平成 18 年度 EC における国際連携の推進に関する調査研究

設計製造データの国際標準化 に関する調査研究報告書



この報告書は、(財)日本情報処理開発協会電子商取引推進センターが 競輪の補助金を受けて実施した事業の成果を取りまとめたものです。 http://keirin.jp



平成19年 3月

財団法人日本情報処理開発協会電子商取引推進センター



この報告書は、(財)日本情報処理開発協会電子商取引推進センターが競輪の補助金を受けて実施した事業の成果を取りまとめたものです。 http://keirin.jp



まえがき

この報告書は、財団法人日本情報処理開発協会が日本自転車振興会の補助金を受けて実施 した平成18年度「電子商取引の推進に関する調査研究等補助事業」の一環として取りまと めたものです。

EC の進展は著しく、世界最適地生産を求める製造業においても重要なインフラになりつつある。製品の設計・生産情報もデジタル化され、国を越えた企業間・部門間で相互にやり取りされている。このような環境の中で我が国が産業競争力を強めていくためには、個々の企業が設計開発力をより高め、その設計の意図を正しく表現し、企業間・部門間において高い品質で速やかに流通させることが重要になる。本調査研究では、主としてデジタルエンジニアリングの観点から、そのための課題を探った。

第 I 編「設計・製造高度化を支える IT」では、これからの設計・製造を踏まえて IT を活用するための課題と IT 活用の要件を報告する。

第Ⅱ編「データ品質規格とその活用」では、日本提案の国際規格案 ISO 10303-59 (製品形状データの品質の表現) の具体的活用法について種々の観点から検討したので報告する。

なお、第I編2章は委員の方々からの報告と討議を事務局でまとめたもので文責はすべて事務局にある。

また、本報告書にある会社名、団体名、製品名、システム名などは、一般にその会社や開発元の登録商標である(本文中には、TM、®は記していない)。

平成19年3月

財団法人日本情報処理開発協会電子商取引推進センター

委員名簿(敬称略)

設計・製造高度化 WG

主査 鈴木 宏正 東京大学

大高 晢彦 株式会社アイヴィス

谷本 茂樹 日本ユニシス・ソリューション株式会社

牧野 寛幸 日産自動車株式会社

松木 則夫 独立行政法人產業技術総合研究所

森 博己 デジタルプロセス株式会社

米田 俊彦 ヤマハ発動機株式会社

データ品質規格普及検討委員会

委員長 大高 晢彦 株式会社アイヴィス 副委員長 相馬 淳人 株式会社エリジオン

井上 和 株式会社富士通九州システムエンジニアリング

平岡 弘之中央大学杉村 延広大阪府立大学小林 一也富山県立大学田中 文基北海道大学

秋山 雅弘 株式会社アルモニコス

石川 義明 法政大学

(事務局)

中川 伸市 財団法人日本情報処理開発協会電子商取引推進センター 調 敏行 財団法人日本情報処理開発協会電子商取引推進センター 鈴木 勝 財団法人日本情報処理開発協会電子商取引推進センター

目 次

第1編 設	計・製造高度化を支える IT	1
1. 設計・	製造と IT	3
1.1 単	战略的 To Be に向けた IT 活用の課題	3
1.2 単	戦略的 To Be に向けた IT 活用の要件	7
2. 事例に	ご基づく IT 活用の課題と具体的要件の摘出	11
2.1	目動車車両設計の事例	11
2.1.1	自動車の特色	12
2.1.2	自動車開発の特色	13
2.1.3	自動車の設計	
2.1.4	自動車の設計と CAD 活用	17
2.1.5	設計者の道具	19
2.1.6	設計者の CAD 離れ	21
2.1.7	CAD の進化と問題点	23
2.1.8	設計者の期待に反する既存 CAD	27
2.1.9	自動車設計の道具の方向性	30
2.1.10	最後に	34
2.2	自動車エンジン設計の事例	35
2.2.1	設計とは	36
2.2.2	設計支援機能	43
2.2.3	構想設計の1シーン	49
2.2.4	まとめ	52
2.3 柞	構想設計を支援する IT システム像	54
2.3.1	話に沿った要件	55
2.3.2	要件を集約した CAD のイメージ	60
2.3.3	実現性に関する考察	62
2.3.4	付論	71
2.4 意	毀計と生産サイマルの事例	73
2.4.1	背景	74
2.4.2	現状認識	75
2.4.3	実例の紹介	76

2.4.4	課題のまとめ	78
2.5 影	計と生産サイマルのシステム要件	79
2.5.1	設計と生産サイマルの課題	80
2.5.2	設計と生産技術間の情報伝達の課題	86
2.5.3	システム要件	98
2.5.4	システムエンジニアの視点からの考察	105
第Ⅱ編 デ	· 一タ品質規格とその活用	111
1. データ	品質の向上と規格の活用	113
1.1 ラ	ータ品質向上の課題	113
1.2 ラ	ータ品質向上の要件	116
2. 規格の	活用法と実務適用の要件	119
2.1 集	! 品データ品質規格 Part 59	119
2.1.1	開発の背景	119
2.1.2	位置付け	120
2.1.3	概念と構成	122
2.1.4	利用シナリオ	129
2.1.5	インスタンス例	133
2.2 P	art 59 の活用法と拡張課題	145
2.2.1	既存 AP への組み込み方法	145
2.2.2	汎用的な組み込み方法の技術的検討	153
2.2.3	非 STEP ファイルに適用するための検討	157
2.3 P	art 59 の実務適用のための要件	161
2.3.1	データ品質向上の阻害要因	161
2.3.2	Part 59 によって見込まれる効果	163
2.3.3	今後の課題	163
付録 1.	Part 59 の品質項目の分類と一覧	165
付録 2.	AP 203 with Part 59 の統合スキーマ(Long Form)	169
付録 3.	検査対象ファイル「TorsionSpringLeft.stp」	180
付録 4.	検査対象ファイル「zone1.stp」	189

第 I 編 設計・製造高度化を支える IT

1. 設計・製造と IT

1.1 戦略的 To Be に向けた IT 活用の課題

鈴木 宏正 主査

ものづくりとは、顧客に提供する製品の設計・開発・生産・購買・販売を行う企業活動のことをいい、我が国ではものづくりを担う製造業はその経済的基盤をなす産業となっている。ものづくりの本質は設計・生産情報を生み出し、それを社内外の関連部門や下流工程に流通させて意図した製品を実現することにある。したがって、ものづくりにおいて情報技術(IT)は、ものづくり能力、あるいは企業競争力に対して決定的な影響をもつ基盤技術であり、企業は例外なくものづくりプロセスへのIT投資を拡大して来ている。

一方、IT の導入だけでは、企業の競争力強化にとって十分な条件とはならないことも確かである。すなわち、IT は、それを活用する組織能力や開発プロセスが伴わない限り、競争優位性をもたらさないのである。企業の目指すものづくり戦略に対して、人間系と IT とが作るシステムが共鳴することによって製品開発力が増幅され、大きな競争力を生むと考えることができる。本調査研究では、自動車産業を主たる対象としてきたので、それを例として述べてみよう。

最近の自動車産業の好調な業績の背景には、短リードタイム/高品質で派生車を含む豊富な車種を開発するという戦略の確立がある。新技術を他に先駆けて実用化するというのもこの戦略の一部である。このような戦略に不可欠な製品開発力を支えるものの一つが IT 技術をベースとしたフロントローディングと呼ばれる開発方法である。これは、設計・開発の早い段階で、設計情報を徹底的に3次元CADでデジタル化し、それを用いて製品性能や製造性の評価をコンピュータ上で行い、実物の試作・実験によらずに機能や製造性を検証し、問題解決を「前倒し」することによって、トータルな車両開発期間を短縮し、また、設計品質を向上させることを狙いとする。

このようなフロントローディングの実現の背景には、それを支える3次元CADを中心としたIT はもちろんのこと、IT を使う人間系、とりわけ上流から下流までの、業務組織の壁を越えた協同問題解決の仕組み(統合型組織能力【藤本】)の存在が基盤となっていることは言うまでもない。去る1980年代に米国の自動車産業のスランプに比して日本の自動車産業の業績が好調であった折、米国の自動車産業を強化するために日本の自動車産業が調査された。その結果、人間系として、重量級プロダクトマネージャーの存在と開発の上流と下流との密なコミュニケーションを行う慣行によるサイマルテイニアスエンジニアリングに日本の自動車産業の強みがあるとされた。この方法論を「仕組み」として導入すべくシステム化されたものがコンカレントエンジニアリングであり、欧米の自動車産業へも導入がなされることになる。一般に自動車開発の多くは先行製品の改良による場合が多く、開発内容やスケジュールの予測がしやすい。また、先行車両の製品情報や生産情報を参照することができ、上流と下流が共通の理解を確立しやすい。従って自動車開発はコンカレントエンジニアリン

グを実現し易い設計開発プロセスと言える。

このように欧米で積極的に行われたコンカレントエンジニアリングの研究は、日本的製品開発を学び、欧米企業に導入するためにシステム化されたものといえる。ところが、日本企業は、その欧米企業向きのシステムを IT システムとして導入することとなり、日本に逆輸入されることになった。本来、日本企業のコンカレントエンジニアリングは、日本流のシステムによって実現されるべきところであったが、欧米企業向きのシステムが IT、特に 3 次元 CAD を代表とするデジタルエンジニアリングを介して導入されることになる。ここに「ボタンの掛け違い」が起きてしまったといえよう。

それにもかかわらず、日本の自動車産業では、いわば力ずくで目前の課題を一つ一つ解決し、ITを拡張し、組織を改変することによってコンカレントエンジニアリングを強化し、上記のようなフロントローディングとして実現して来ている。特に、デジタルエンジニアリングと呼ばれる IT の圧倒的な情報処理能力、つまり製品情報の生成・表現・伝達・共有の能力が加わり、コンカレントエンジニアリングとデジタルエンジニアリングとが相乗効果を上げ、統合型組織能力を更に強化する方向に開発プロセスを変革した。これによって製品開発力は大幅に向上し、短い商品サイクルで、高品質で豊富な車種を市場へと送り出すための「大量製品開発方式」を実現している。

さて、フロントローディングが企業において強力に推進されている背景には、日本の製造業のリードタイム短縮を至上とする企業戦略がある。上記のフロントローディングの実現の過程は、リードタイム短縮を実現する最も現実的かつ効果的なアプローチとして、ITを活用する製品開発プロセスが構築されたと考えるのが自然であろう。IT抜きでは、このプロセスの実現は不可能であったことから考えれば、ITが開発プロセスを変え、その性格を強く決定付けていることになる。例えば、フロントローディングではCADデータを迅速に作成することが最も重要なポイントとなり、また、いわゆる図面工に相当するCAD専任オペレータ向けに作られた欧米型のシステムを使わざるを得ないことから、設計者とオペレータの分業化が進んでいる。最近では、自動車メーカーが、子会社として設計会社を持つことが多い。

しかし、そのようにして導入された汎用システムが、開発プロセスに対して様々な影響を与えていることは、前述のとおりである。CAD の背後にあるものの影響を大なり小なり受けていることは確かで、その功罪が問われることとなる。例えば、上記の設計者とCAD 専任のオペレータの分業化は、設計者自らが図面を書くという日本流の設計文化とは異なるものであり、設計者の作業や位置づけを変えてきている。特に、CAD データを作ることは設計ではないといいながらも、実はCAD データ作成の「前倒し」が起きており、本当の設計の部分が侵食され軽視される傾向にあると言われるし、さらには設計効率を上げるために、CAD 上での設計の標準化が徹底的に進められており、そのため「普通の設計」はできても、創造的な優れた設計には適さないといった課題も指摘されている。また、現在のCAD データは形状情報が中心であり、「CAD データ=製品情報」ではないし、さらにはCAD データが、設計者が他の技術者に伝えたい「意図」を表しているわけでもなく、本質的には昔と変わらないやり方

