成
  - (3) 仕様種別”I”の行の値を使用する入力ファイルに書込み
  - (4) 入力・出力ファイルを指定して外部ソフトウェアを実行，外部ソフトウェアが実行結果を出力ファイルに書込み
  - (5) 出力ファイルの値を入出力シートの仕様種別”O”の行に書込み
  - (6) 入出力シートの仕様種別”O”の行の値を BOM に格納
  - (7) 入力・出力ファイルを削除

#### 付. 2. 4 3D-CAD の計算書としての活用

近年，3D-CAD の発達には目覚ましいものがあり，単に 3D 形状を作成することにとどまらず，さまざまな計算機能を提供している。最も分かりやすい例は質量計算である。機械製品システムの技術計算においては，構成部品の質量をもとに構造物の強度や変形を計算する機会が多くあり，それに用いる構成部品の質量を計算する作業が発生する。近年の複雑な機械製品システムにおいては，質量を計算する部品の点数も多く，またその形状も複雑である。これらすべてに対して質量を計算する計算書を個別に作成することは現実的でなく，またメンテナンスにも労力を必要とする。これに対して 3D-CAD を用いることで，”質量を得たい部品のファイルを指定し質量メンテナンス一つの処理に置き換えることが可能であり，

計算書の作成およびメンテナンスにかかる労力を大きく低減することが可能である。下記に本研究にて 3D-CAD を計算書として使用した値の例を示す。特に，発電機用水車は回転体を有し，また羽根車の形状が複雑であることから，本手法を用いて正確な慣性モーメントを容易に求めることが可能となった効果は大きい。

- ・ 質量・重心位置
- ・ 面積（表面積，特定断面積等）
- ・ 慣性モーメント
- ・ 複数部品間を跨ぐ寸法

なお，従来は設計システムのような複雑なシステムにおいては 3D-CAD を計算書として用いることを推奨していなかった。これは，特定の値をどこで計算しているかが外部から見えづらくなってしまい，メンテナンスを困難にするからである。しかし，本研究においては JSM を開発・使用したことで，特定の値をどこで計算しているかが容易に確認できるようになった。このため本研究においては積極的に 3D-CAD を計算書として使用している。