しかやられていない工程も多い。

以上のように、ようやくデジタルエンジニアリングのツールが曲がりなりにも開発作業をカバーできることが分かり、また、その効果も疑いの余地のないものとなり、現在の IT の弊害やその限界もはっきりとしてきている中で、企業の製品戦略に照らして、現状の製品開発プロセスに疑問を持ち、そのあるべき姿や課題を模索する企業が増えてきている。例えば、フロントローディングをより強化しようとする場合に現状のシステムをどのように改良すべきか、という課題は山積しているし、さらに、フロントローディングに代わる新しいあるべき姿を考えることも重要な視点である。企業活動として、ものづくりプロセスは日々変化し、進化を遂げている。その中で、もっとより良い製品開発のやりかたがあるのではないか、また、より良い人間系や IT があるのではないか、という問題意識が大きくなってきていることが、今回の調査からも明らかになっている。しかし、その取り組みへの考え方は一様ではない。

一つの考え方は、現状システムをベースとして、一つ一つ課題を解決し、その機能を強化していこうとするものである。あるべき姿を現状の延長上に捉え(保守的 TO BE)、これまでの取り組みをさらに継続していこうとするものである。否定的な言い方をすれば、「画期的な IT ツールなどは期待できないのだから、着実に現システムを拡張していこう。現状システムでも実現できそうなのに、実現していないものは多い。」ということであろう。しかし、本調査研究から、次第に肥大化し、複雑化しているシステムに対する危機感や、現状のシステムでは上記のように設計者の支援が貧弱なために、さらなる高品質な製品の設計には適用できず、設計品質問題へとつながるのではないかという危機感が明らかとなっている。

本調査研究の提言として、もう一つの考え方は、上記のような保守的な TO BE ではなく、 戦略的にあるべき姿(戦略的 TO BE)を設定し、従来の路線からシフトして、それに向かって 技術開発を進めていこうというものである。つまり、これまでのように日本のものづくりに 不向きなシステムを力ずくで何とかする IT 化ではなく、日本の得意とするものづくりの力 がより発揮できるようにするための IT 化を進めようとするものである。本調査研究で、企 業の有識者による議論によれば、これができなければ将来の日本のものづくりの競争力の確 保は難しいといわざるを得ないという結論を得た。

特に、このままでは設計がダメになるという危機感は強く、設計者を真に支援できる開発プロセス(人間系+IT)の実現を目指し、現状システムの延長にはない新しいコンセプトのシステムを開発することが重要である。例えば、構想設計支援、設計意図表現、感性設計技術、現物融合化技術、あいまい設計情報による擦り合わせ技術などが考えられる。そして、このような戦略的 TO BE を描くこと自体が非常に重要な課題となっている。つまり、日々変化するマーケットと、日々進化するものづくりに対して、このような TO BE 自体が固定的なものではなく、常に意識して検討されるべきであろう。そのためには、調査研究や基礎研究、実証実験を定常的に実施するための仕組みが必要である。その中で、戦略的 TO BE に向かって、新しい概念や技術を積極的にトライしていくことが必要なのだ。

現実には、現状システムの慣性力は強く、簡単にシフトすることはできないだろう。また、保守的 TO BE への道もある境界条件の下では効果的な取り組みとなることは言うまでもない。 戦略的 TO BE と保守的 TO BE がきちんとした棲み分けを行い、それらの間で、企業それぞれの境界条件において TO BE を選択できることが理想的であり、そのためにも戦略的 TO BE のための技術開発を日本のものづくりの基盤技術の一つとして推進していくことが重要である。

参考文献

【藤本】藤本隆宏 日本のもの造り哲学、日本経済新聞社、2004

1.2 戦略的 To Be に向けた IT 活用の要件

大高 晢彦 委員

本節は2章: 事例に基づく IT 活用の課題と具体的要件の摘出 に収められている2. $1\sim2$. 5 の報告の要旨を示すと共に、それらが1. 1 戦略的 To Be に向けた IT 活用の課題 とどう関係するかを述べたものである。

森は<自動車車両設計の事例>で、自動車開発工程の中でも最もデジタルエンジニアリングの活用が盛んな車体設計を中心に議論している。先ず自動車という製品の特徴、自動車開発の特徴、車体設計に関する設計者の道具の歴史的変遷を解説した後、設計者の CAD 離れが何時から何故起こったかを論じている。その骨子は以下の諸点である;

- ワイヤフレームベース(離散的曲面表現も含む)の CAD システムまでは設計者が違和感なく CAD システムを使えた。 CAD 離れは位相付き曲面モデルからソリッドモデルへと CAD システムのモデル表現が進化した時期と符合する。
- 設計の進捗にあわせ、設計の要求レベルに合わせた精度で形状を表現できる CAD システム は皆無で、常に(未だ設計が固まっていない部分も含め)精密に表現することが要求される (設計者の違和感のもと)。
- CAD システムを用いて首尾よく形状を表現するには、CAD システムの制約や癖など設計と 無関係な事項の十分な理解が不可欠なため、モデラーと呼ばれる専任集団が発生した。
- 日本の車作りは各開発工程の技術者の意識の高さ、工程をまたがった問題の柔軟な調整力などを特徴とするが、分業を前提とした欧米製の CAD システムの利用が、本来黒子であるべきシステムやモデラーを前面に出し、開発プロセスや人の役割にも影響を与えている。
- 現在デジタル化推進の旗印のもと詳細設計が構想設計の領域を侵食しており、魅力的な構想 設計ができる人材減少の一因になっている。
- 車体設計は非常に特殊で癖があるが、汎用 CAD システムはこの癖が扱えないのでモデリン グ工数が増大する。

次に更に踏み込んで現状の CAD システムの何が問題かを議論する。そのポイントは、'設計者の 思考と異なること'、'頑健性不足'、'賢くないこと'であるとし、その具体例を示している。次 にあるべき姿の議論があり、以下がその主な主張である;

- 車両の構想設計者が使える CAD システムを開発すべきである。
- 設計の詳細化は分野知識を内蔵させることにより極力自動化すべきで 'モデラー'という職種が不要な方向に進むべきである。

森の報告は1.1に述べられている現状のIT化の問題点である、いわば'力ずくのIT化'の実態の一面を表していると共に戦略的TOBEで検討すべきことの一部にも言及していると言えよう。

米田は<自動車エンジン設計の事例>と題して乗用車のエンジン設計(特に上流設計)を取り上げ、設計シーン(設計の場面)に着目し、実際の設計がどう進められているか、IT の活用状況はどうか、IT の活用効果の画期的向上には何が必要かなどを現在該氏が所属する企業で進められている次世代 CAD 開発の要求仕様をもとに詳細に論じている。

まず設計シーンという概念を、構想・計画→設計→製図と流れる開発工程の個々の工程、工程内の '部位'、部位内の '部分'に対応付ける、つまり 3 階層モデルと捉える。次に構想・計画段階で例えばエンジンのシリンダヘッド周りのバルブレイアウト、燃焼室レイアウト、動弁系レイアウトなどで用いられる 2 D 設計、これらを膨らますために必要な所に断面図や、平面図を配置する 2.5 D 設計、個々の部品の詳細化で使われる 3 D 設計の相互の関係や流れが示される。この流れに沿って使われている CAD システム、入力情報、出力情報、設計評価、チェックシート/設計シートなどが示されるが、特に参照情報は社内規格、参考図、参考書類、規格品など多岐に渡り、チェックシートなどと併せ CAD システムと十分統合されていない実態が示される。

設計の成熟度を早い段階から高め設計シーンを後戻りさせないことが本質的な設計のフロントローディングで、それが狙いであるが、そのためには各々の設計シーンごとに必要な設計評価を必要な精度で手軽に実施できること(サクサク評価)が基本要件であるとする。それに対して現実は、構想段階である程度の簡易評価はされているものの、3D形状が出来上がってから大規模FEM解析が行われそこで判明することが多い。例えばエンジン全体のパワートレーンの曲げや振動の評価はメッシュ作成に1ヶ月も要し、CAEの解析結果が出る頃は試作品はできていて物としての評価ができている。従ってCAE結果は次の試作に盛り込むというフェーズずれが発生している。つまりサクサク評価の実現は今後の大きな課題と位置付ける。更にサクサク評価実現のポイントとして、設計の質を決める設計案とその評価は個人依存性が高いので、汎用性のある仕組みで、どういう着眼で、何をやり、結果がどうで、どう決着したかを残すことの重要性を指摘している。

以上の分析にもとづき新 CAD の要求仕様として整理されたのは、キャッチフレーズ的には'設計シーンごとに3D化・簡易解析を実施し、設計意図の明確化を図り、システムに蓄積・活用すること'である。この要求仕様は更に'3D化支援'、'デジタル検証支援'、'データ連携支援'、'設計意図支援'、に分類されて一段詳細化されている。

米田は更に設計シーンとサクサク評価について具体例を示している。これはクランク室形状の構想設計を1つのシーンとして取り上げたものである。

米田の報告は1. 1の'戦略的 TO BE'に係わるもので、自動車エンジンの構想設計を主題に、現状の IT 活用の問題点は何か、将来どうすべきかを詳しく論じたものと言えよう。

大高は<構想設計を支援するITシステム像>で米田が提示した要件のIT面から見た課題の整理と実現可能性の検討を試みている。設計シーン、協調設計、社内規格等の設計作業を制御する情報の扱い、設計サイクルで使える高速・適切精度の解析機能、設計記録、設計者用語で使えるCAD、設計のストーリー化、設計チェックの効率化などの要件を再確認し、要件を集約したCADのイメージとして;

- (1) 設計者がこめた意図や意味がデータに反映されていて他者にも理解可能であり、システムも それに応じた振る舞いをする CAD
- (2) 設計シーン単位のデータ構造を持ち、シーン単位の設計評価が可能であると共に評価結果と 判断が蓄積・活用できる CAD

を示している。更にその実現性に関して、データ表現、システムイメージ、機能体系の諸観点から個々の要件の実現可能性を検討している。なお以下の事項も要検討課題として指摘している。

- (1) 設計者間、設計者と生産技術者間の緊密な・効率的な相互連携を支える IT 技術開発
- (2) 情物融合化による情報モデルの信頼性向上と活用性拡大

牧野は '設計と生産サイマルの事例'で車両の詳細設計から生産にいたるデジタル化の実態と課題を報告している。

フロントローディングやグローバル化に伴う技術工数の増大、デジタルフェーズでの仕事のやり方の整備などが議論されている。昨今、品質不良が問われている実態は、従来から英知を集めて蓄積してきた開発プロセスとデジタル化によって可能になるフロントローディングの折り合いをどの辺にすべきかの検討が急務であることを示している。グローバル化に伴う生産地域の拡大による技術検討工数増大も単に工数増大というだけでなく、地域特性を考慮した製品開発はいかにあるべきかという課題である。地域によって人の技術力、技術量、意識に差があり、使用できる装置の質・量も異なる。また適用される基準の差異もある。これらを踏まえた解決が必要である。デジタルフェーズの仕事のやり方の整備も着眼を誤ると間違った方向に進む。従来から英知を集めて蓄積してきた開発プロセスをデジタルツールに合わせて変えるのか、開発プロセスに合わせてデジタルツールを変えるのかが考え方の分岐点である。基本的には明らかに後者中心で進めるべきで、間違っても何時変わるか知れぬツールに合わせて開発プロセスを変えることがあってはならない。基本的に後者のスタンスで、人間の技術力をより良く発揮でき、開発効率もより高められるなら開発プロセスのマイナーチェンジ(改良)は行うという考え方であろう。

牧野は課題を一段掘り下げて、設計標準書、不具合情報フィードバックが帳票データベース+ Email に電子化されただけで工数低減ができていないという実態、CAD と周辺システムの連動 に関しては CAD と不具合情報が連動していないこと、CAD と周辺のシミュレーション、ナレッジなどとの連携が希薄なこと、本来 CAD は物と一緒で情報のハブでありたい(そこから全部の情報が手繰れる)が現状の CAD は参照される側の位置付けであること等を挙げている。更に、

'設計・生産技術含めて CAD と品質の情報が連携していない'と指摘する。これも重要な視点である。従来は'品質規準書'に準拠して営々と品質の作りこみがされたはずだが、デジタル化

に伴い品質規準書がほこりをかぶっていないか?前述の社内規準書の扱いと同じで、電子化された品質規準書が CAD と統合され、黙っていても品質規準の観点から注意や、アドバイスがユーザに示される仕組みが望まれる。 '品質チェックしてそれをフィードバックしても、次に修正されたものにそれが反映されているかどうかわからないので繰り返しのチェック工数が多くなる'という指摘は品質規準書 or 設計評価項目と CAD の適切な統合の必要性を示している。

設計と生産サイマルの要件を整理すると以下の通りである。

- (1) 設計要件確認・評価方法について設計側と生産技術側で最適なプロセス(約束事・プロセス) を再検討することの必要性
- (2) 設計段階で生産要件をチェックできる仕組みの構築、設計から生産性検討のプロセスのシーンを変えられる CAD、CAE のより高度な活用法の追求
- (3) バラバラなシミュレーション環境の統合(不必要な分業の廃止)

以上牧野の報告は'力ずくの IT 化'がたどり着いた現状を良く表している。牧野が示す要件は、現状のデジタル化の延長としてその改善で解決できるものも一部あるが、(1) ~ (3)の本質的課題は'戦略的 To Be'の視点での検討が必要なことを示している。

森は<設計と生産サイマルのシステム要件>で、牧野が示した要件の中の'チェック工数削減'に焦点を当ててシステム化要件を検討している。'設計変更と製造要件チェックの関係の課題'について、設計変更に対応しチェックすべき製造要件が 100%正確に最小個数に限定できることが理想であるが、必ず関係するもの、確実に関係しないものに加えて、連鎖的に関係するかもしれないもの、関係するか否かが分からないものが多いためチェック工数増大を招いていると指摘している。また、多種多様な設計変更に伴い設計形状や属性が変わっているか否かが不明なためにチェックせざるを得ない事項も多い。従って製造要件チェックの効率化を狙った IT 化は設計情報の伝達のシステム化が1つの重要な要件となると指摘している。

2. 事例に基づく IT 活用の課題と具体的要件の摘出

2.1 自動車車両設計の事例

自動車 車両設計の事例

デジタルプロセス株式会社 森 博己

図 I-2-1 自動車 車両設計の事例

本質のところは後半少し触れているのですが、それに行き着くまでの自動車の設計の構想段階というものを、逆にたどって、何故こういった特色がでるのかを1頁目、2頁目... とまとめていきました。最後にどうするかということでまとめました。

2.1.1 自動車の特色

自動車の特色

- 値段*量産数の積において、最も高額な一般向け商品
- 世間の目に触れ、ブランド志向の強い商品
- 意匠により、売れ行きが最も左右される商品
- 仕向け地、グレード、色などバリエーションが多い商品
- 定期的にモデルチェンジを繰り返される商品
- 嗜好によって買い替えられる商品
- 安全、環境、税など、制約レギュレーションが多い商品

図 I- 2-2 自動車の特色

自動車という商品の特色を上げてみました。

- ・自動車は高額で量産の数も多い商品です。したがって値段*量産数の積において、最も高額な一般向け商品です。
- ・世間の目に触れやすく、だからある一面でブランド志向の強い商品です。
- ・とくに最近は意匠(スタイル)が良いかどうかにより、売れ行きが決まってしまう商品です。
- ・仕向け地、グレード、色などを含めると、ものすごい数のバリエーションを持っている商品で す。
- ・定期的にモデルチェンジを繰り返される商品です。日本の場合は大概4年サイクルでモデルチェンジがされています。
- ・嗜好によって買い替えられる商品です。家電の白物などは壊れたから買い換えるのが最も多いが、自動車はいい車が出たと思った時に買い換えられます。
- ・安全、環境、税金など、いろんなレギュレーションで制約されている面が多い商品です。

2.1.2 自動車開発の特色

自動車開発の特色

- 百年近い歴史と、年間で多くの車種開発が行われた為に、開発プロセスが定常化している
- 仕向け地、グレード、色などバリエーション管理が必要
- 量産されることを前提に開発
- 1つの製品開発に多くの人が短期間に関わる
- 一人の分担する領域は極めて狭く、相互の情報流通オーバーヘッド が大きい
- 搭載ユニットは、概略の仕組みや機能に、大きな変更はない
- 自由曲面で造られた意匠に、車両のほぼ全体が関連している
- 車両部品(BIW、内装トリム)には、僅かな平面はあるものの、直線 は殆ど存在しない特殊な形状である
- 重要な強度部材となる車体部品(BIW)は、全て板を圧造して製造されるために、ソリッドではなくOpenShell + 板厚で表現される
- 圧造&インジュクション用金型の製造が大きなコストを占める

図 I-2-3 自動車開発の特色

開発をするという点での自動車の特色は何でしょうか。

- ・T型フォードから考えると百年近い歴史と、年間で多くの車種開発が行われています。ともかく何十回と繰り返して同じようなものを作っているので、開発プロセスが非常に定型化しています。例えば、ダムとか橋を作るときに、何月何日に通行させるという見通しを立てても、伸ばされることがあるが、自動車の場合は4月発表だと決めると、4月に発表できてしまう。この作業はどのくらいの単位的な期間がいるかという標準値がでている。そういった面でプロセスが定型化しています。
- ・仕向け地、グレード、色などバリエーションが多いのでその管理が重要になってきます。
- ・試作されるだけで終わるわけではなくて量産されることを前提に開発する。また量産の数も多い。
- ・1つの製品開発に多くの人が短期間に関わる。延べ人数では飛行機、ロケット、原子力発電所は大きいが、2~3年の短期間に多くの人が関わる点では自動車開発の特色の一つでしょう。
- ・一人の分担する領域は極めて狭くて、そのために情報の流通が重要になってきて、そのオーバーヘッドが大きい。

それに関連して、

・搭載ユニットは、概略の仕組みや機能に、大きな変更はありません。タイヤは昔から4つです し、ワイパーはあのような形をしていますし、エンジンはまだレシプロかディーゼルかで大きな 変更はなく、車両開発とは大きなドンガラの開発という意味です。

ここから CAD 屋らしい表現になってきますが、

- ・自由曲面で造られた意匠に、車両のほぼ全体が関連している。外側と、内側の内装の意匠も含めてです。
- ・ユニットと区別して、ホワイトボディーと内装のトリム部品を総称して車両部品と表現しました。こういったものは取り付け面とか僅かな平面はあるものの、直線は殆ど存在しない特殊な形状です。
- ・重要な強度部材となる車体部品(BIW)は、全て板をプレスして製造されるために、自動車会社のホワイトボディーの表現はソリッドではなくてOpenなShellに板厚情報を付加しています。
- ・プレスとかインジェクション用金型の製造が大きなコストを占めています。

2.1.3 自動車の設計

主要な(車両系)設計の役割

• 意匠設計

- イメージスケッチ⇒コンセプトスタイリング(CADもしくはクレイ)
- コンセプトスタイリング⇒CLASS1サーフェイス

• 車体設計

- 外板形状の意匠を保ちながら、強度、重量、成形性を考慮しながら、内板形状(CLASS-2)を決定する
- 内臓ユニットの配置(車両計画として独立している場合もある)

艤装設計

- 意匠を基本とした表面形状と、内板パネル形状、内臓物との関連性を決める (干渉、取付性など、音震?)
- 具体的な作業は、トリム部品メーカーへ設計委託
- シート類も同様に外部委託

シャシー設計

- シミュレーションと実験による性能設計重視 (操安性、強度音振)

図 I-2-4 主要な(車両系)設計の役割

このような前提のなかで、自動車の設計の業務としてどのような区分があるか、分けてみました。 まず、車両とユニットに分けて、ここでは車両だけをピックアップしてみました。上流から、

• 意匠設計

造型という表現でいう会社もある。具体的には、イメージスケッチからコンセプトスタイリングを Alias のようなな CAD を使ったりクレイモデルを作って行います。

コンセプトスタイリングから Class1 サーフェイス (スタイリングの曲面)を造ります。

自動車会社によっては ClassA、ClassB という言い方をします。Class 1 は意匠表面の最終表現、Class2 は意匠として目に見えない内板部分です。大域的滑らかさを表す言葉ではありません。Class1 としては自動車の外板や、インストルメントパネルの内装があります。

• 車体設計

外板形状の意匠をいじらないで、強度、重量、成形性などを考慮しながら、内板形状(Class2)を決定する作業があります。

もうひとつの役割は、エンジンルームのレイアウト、リアの燃料タンクの配置とかの主要な部品・ ユニットの配置業務を行います。

自動車会社によっては車両計画部署として独立した組織をもっている場合もあります。

→パワーウィンドーレギュレータも内部ユニットに入れているのですか。

←とりあえず入れています。車両計画部は主にはエンジンルームのレイアウトをしています。車体設計部でやっている自動車会社もあります。

• 艤装設計

意匠を基本とした表面の形状は造型で決まります。内板パネル形状は車体設計で決まります。その間に何かあった場合は、内蔵物との関連性を決めますが、何も無ければ、何も決めません。干渉、取付性など、また音震などの考慮は必要です。

最大の仕事は、トリム部品メーカーへ設計委託をすることです。現在の自動車は、昔のように鉄板がじかに目に触れることはありません、樹脂の部品が貼ってあります。それらを総称してトリム部品といいます。

シート類も同様に外部委託します。

シャシー設計

車両にはいるかどうか、各社でいろいろ違うのですが、シミュレーションと実験による性能設計 を重視する部署です。操安性とか、強度音振を決めます。

車体設計にくらべると、形にあまりとらわれない、形よりも性能を重視するところです。

- →車体設計とシャシー設計の境目はどこですか。
- ←ボディーの板物は全部車体です。シャシー設計は、ブレーキローターが付いているアクスル、 サスペンションを支えるアーム(アッパーアーム、ロアアーム)とブッシュ。そこまでがシャシー設計です。ボディーに取り付けるための補強の板は車体設計です。位置決めはシャシー設計です。ブレーキの配管はシャシー設計で、ブレーキのペダルは車体設計です。これは某自動車会社の例です。

2.1.4 自動車の設計と CAD 活用

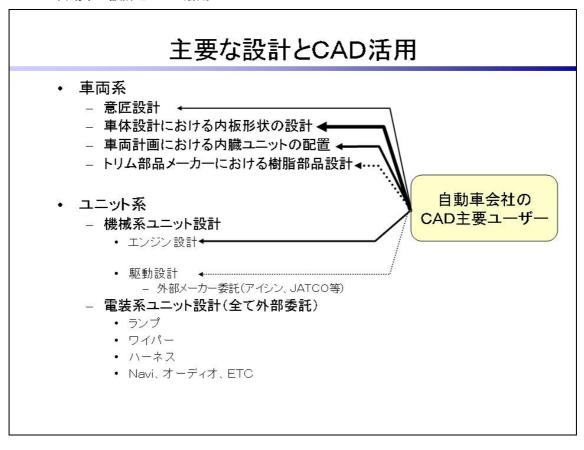


図 I-2-5 主要な設計と CAD 活用

CAD を使っている設計者は、総じて以下のようになります。

車両系では

- 一意匠設計
- 車体設計における内板形状の設計
- 車両計画における内蔵ユニットの配置
- ートリム部品メーカーにおける樹脂部品設計
- ユニット系では
- -機械系ユニット設計
 - エンジン設計
 - 駆動設計

トランスミッション、デフ、ドライブシャフトなどの設計で、大概の自動車会社は外部メーカー委託しています。

-電装系ユニット設計

ランプ、ワイパー、ハーネス、Navi、オーディオなど沢山種類があって全て外部委託していま

す。

自動車会社の中の CAD のユーザを線の太さで表しました。一番大きいのは 車体設計の内板に関する部分、つぎに車両計画、つぎにスタイリングの部分です。

エンジン設計は多いのですが、サイクルが違うので、車両の方が使っている頻度が大きい。 CAD の効率を上げるとしたら、ユニット、車体のどっちの方を向いたらよいかという議論をしたことがあります。ユニットやエンジンはライフサイクルが長いので、効率が上がったときの成果は車体の方が効果がでます。

- →最近の車は機械製品というよりはエレクトロニクス製品に近くなってきているという状況がありますが、その状況が自動車屋さんを何か変えていますか。
- ←ここで述べるほどの大きな変更には至っていないと思う。電装系を全て外部委託などと書きましたが、車両の電気システムとして取りまとめる部署は存在します。それは従来は無かった概念ですが、それで実際何が変わったかというとハーネスが少し太くなった、東が増えたということはあるが、今までと同じ程度の影響です。
- →ドイツの ProSTEP のレポートを見ていると、電子系と機械系をどうやって相性よくするのか、機械系中心のデータモデルを見直さなければいけないのではないかという議論が始まっています。 ←そういう議論をしている人もいますが、自動車会社の中では、電子という点はさておいて、線としての認識しかない。こういう電装品の回路という側面は、全く別の CAD でしています。その論理情報を使って、3 次元形状 CAD に持ってくるときは、ハーネスという形に落ちている。 車体の3次元 CAD に持って来るときには、そのハーネスに10本入っているか、20本入っているか関係ないわけですね。このルートを通る、このくらいの断面積をもったものという意識しかまだない。それで今のところの業務は十分です。
- →ということは、データモデルはあくまで機械部品単位の区切りになっているということですね。 →図中の矢印は、その元々の、オリジナルのデータを作っている人であることは意識しておく必要がある。だからユーザという言葉は少し誤解をまねく。
- ←そうですね、自動車のエンジンルームは一番巨大なのですが、それを創る人よりも、参照している人の方が多いですね。シャシー設計は自分のデータは軽いのに、その周囲を全部みているのですね。当たるか当たらないかを。

2.1.5 設計者の道具

設計者の道具(1)

~1970

- 鉛筆、バッテン、雲形定規、ドラフター、T定規、コンパス、デバイダ・・・

1970年代

- 造形線図をコンピュータ処理開始⇒外板はCAMで切削

1980年代

- 内板を含めてCAD化
- プレスメーカーと、大手の樹脂メーカーからCAD化が始まる

ここまでは良かった!! CADは設計者の道具だった

- 殆どの内板部品はワイヤーフレームだった
- 外板、及び一部の大物部品のみサーフェイス化⇒金型のCAM 切削の為

図 I- 2-6 設計者の道具(1)

設計者の道具という側面から見ていきます。

CAD が出てきたのは 1970 年になってからなので、それまでは、鉛筆、バッテン、雲形定規、ドラフター、T定規、コンパス、デバイダ等を使っていました。

1970 年代になると、造形の線図だけがコンピュータ処理されました。これはすぐに外板の CAM 切削につなげました。ですから、自動車は2次元から始まったわけではなくて、最初は曲線しか 扱わない3次元の線から始まり CAM につなげました。

1980 年代になると、内板を含めて CAD になっていきます。ですから図面を書くならば CAD で書きました。また、自動車メーカーだけでなく、プレスメーカーと大手の樹脂メーカー(トリム部品メーカー)から CAD 化が始まりました。

ここまでは CAD は設計者が使っていました。ただ、殆どの内板部品はワイヤーフレームでした。 外板、及び内板の一部の大物部品(ドアのインナーパネル、ボンネットのレインフォース)は内

設計者の道具(2)

1990年代

- 干渉問題を試作する前に発見したい⇒デジタルモックアップ
- デジタルモックアップのためのは全部品サーフェイス化
- 専用の道具が使われ始めた⇒専門家集団に依頼
 - 下流でCAMに繋がる部品は曲面専用CAD 1990
 - ワイヤーフレームからの面張専用ソフト 1995
 - ワイヤーフレームを経由しないソリッドCAD 1997

2000年代

- ソリッドCADへの移行が完了し、設計者はCADから解放!?
- 設計者はメモでモデラーに指示し、結果をViewerで見るだけ

2010年代

- 設計行為の定常化が進み、設計の主体は(ベテランモデラーを 抱える)外部設計会社へ
- コスト削減要請により、外部設計会社の海外転出が完了

図 I-2-7 設計者の道具(2)

1990年代になると、干渉問題を試作する前に発見したいので、デジタルモックアップを考えました。エンジンルームのデジタルモックアップのために、全エンジン部品をサーフェイス化しなければなりませんでした。そのために専用の道具が使われ始めました。

専用の道具というのは、ワイヤーからラフにでも曲面を張る道具です。

1997 年からはソリッド CAD を使い始めました。こういった道具は設計者ではなくて、モデリングの専門家集団に依頼するようになりました。

2000年代に入ると、ソリッド CAD への移行が完了し、設計者は CAD から解放、メモでモデラーに指示して、結果を Viewer で見るだけになりました。

ここからは憶測ですが 2010 年代になると、設計行為の定形化が進み、設計の主体はベテランモデラーを抱える外部設計会社になるでしょう。米国などはそちらの方向に進んでいます。また、コスト削減要請により、外部設計会社は海外転出し、国内には残っていないでしょう。

2.1.6 設計者の CAD 離れ

設計者のCAD離れ

設計者とCADの関係

- ワイヤーフレームまでは使っていた
 - 初期にはキーボード・アレルギーで、何割かの設計者が抵抗を示したが・・・・
- ソリッドCADでは、断面の指示もメモ書きに
 - やっぱりソリッドCADの問題?
- Viewerなら使っている
 - 3Dの分かり易さは、ちゃんと理解している
- メモ書きではメール添付できないからPowerPointで描く
 - こんな道具でも使おうとするのは?
 - 使い難くても、分かりやすい道具なら使うのか?

図 I-2-8 設計者の CAD 離れ

設計者と CAD の関係を整理してみますと、ワイヤーフレームまでは使っていました。 図面を CAD で書き始めた初期にはキーボード・アレルギーで、何割かの設計者が抵抗を示しま したが、これは 20 年前の話になりました。

→スタイリングデザイナーは入っていますか。

←ここではボディー設計を中心に書いています。スタイリングデザイナー、すなわちイメージスケッチを描く人は Alias で描く人もいれば、昔ながらの手で描くのが好きな人もいます。クレイモデラーやスタイリングモデラーはもう完全に CAD を使っています。

各社の設計者はソリッド CAD は使わなくなって来ています。断面の指示もメモ書きになり、ポンチ絵になりました。でも一方では、Viewer は使っています。

3Dの分かり易さは、きちんと理解しています。メモ書きではメール添付できないから PowerPoint で描くようになりました。

PowerPoint でポンチ絵を描くのも大変なときがあります。でも、こんな道具でも使おうとする

のは何故だろうか、使い難くても、分かりやすい、あるいは自分が使える道具なら使うのだろうかと、いくつかのキーワードを拾ってみました。

2.1.7 CAD の進化と問題点

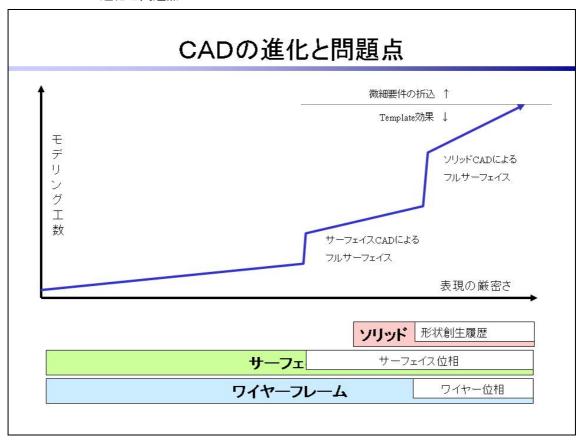


図 I- 2-9 CAD の進化と問題点

図の横軸の下側は自動車のパネル設計に適用されていた CAD の表現です。

ワイヤーフレームはずっと昔からありました。それと同じ時期からサーフェイスもありました。 ただこの場合のサーフェイスは位相がありません、単面です、基準面しかない。で、あるときからサーフェイスに位相が付いたものが出ました。

ワイヤーフレームは直線、曲線、円がバラバラでした。ワイヤーフレームに位相が付いたものは ソリッド CAD が出てくるタイミングでしか使っていません。

ソリッドの概念は早くからでているのですが、一般的にソリッドモデルといわれているものは、 形状創生履歴とかパラメトリックだとか色々な言い方をしますが、フィーチャの機能が付いたも のです。

必要があるかないかを別にして、右に行くほど表現が厳密になっていきます。 図の縦軸にモデリング工数をとると、大体このような線が描けます。 ワイヤーフレームで描いていた頃は、時代とともに細かくは描いていくのですが、あるとき位相付きサーフェイスでやろうということになったに瞬間に工数がはねあがりました。

サーフェイスでのモデリングが詳細化していくと、工数は滑らかにあがっていくが、最後にソリッド CAD になった時に一気にあがります。

ソリッド CAD になった後も、微細なフィレットを付けると、フィレットを入れない場合の工数の倍になります。

- 一番右の矢印の尖端は、微細な要件の90%以上がフィレットなのですが、フィレットを織り込むと上がります。逆にソリッド CAD のパラメトリック機能を活かせたテンプレートの効果が出ると、もう少し下がります。
- →モデルの「ものになる厳密さ」という最適ポイントがあるのではないかと思いますが、そういう意味でみると、どう見ればよいのですか。
- →この図に描かれた工数には、要求データのレベルとか量が入っているので、単位(作業)あたりの作成工数はもう少しマイルドに上がるのではないでしょうか。モデラーの数はここまで増えていないですから。
- ←ご質問に関して、私がこの図でもう一ついいたかったことは、どこまで、どういう細かさまで作るのが一番適当なのかが本当はある。構想段階から詳細段階に進むにつれて、詳細化していくのは当たり前なのですが、ところが、あるとき、まだ早い段階なのに細か過ぎるデータを作らざるを得ない状況になっている。まだ完全に詰まってないのに、実際形状まで表現してしまうものが結構あると思います。
- →設計の要求レベルに合わせた精度で、ざくっと形状を作ってくれる CAD がない、常に厳密に作るしかなくなる。だから設計の要求レベルと合わない。
- \rightarrow フィレットが典型なのですが、本当は、微細フィレットは CAM 側からすれば掛けないでくれた方が助かるのです。見込みをしたり、色々なことをするので、フィレットをわざわざ外さなければならない。CAM 側から見ると余計な世話ですが、車両解析 CAE ではフィレットは効くので掛けていなければだめなのです。
- ←私の経験では、微細フィレットを掛けていないと、一番問題が出たのはデジタルモックアップの干渉に影響がでます。それから、いまお話があった解析のメッシュに影響がでること、もう一つは見栄えです。設計者は、ここのフィレットはきちんと付けて、ここは関係ないからいいというのがあるはずです。でも、そんなことを分けないでモデラーに張ってもらっています。
- →典型的な設計の言葉は、指示なき R は XX です。
- ←基準として、このタイミングでは詳細なデータを提供して下さいといっています。異なるタイミングであれば、フィレットなしのデータを出すというのはありますが、設計に関心のあるフィレットだけ張れという定義はないのです。

- →本当は、そこは CAD が利口になってくれればいいですね。
- →結局のところ、位相とフィレットが癌だといっているのですか。
- ←いいえ。表現を実体に近くみせるために(それが、そのとき、いいのかどうかはわかりませんが)、ものすごい工数をかけている。
- →一番右端が少ない工数で出来るのであれば許してあげるということですね。
- ←例えば、車体の設計者が内板を所詮ワイヤーフレームで、頭の中で考えて、こういうように通せばいいなと思っているのを、何でサーフェイスにまでするのか。そうすると、まだ決まっていないことまで作り込んでいかなければならない。もしかしたらワイヤーフレームを変えるかもしれないのにです。
- →それは日本の自動車屋さんだけの話をしているのでしょうか。ドイツの自動車屋さんとか、アメリカのフォードも本当にワイヤー主導でしょうか。スタイルデザインは、明らかに彼らは面でやっています。日本のように線を綺麗にした後、面を当てはめるという発想ではなく、面がマスターデータだと彼らは言っているわけです。
- ←いや、それも振り返ってみると、スタイリングデザイナーは線です、キーラインです。イタル デザインがそうですから。かならず線を描く。
- →でもそれは面を創るための線でしょう、つまりあくまでも彼らにとってマスターデータは面な のではないでしょうか。日本のスタイルデザイナーは線ばかり一生懸命描くから、面はあくまで も間を埋めるものでしかない。
- ←スタイリングデザインに関してはどちらも同じだと思っています。最初に注目するのはキーラインで、その次に面を張って全体のイメージをとるという発想は変わらないと思います。
- →例えば、製品データ品質をチェックして悪いときに、面か線のどちらを直すかというときに、 明らかな違いを感じたのですよ。
- ←それは狭義の製品データ品質ですね。多分デザイナーから見た場合はこの面のうねりからどうしても納得できない形になっているという場合だと思うのですよ。Alias では面のつなぎは考えませんから。
- 形がカッコよいかどうかを決めるコンセプトスタイリング段階においては、キーラインとそれを繋ぐ面のボリューム感とかいイライトの流れとかで、線が主(で面が従)だと思います。イタリアにデザインを委託した場合も、アウトプットは線と発砲スチロールで作ったモックアップです。発砲スチロールも、キーラインが貼ってあります。そこから CAD にしていくわけです。ほとんどそこで車の顔形は決まります。
- →日本の自動車会社の造形モデラーが書かれたスタイルデザインの本を見せたところ、「明らか違うところがある、ずっと線で押し通そうとしているが私たちは...」との話があった。

→日本もいろいろな理由から、設計者と図面工が分業化されている欧米文化の中で作られた CAD を使うことになって、日本も分業になってしまったというストーリーは判るのですが、この図でいっているモデリング工数とか、ソリッド化によって、いくら内製で頑張ってやっていても分業になったのでしょうか。分業化された本質は、欧米性の CAD を使ったからなのか、工数が増えたときにどう頑張っても設計者が使えるものは難しいのか。

←自動車会社が内製の CAD を作っているときには、大概焦点を車体だとかに決めるわけです。 市販の CAD は狙いが定まっていません。航空機やジェットエンジンから生まれたものです、これらから車体の設計が出来るものでしょうか。

車体は今まで述べたように、ものすごく癖のあるものです。もしかすると、自動車会社が車体の 内板だけを焦点にやっていれば、作れたのではないかと、私は思っています。

- →工数の問題というよりもスコープの問題ということですか。
- ←一つの例を挙げると、車体で一番主要な機能はスイープです。そのスイープが全然違う。普通のスイープは断面一定ですが、車体設計ではアウタパネルからのオフセット一定なのです。そういうものが機能してもないので特注して作らせるわけです。そのくらい、車体は市販の CAD からみれば癖があります。
- →ソリッド化であっても、仕様を書けば、出来たのではないかということですね。
- ←あるレベルでは出来たと思います。私としては、車体というのは、非常に特殊なものであるということを、述べたつもりです。

2.1.8 設計者の期待に反する既存 CAD

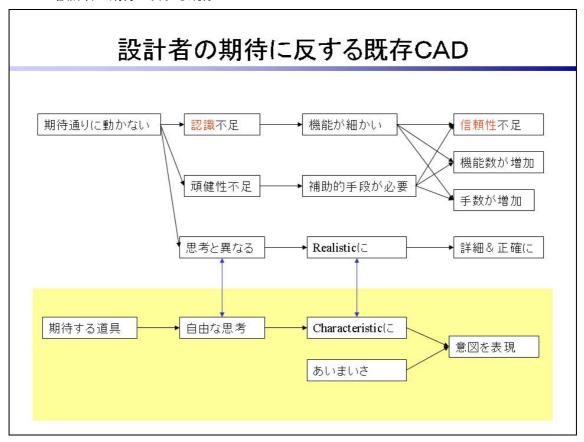


図 I-2-10 設計者の期待に反する既存 CAD

期待通りに動かないというのはどういうことかというと、

「認識不足」はコンピュータが賢くないということです。ひとつの例を挙げると、内板が一つの 板だという前提があれば、もっともっとコンピュータは頭が良いことができる。コンピュータは 板かどうかを知らない。頭が悪いから、機能が細かいのです。そして、機能が細かくて、補助的 な手段が必要だから、機能の数が増えて、手数が増える。

それから、ここにフィレットを掛けたいのに、掛からないのが一杯あるという頑健性不足。頑健 性が足りないから、切った貼ったの補助的な手段がいる。

この形をつくりたいのに、このアプローチだけしか出してくれないため、思考と異なる。 今までの CAD は何を狙っているかというと、より現物に近いように、近いようにをやっていた はずなのです。それを「Realistic に」と書いていますが、最終的に詳細でかつ正確なものを狙っ ていた。 でも、設計者は、多分違う。図の下部にその反意語を並べてみました。

設計者が期待する道具というのは、「自由な思考」をしたい、結果的にこの形を作るときに、ワイヤーを張って面で繋いでもいいし、最初から面で作ってもいいし、立体を削ってもいいし、自由な発想で形をつくりたいはずです。「Realistic に」形を作るのではなくて、自分の意図をはっきり示したい。

例えば、微細なフィレットが張られた内板のパネルのシェーディング画像では、意図は全くわかりません。とってしまえば、意図はわかります。シェーディングでなくて、ワイヤーで表示したらもっとよく解ります。イラストで表示したらもっとよくわかります。つまり、現物らしい表現が、逆に設計者の意図を表すのに災いになっています。

図の下部が今後本来の設計者に望まれる道具なのだろうなと思います。

- →少し反論があります。希望的な気持ちで、設計者のスキルがこうあって欲しいというのは私も アグリーです。一方で、ビューアなら使っているといわれたが、最近の設計者は3次元のデータ がリアリスティックに詳細に正確に確認できるということのメリットに慣れてしまっている。出 てきた結果で判断するというサイクルで設計を考えるという次元になって来ている。こうありた いというのはわかるのですが、お話のものを与えて、今の若い設計者はできるのだろうか、不安 がある。そうありたいけど、出来ない環境になったしまっているのではないでしょうか。
- ←「微細なフィレットをとった形状でないと設計者の本当の意図はわからない」ということ自身を問題としているような設計者がいるのかどうかが重要なのです。この話は設計の課長さんクラスとかと話をしており、このクラスの方はわかるのですが、2~3年の設計者がこの問題意識を持っているかというと持っていないでしょう。
- →気持ちはわかるのですが、出来るのかな。時代を戻せといっているのかな。いや、こうありたいですよ。言い換えると、設計者のスキルをどういうレベルにおきたいのですか。そこが、すごく大事なことです。
- ←初期の CAD のゾーンはこうでした。最近のソリッド CAD は下流のこういうゾーンで使われだして、ここら辺はどんどん使われなくなった。
- じゃ、本当は、ここら辺ってあるでしょう。設計者がワイヤーフレームで気楽に描いていた3次元というゾーンで今はCADがなくなっているわけです。紙になってしまっている。最終的なモデラーが使う詳細な3Dのモデリングはある。そして、昔からもやってなかったゾーンもある。
- →後工程側としても、これはどういう意図でつくられているのかを見たい。そして3Dの世界も入っていて、両方見えて欲しい。
- →森委員は、今の3次元に浸かっている人を上流の方に自然と呼び戻せる CAD にしたいときっと思っているのですね。

←そうです。

→図の下部を「Realistic に」「詳細&正確に」まで囲っておいて欲しい。残しておいて欲しい。

2.1.9 自動車設計の道具の方向性

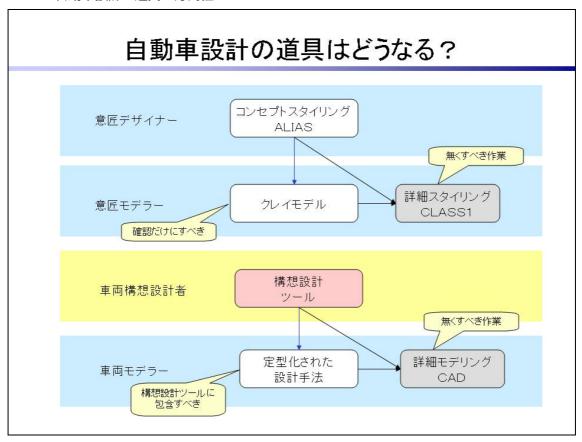


図 I-2-11 自動車設計の道具はどうなる

これが理想だとは思ってないのですが、今後どうなるのかなという一つの仮説です。

上2段の方を見て下さい。いま、意匠のデザインのコンセプトスタイリングで使われているのは、 どこの会社も Alias です。 もう一つは、実物のクレイモデルです。 なのに、いくつかの自動車会 社は詳細のスタイリング CAD を持っています。 ともかく、こういう構図になっています。

本来は、コンセプトスタイリングをやっている Alias が、Class 1 まで出来るような品質の面の接続性とかを制御できれば、そんなことは必要ないわけです。でも、そうすると Alias の使いやすさを犠牲にするわけです。だから今はこういう構図になっている。

車体設計者にこれを当てはめてみると、下2段になります。

詳細モデリング CAD が、今ある CAD です。では、車両の Alias に相当するようなコンセプトを考える道具が一個あってもいいのではないでしょうか。

今、車両のモデラーは、詳細モデリング CAD を使うときに、「定型化されたドキュメント」「定型化された設計手法」を見てやっています。

本来は構想設計の道具があって、それらが「定型化された(設計)手法」もカバーし、そしてその結果、最終的な詳細化した形が出て来ればこしたことはないのですが、もしかすると、上2段の例に対応するかたちで、過渡期はこの3つの構造になるかもしれない。

- →構想設計ツールから定型化された設計の場合は真下に行って、非定型化された設計の場合は詳細モデリング CAD に行くということですか。
- →上から下の青矢印は基本的には何も行かないのでしょう。順序を表しているだけですよね。
- ←そうです方向だけです。上から下に渡るという意味ではありません。情報は斜めに渡ります。
- →その矢印は双方向であってほしい。
- →構想設計ツールで包含すべきと書いているということはどういうことですか。
- ←「(定型化された) 設計手法」が仮にあるのであれば、構想設計ツールでガイダンスをしてくれるべきでしょうというくらいの意味です。
- →無くすべき作業と書いてあることは、構想設計ツールが段々進化して行って、かつうまいイン テリジェンスを持って自動化することが出来たならば、微細なフィレットなんか手数をかけない で自動化するということですか。
- ←そうです。微細のフィレットなんかはボタン1個でやるべきです。
- →過渡期には既存の今のものを繋がなければいけないと言っているのですね。
- ←ボタン1個でやるのは大変ではないか、今、倍の工数がかっているわけですから、ということですが、でも車体のドアの内板パネルで微細なフィレットがいる場所はもうわかっています。いくらで掛ければいいかも大体想定できている。
- →一般的なフィレット問題だと思ったら全然だめ。
- ←そうです。
- →自動化していくことを狙っている?
- ←そのとおりです。
- →私もすごく賛成なのですけど、ここ4~5年前までは設計者がCADを使っていました。しかし、 設計者が作った3Dモデルはものづくりに使えないということで、そこをきちんとしなければい けないという発想のもとに3次元化がすごく叫ばれたのです。それが何故、今になって設計者が CAD使わないとなったのは、誰が悪いと思っていますか。あそこの詳細モデリングにはモデラ ーが介在するわけですよね。だからあそこの価値というか比重がすごく上がっているわけです。
- ←ツールの問題でいうと、フィレットが掛からないということについて、我々も市販のものを扱うことになって、自分達がフィレットをどう掛けようかと思うことが無くなってしまった。CAD ベンダに要望するだけになってしまった。

- →意匠デザイナーと意匠モデラーが違うように、設計者とモデラーは違うという捉えなおしは出来ないですか。
- ←上2段と下2段はすごくよく似ています。まず、上の意匠デザイナーはご存知のとおりです。 上2段の下はクレイモデルを使っていた人たちが詳細スタイリング CAD を使ってモデルを作っ ています。

下2段の場合、「構想設計者」と書いているのは普通の設計者です。下2段の下は試作だったのです、木型製作者です。その人たちが実は CAD のモデラーに変わったわけです。だから、両方とも、ものを作っている人たちがそれを、バーチャルな世界に持ってこようとしているのです。ただ、木型の場合はなくなりましたね。クレイモデルは承認モデルなどが残っています。

- →下2段の上と下2段の下を明確に区切れないですか。
- ←そうクリアに区切れるものではなくて、今、そこの線は組織で切れています。米国の GM とかフォードは、今の型式の車は全部設計会社にいきます。そしてエンジニアといわれる人は全く新しいことをやっている。対象の違いで区切るくらいしかないかなと思います。
- →私は下2段も上2段と同じだと思います。下2段の下の「定型化された設計手法」の代わりに、 クレイモデルに相当する「デジタルデータ」があって、それを皆が使う。全くクレイモデル同じ です。それでいろんなものの確認もするし、それを後から試す。だから同じ構図でよい。
- →それを構想設計に包含すべきか。
- →いや、ならないと思う。
- ←ぼくは包含すべきだと思う。
- →包含すべきというならば、さっき私が言った、両方の意図が1個の CAD になって表現される べきということです。同じことを言っているかもしれないが、ツールに包含すべきというと誤解 されそうですね。

あれが、一つの CAD であること、一つのデータベースで表現を持っていることが重要。

構想と詳細は双方向だと思っています。構想設計者は詳細化されたデータをリアリスティックに見る、利用する。だからプラットフォーム上は一緒でないとだめ。もうそうじゃなければ許されない環境になっている。設計者の意図を見るのは、構想設計者だけではない。詳細モデリングをしている人も見るし、場合によっては後工程の人も見る。これがどういう意図で作っているのかはすごく大事です。だからその意味でもっと拡張されたデジタルデータがあってよい。

だから、「構想設計ツールが包含される」、「に」と「が」の違いです。上に包含されるべきといわれると考えてしまう。

- ←私の意図は構想設計という段階で完ではない、形が決まった。ただ小さな水抜き穴だとか、皺とりのビードは(設計者は)考えなくてモデラーの人にやってもらうかもしれない。同じ人が何処まで細かく定義するかは別にして、最終的には一つのツールの中に入っているべきでしょうということです。
- →ひとつの考え方は、下流のデジタルデータで構想の意図が明解になるようなもの、つまり、シ

ステムとしては今の詳細モデリング CAD にプラスアルファで構想設計ツールを作り込む、もう一つの考え方は、いまは構想設計がない世界がドンドン進化するといまの詳細モデリング機能が進化した形で入ってくると言っているのではないでしょうか。

→4~5 年前は、設計が作ったモデルがモノ作りに繋がらないから問題だった。そこに生産技術、後ろから前に来てモデリングしましょうというのが主流だったはず。それがいつの間にか、もっと上流にと言い出してから、設計者がそういう形で仕事出来なくなった。

本来、詳細モデルが主役で、人とシステムは黒子だったはずなのが、それが今、ここが大きく見えるようになって来た。そういう風に仕事の仕組みが変わってきています。それが道具によって今度、構想まで食いつぶしてきている。

←もう一つ、会議とか決め事をするのが設計で、CAD を使っているのは設計ではないという意味の言葉が結構あります。主任以上のクラスになると CAD を使わない。構想は紙に描かいています。

2.1.10 最後に

CAD などの道具は、高級になっていくと、専用工具になると、使えるシチュエーションはすごく狭くなります。しかし効率は一般の道具に比べて圧倒的に良い。一番望んでいるものは、使える範疇が低レベルの道具に近い幅をもって、そして専用ツールと同じくらいのパフォーマンスがあるものです。

専用システムの塊にすると、フロントドア設計 CAD、リアドア設計 CAD、フロントバンパ設計 CAD となってしまう、これはありえないでしょう。

- →設計力が向上する CAD とか、もっと賢い CAD という視点はどうですか。
- ←そういう視点はありません。私はあまり考えていません。本来の設計者が使える CAD です。

私は微細フィレットを付けは人がやるべきことではないと思っています。

そんな単純な仕事を人間はやるなと、もしそれでベテランの人が余ったら、設計をやってもらえば良い。

自分で線を描いたことがない、形を作ったことがない設計者では、まともなものは設計出来ない のではないでしょうか。

- →例えば、ボタン1つで、フィレットがかかる効果、何ぼ儲かりますよと。
- ←数字は出るでしょうね。
- →フィレットかけるだけで、モデリング工数が倍かかっているから。
- ←ただ、設計者はこれに興味を持たない。また、もしかしたらモデラー集団の会社もお金にならないから興味を持たないかもしれない。
- →設計者は、とりあえず投げてリアリスティックになったものでそこから考えようとなっている。
- ←結局みんな同じ方向に行っている。私は個人的には、やはりその方向はおかしいと思います。



自動車エンジン設計の事例

ヤマハ発動機株式会社 米田 俊彦

図 I-2-12 自動車エンジン設計の事例

今、新しい CAD システムを作ろうとしています。

実際には設計のやり方を少し変えさせたいこともあって、その中の資料を引用しています。

2.2.1 設計とは

(1) 設計シーン

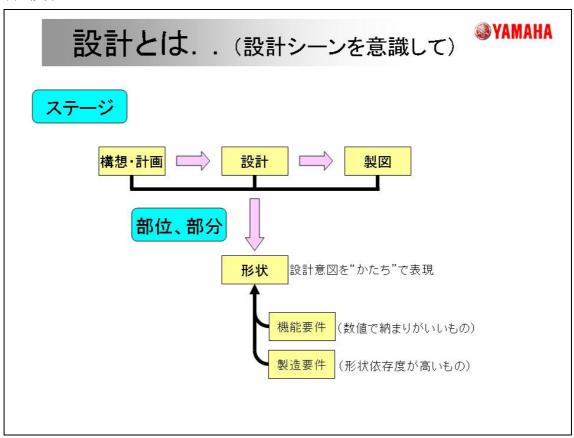


図 I-2-13 設計シーンを意識して

「設計シーン」という言葉を考えてみると「構想・計画→設計→製図」というステージと、ステージの中の1個1個で大きくて「部位」、もう少し小さくて「部分」があると考えられます。 牧野委員は製造寄りの話とすると、私は設計の上流の話をいたします。

設計作業では形状を決めていくわけですが、その中には設計要件が入ってきます。 設計要件には、機能要件と製造要件があります。

機能要件は数字で収まりがいいもの(強度、耐久性、・・・)、製造要件は形が決まらないと、どうしたらよいかというところまで来ない(形状から見ると機能要件が主、製造要件は従)。 この両方を持って形を表現していくのが設計ではないか。

(2) 実例

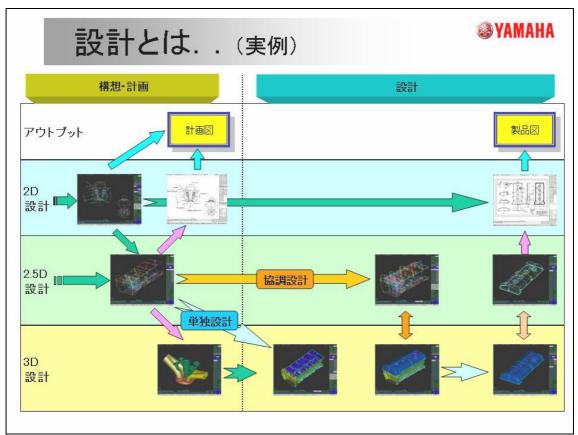


図 I- 2-14 実例

これが我々の仕事のやり方の実例です。

・構想・計画

エンジンは試作の回数がまだまだ沢山あります。一般的に先行試作と量産試作があります。巷でよく言われる期間短縮というのは、後者の部分のことです。

その前の先行的開発には数年費やしています。長くて5年、短くて2~3年が一般的です。

<2D 設計>

自分はヘッド回りの設計が多いので、2次元的にバルブのレイアウト、燃焼室のレイアウト、動 弁系のレイアウトをします。

これらが決まるとある程度の性能予測がつくので、一般的な計画はこういうところから入っていきます。

<2.5D 設計>

これを膨らますため、断面や平面図を必要なところに置いていきます。これを 2.5 次元と我々は 呼びます。

性能要件が非常に高いものは、計画の段階でポートと燃焼室まで3次元を作り込みます。

このあと単独設計に移行する場合は、3D 設計になりますが、大体の場合はここでいう協調設計 $(2.5 \, \% \, \pi)$ \sim 80ます。

計画図が出来上がってくるとこれを図面にして計画図として出します。昔の紙の世界だと1回きって出していたが、今はだらだらと進んで行きます。

• 設計

協調設計をやる場合は 2.5 次元から形にしていきます。これはヘッド、中のキャップ類、ヘッド カバーの例です。

人数は、構想・計画では2~3人、設計では8~10人です。

ここで協調設計がある程度終わると単独設計に行って、細かいところを詰めます。面の情報から 断面などを抜き出し、最終的に製品図にします。

当然この 2.5D 設計と 3D 設計の間は行ったり来たりしながら形を作ります。

ヘッドカバーなどは製造要件としての型構成の可動と固定はわかりますので、この状態で抜き勾配をつけています。

我々も SYS 体制ということで技術と製造は一体になった組織にしており、シリンダヘッドなどは 生産技術の人が横にいますので一緒に仕事して決めていきます。

- →ボディー設計みたいに本来 1 個のものを N 人に分割してやるような協同設計はないのですね。 →サイドパネルは決める要件が多くて一人でやりきれない。
- ←エンジンの場合は部品単位で分かれているか、シリンダヘッド廻り、吸気系... など機能単位で分かれている。ヘッドと吸気系の関係ではフランジがどうあるか程度で相手の部品を見る量も少ない。
- ーボディーはどちらかというと部位ではないかな。

(3) 工程別要件

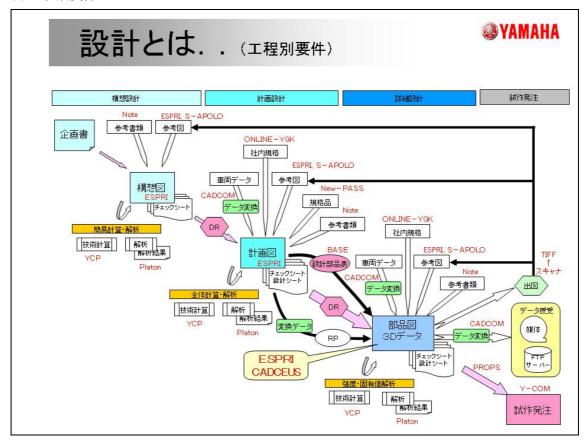


図 I- 2-15 工程別要件

牧野委員が話されたチェックシートは我々も持っています。企画からはじまり最終的な部品図に くるのですが、ここではチェックシートを何回も同じようなことをやる、ただ出している内容が 段々高度になってくる。

我々のような内製のオートバイは自動車開発ほど厳しくないので、設計者が自ら形状を作ってい くやりかたをしている。

赤字のところは我々のシステムです。最終的には ESPRI と CADCEUS を使ってやっています。 絵に描くとシステマチックだが、全部きれいに繋がっていない、切れています。 例えば、「社内規格」は見るだけです。ポンと置いたら何とかとはなっていません。

- →「チェックシート」は本当の紙ですか。
- ←紙と言うよりは、一応、電子データです。基本的には紙です。

(4) 設計ステージと成熟度

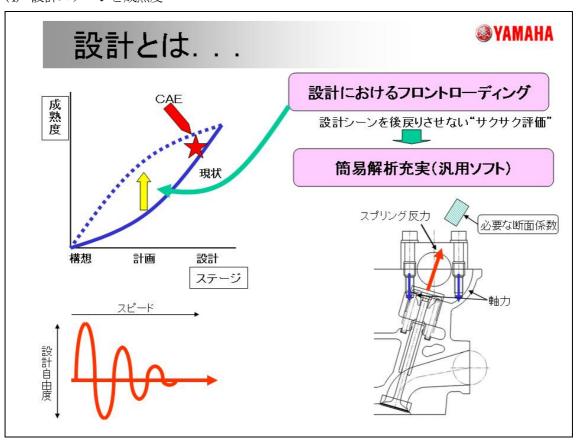


図 I-2-16 ステージと成熟度

別の側面で見てみると、ステージと成熟度の関係はこのようになっている。

開発全体ではフロントローディングは言われて久しいので、できるだけそっち方向に近づいているのですが、設計の工程だけ見たときに、まだまだ最後に決まるところがあります。 ひとつは CAE です。3D 形状が出来上がったあとにメッシュをドンと切ってやる話があります。 ここで決まることが非常に大きい。

一番かかるのが、エンジン全体のパワートレーンのベンディングとか振動のところです。 そうすると全データのメッシュ切りが必要で、1ヶ月かかる。設計が終わって1ヶ月待たないと いけない。CAE の結果が出る頃にはモノは出来ていて、評価試験が出来ている状態です。このた め、次の試作のときにそれを織り込むパターンが多い。前回開発機種はこうだった、という参考 データとなっている。ということで、くるくる回っている風にはなっていない。

この辺は設計の自由度が非常に大きいところですが、ターニングポイントで解析をちょこちょこ

入れないと決められないのが実際です。それを、たくさんやってうまくやると、スピードが上がる。

グラフの成熟度を引き上げるために、設計におけるフロントローディングが必要です。

設計シーンという言葉を使うと、部位を決めた段階で1個1個きちんと評価していかなければいけない。それも時間がかかったのでは意味がない。

昨年のサクサク評価と設計シーンという言葉を合わせると、こんな風にいえると思います。

CADデータで数値を扱っていると、モノの出来が良くなったように勘違いする。

動弁系で、カムで押されるとスプリング反力がでて、カムキャップはボルトの軸力で止まっているので、ここに応力がでて、必要な断面係数が決まります。この辺、我々はどのくらいの断面係数が欲しいか分かるが、こういうことを新入社員はやらない。後になって、軸力が足らないので変更しようとするとヘッドカバーが干渉するなど、以外とつまらない繰り返しが多い。

簡易的なところが、もうちょうとピシピシ出来るとよいと思います。

手計算でやったのものは当然ありますが、紙の形です。例えば、CAD データとそういうデータが一体となっているともっとやり易いのですけど、そうはなっていない。

- →簡易解析というのは FEM までいかないのですね。
- ←そういうところまでいかない。例えば、「ボルトはこれでいいですね。横幅はこれだけ欲しいで すね。」という話の方が大きくて、この段階ではたいそうなシステムは要らないのではないか。
- →ああいうふうに聞くと簡単なのですが、何がパラメータで、何に影響が行くかの方が難しい。 いろいろなケースがあって、起きてしまうと、なぜそんなことに気が付かないのということが多い。
- ←何処に注意力が行くかの問題が大きい。ある人はこういう見方をするかもしれないし、別の人は例えばこういうところを見る。その辺が汎用ソフトでないと見られない。専用ソフトだとそれしかやらないという問題が非常に大きいと思う。梁の計算で済むようなことをきちんとやらしてくれれば、もっと見るべきところが一杯あるはずです。

今は立派な専用ソフトを作るので、後工程にいかないと評価ができない。

最初の工程でやるべきことは、「この幅がわかると、ヘッドカバーにも危害を加えないし、次のステップに進める」ということが分かればよいわけで、8割がた当たっていればOKという感覚です。

- →こういう簡易的な確認を全てやるということですか。
- ←全てやるとは云いません。
- →ものすごく簡易にして自動化して、全部通すようにできれば一番いいでしょうけど。

←例えば、この横方向の断面係数を上げることになったとき、ネジサイズを上げる人もいるし、ネジサイズをキープして2本で止めるという発想も出てくるはず。ボルト2本で何とか止めたいというのと、小型化するから4本でもいいと考えるのでは、設計方式が全然違う。その辺を全部評価しなさいということは大事なのですが、やった結果がどういう結果で、どう決着したかを残したいという思いがある。

→ だからテンプレート設計ではないということですね。 ← そうです。

2.2.2 設計支援機能

(1) 設計支援機能の課題

設計支援機能の課題



システム要件	機能要件
①初期段階で容易に3Dデータが作成できること 3D化支援	◎ 計画図と連動した3Dデータが作成可能なこと◎ 3Dデータを効率的に作成できるモデリング機能があること◎ 製品形状に特化した3Dデータを生成できること◎ 手順に捉われない3Dデータの創成、変更と関連データの連想性があること
②デジタルで機能検証が行えること デジタル検証支援	◎ 設計途中の簡易モデルで解析ができること◎ 評価の妥当性を支援できること◎ 評価結果、評価手法を蓄積・実用できること◎ 他システムと連携できること
③データの変更が後工程まで連携できること データ連携支援	○ 構想から製造までデータが連想できること○ 関連データの更新を検知できること
④ノウハウの蓄積し設計を支援できること 設計意図支援	● デジタル検証により、設計者が容易に試行錯誤できること● データ変更が協調設計・後工程で連携ができること● 設計手法・評価手法を、システムに蓄積・活用できること

設計シーン毎に3D化・簡易解析を実施し、設計意図の明確化を図り、 システムに蓄積・活用することが望まれる

図 I-2-17 設計支援機能の課題

新しい CAD を作る上でやらなければいけないことは 3D 化ということです。3D 化支援をきちんとやらなければいけない。次のデジタル検証支援もきちんとやらなければいけない。それからデータ連携支援、設計意図支援です。

設計意図支援が一番難しいが、「残す」ということがキーワードです。設計の手法とか評価の手法をシステムに蓄積して活用する。結果だけを残すのではなく、何と比べたとかどのように判断したかを残せるかどうかです。それをやるためには、データをきちんと残さなければいけないので、上から下にだんだん下りてくるということです。

最終的には設計シーン毎に何を決めるかです。先程説明したスパイラルのところのターゲットで、 やるべきことをきちんとやりましょうということをここでは言っています。

→腕のいい設計者とそうでない人で違いがあるので、「残す」ことが基本にあるが、「残したこと」 をブラッシュアップ (洗練) していくことが必要だと思うが、どう考えますか。 ←クライテリアは守らないといけないが、設計の個性を残さないといけないと思う。ブラッシュアップと個性を残すことを、矛盾するというか、バランスをとらないといけない。

ある人の仕事の仕方が正しいというのは、判断をするという行為が正しいのであって、選んだ形 や数値は正しくないかもしれない。そういうところをどの様に残して、どの様に磨いていくかは 非常に難しい。ただそこまで行く前に、まずは蓄えないといけない。

→先生とのお話で蓄積のやり方よりも活用するやりかたの方が大事ということが耳に残っている。 まさにそうなっている。

→主は形ではなく、設計条件や設計判断と思っている。設計条件や設計判断がマスターデータで、 むしろこれに形状を貼り付けなければいけないと思っているのですが、その辺はどう思われます

←そこは非常に難しいですね。いい案が無いです。

→でも設計側からみるとそうなのでしょう? 形状に属性がべたべた入っているのはナンセンス なのでしょう?

←我々が判断していること、こうしたいということと実際の形の間を CAD で翻訳しないといけない。その翻訳の手法が今の CAD システムはお粗末な気がする。「肉厚」を「オフセット」、「除変」をまた別の言葉に置き換えなければならない。設計の用語が使えない。そこを変えたい、変えて欲しい。

→もうひとつ、CAD のコマンドは設計者の 1 つの行為と対応するようには出来ていない。フィーチャも設計者の一つの認識に対応するようには出来ていない。小さ過ぎる。 CAD のコマンドを設計者用に近づけようとすると、設計シーンに対応するくくりがないとつらい。

←我々はスケッチという言葉とビューという世界を使っています。

スケッチというのは、ある平面に書かれた線の集まりだと思います。それは、それから立体が立ち上がっていく基本だと思っています。

ワークビューが1スケッチで出来ていて、1立体が出来上がるそれが一つのものを決めるための 判断の一つだと思っています。

そこで、スケッチが変われば当然立体が変わる。その単位でモノを判断していく。その中の線 1本ではなくて、それ全体で意味のある形だと思う。それがボスだとか、そういう名前になっていくと思う。

このボスはいいか悪いか、そこで解析が必要か、必要じゃないかをそういう単位で判断をきちん としていく。

解析といっても立派な解析ではなくて、例えば簡易解析でもいいし、ネジの径が決まればボスの 径、肉厚は決まる。そんな簡単なところから始まって行くような気がする。締めるボルトの種類 によって径は変わる。フランジ付きボルト、ワッシャボルトと六角穴付きボルトでは必要とする ボスの面圧が違うので径が変わる。そういう簡単なチェックから全て始まっていくと思う。 ベテラン設計者が、今回ここはこういうネジを締めたいなと思ったら、必然的にM6 だと ϕ 12 の ボスです、フランジボルトだと ϕ 14 のボスですと頭のなかに入っているのでよいのですが、そう でないときは、データベースを拾ってきてやらなければいけない。

→それは、どれくらいスマートに出来るかは別として、技術的に出来ないことではない。 ←そういうデータベースはあるのですが、CADとは繋がっていない。

(2) 仕事の流れと設計支援機能

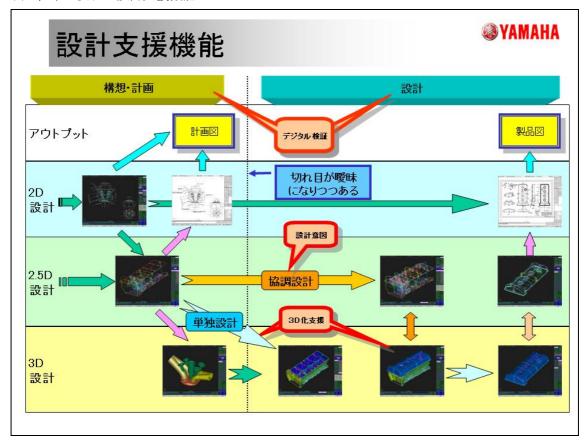


図 I-2-18 設計支援機能

さっきの背景を入れていくと、「デジタル検証」は全てのところに及びます。

「設計意図」は、周りの人に、どういうふうに形として現れている部分をどうして欲しいかを伝える、という意味で協調設計に関連付けた。これ以降のモノを作る世界にいくと、「設計意図」は 非常に比重が上がっていくのですが、現時点ではこの協調設計の比重が非常に大きいと思います。

(3) デジタル検証支援

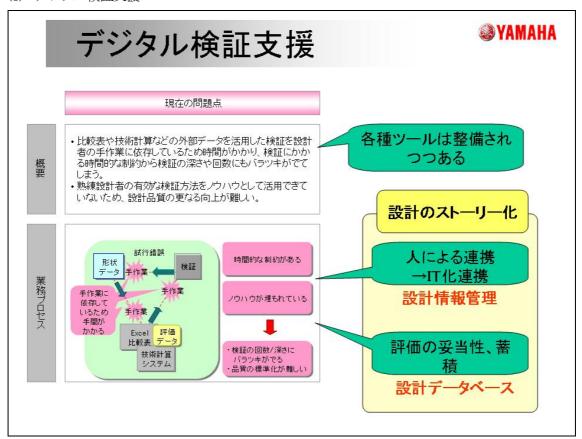


図 I-2-19 デジタル検証支援

ツールはいろいろあります。

ただ、今は全て手作業です。形状を出して、Excel を作って、評価をして、みんな手作業になっている。評価の妥当性を蓄積して、データベースを作って、それを使えるようにして行きたい。

そうすることによって「設計のストーリー」を作っていける。そのストーリーには、さきほど牧 野委員がいわれたように、妥当性だとか、どこをチェックしたい、どこをチェックした、が大事 です。その結果がどうのこうのは二の次です。

熟練設計者がどこを見て仕事をしているかを早く抜き出して、ストーリー化していくことが大事です。これはややもすると類似設計とかになってしまうので困るのですが、類似設計を強く訴えたいのではありません。

チェックする項目というのは、設計の方針で変わっていくもので、自分が必要としているところをうまく抜き出してくるようなかたちの連想性がないと困る。そこが一番の肝になるところで、難しいところだと思います。

- →CAD、CAE はソリューションに走っていますね。ガイドとか、本当の意味のエイドになっていない。
- →「評価の妥当性、蓄積」の製品・部品への依存性はどんな感じだと思われますか。自動車エンジンのどういう単位毎に変わっていきますか、それともエンジン全体でこういうことをいうのですか。
- ←実際には部品で分かれています。自分が持っている資料をみると、例えば、性能という見方をすると、性能が吸気系の径から燃焼室まで揃っている。性能を見た後に動弁系の何々を見るとだんだん細かくなっていく。個人資料はかなりの人が持っている。そこの数字を変えると、次のエンジンは何馬力出るという表も持っている。ボアストロークが決まると大体性能が決まるという資料になっている。大きいものから、小さなものではバルブ1本の重量であり強度だとかの資料が全部揃っている。
- →ということは、一般化はかなり難しいということですか。
- ←一般化は難しいと思います。何もない世界から作っていく世界と、形がある程度見えたなかで やっていくものでは少し違うと思います。
- →エンジンでも、最初の新規エンジンとか、2.5 リッタのエンジンを 3 リッタにするとか、A車に適用したものをB車に適用するとかありますよね。後者の適合設計は定常化していくのではないでしょうか。
- ←それはそうです。レシプロ系の基本設計の部分と車に依存するいわゆる搭載設計に近い吸排気 系の部分は考え方が全然違います。
- →関わっている人数比とか、工数はどちらかというと適合設計の方がはるかに多いのではないで しょうか。
- ←そういう意味ではそうです。先行試作と量産試作があるので、先行試作はエンジンの基本の部分を作っていきましょうというところが大きい。量産試作では車体データがきちっとしてくるので搭載性を気にします。ステージ毎に配分されるような部分が大きい。
- 先行試作では思い切ったことができるが、量産試作では今から車体を変えるわけにはいかないので、自分の性能を落とさないようどこまで頑張れるかになる。
- →構想段階に焦点を当てると、いい道具を提供しようとしても、ありがたい人の数は少ない。下 流の方は、ありがたい人の数は多い。
- →それと技術のブレークスルーは別の話ですね。
- ←ひっくるめて「設計」という言い方を変えていかないといけない。基本的に開発設計と生産設計は全然違うと思う。先行試作は、遠くに生産を睨みながらなので、結構好きなことができる。 生産を伴った開発設計者との人数比は1対10です。
- 試作の段階では生産設計者に構ってもらえない。手法も違う。先行開発は木型であり、製品開発は金型です。

(4) エンジン検証:動きとグラフ

CAD データと Excel のグラフをうまく連動してやらないといけない。これも、表を作るのに意味があるのではなく、どこを、何を見たいかなのです。

例えば、ピストンとバルブの最小隙間がどうなっているかとかを見る。また、3次元の動きと専用ソフトでないと出てこないガス圧と CAD データからわかる慣性トルクから合成トルクを出したい。専用部分と汎用部分をうまく組み合わせてやりたい。

見たいものが簡単に見られることが必要です。この辺だとすぐ専用プログラムになってしまう。 構想と設計で使い方も変わる。同じ排気量で、ボアとストロークが変われば当然グラフは変わる ので、構想段階ではそれを見なければいけない。設計の終りに近づけば、これで間違っていない かをチェックしなければいけない。

→バルブクリアランスにしても、ガス圧にしても、本当は、見たいのは 6000 回転の時のもので はありませんか?

←そうです。当然そうなると、複雑になります。こんなにきれいに廻っているわけではなく、回 転変動がどんどん出てきます。

そういうところも、本当は織り込んでこなくてはいけなくて、こういうの(最初に提示したの) を見ることによって、だんだんそういうことがわかってくるのですよ。だから次はそういうのを 見たくなるのですよ。

実を言うと専用プログラムだと、そのような発想がなかなか出てこない。1 個やると次の話題に 膨らんでいくようなものが必要なのです。

→だから、ないといっているのですよ。

2.2.3 構想設計の1シーン

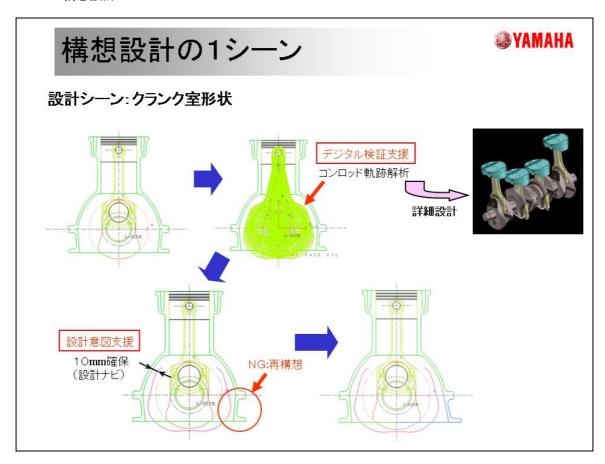


図 I-2-20 構想設計の1シーン

実際の設計はどうやってやっているのかという話になるわけですが、これが構想設計の一つのシーンと捉えていただきたいと思います。

ボア×ストロークが決まって、ある程度コンロッドの長さが決まりましたということになると、クランク室の形状を概略決めなければいけない。このときに何を注意しなければいけないかということになる。

このときにコンロッドをグルグルグルと動かさなければいけないのです。廻った形がこれです。 専用プログラム(コンロッドの軌跡解析)の軌跡から、最外郭線が目で見てわかる状態になる。 もし専用プログラムがないと自分で置いていくやり方になる。

「最外郭線(包絡線)とクランク室の壁の間は10mm欲しい。」という設計ナビがあります。 設計ナビは、紙として、ここに隙間があるとロス馬力はどれ位かという数字があります。 CADと連携して、この隙間を測らなければいけないとか、クライテリアがいくつですと出てくるわけではありません。 あくまでも紙とかデータがあることを知っている人が見て、ここはこれだけなければならないという形になる。

そうすると、ここで初めて、外郭形状の○印の部分は短いから NG であることがわかる。 次に半分だけ作り直します。

この形になった時にモノが作れるか作れないかの判断をする。型をどっちに抜くかの話になる。 わかる人はスーとやるが、わからないと適当にやってしまう部分です。

このあと詳細設計に入っていくと、図右上のソリッドになって、動きを伴った解析をやるようになる。

このクランク室を決める段階では、図のような流れに沿うことになります。大事なのは、ここの 設計意図をどういう風に CAD の世界と結びつけるかです。それから NG ですよという判断をし て再構想をしなければいけない。ここで判断するのが大事です。

このあと再設計したものを出てきたいろいろな要件、この場合は隙間ですが、を再評価することになる。

これが、クランク室の形状を決めるときのルートの一例です。

- →1 人でやっているのですか。
- ←そうです。
- →このとき隙間が 9mm だったとします。どうして 10mm 確保しなければいけないのかの理由を検索できますか。
- ←検索までは出来ないですね。
- →もしかすると、10mm を目指せという意図で、10mm 均一を要求しているのではなく、狭いところは 9mm でもよいということもあるかなと思ったのですが。
- ←実際には 9mm でも問題ないのです、本当はここの流速の話です。ボアストロークに対してここの周速をいくつにしたいか。周速をいくつにすると、どれだけ壁が離れてないとオイルが連れ廻るかというのが本当は正しい設計です。

10mm という数字を使った方が楽なものですから。値に置き換えています。

この辺は、検証が進んでいくと、周速に対して、壁が何 mm 離れたところの圧力がどうなるから 9mm でもいいのか 12mm 欲しいのかが初めてわかる。だけど、当たりをつけるという面では、これで間違いがない。

ここまでくると、ここの細かい形まで決まるので、そうすると初めて確定する。

ですから、厚み方向の先端を尖らせて、周速の早い部分を少なくする方向に進みつつある。

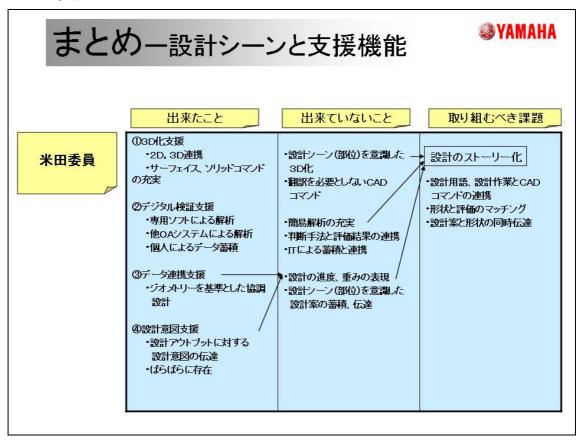
こういう流れの中で段々煮詰めて行くというところが、そういうところになるのではないでしょうか。

- →狭くすると、抵抗が大きくなるということですか。
- ←狭くなると、入っているオイルが一緒にグルグル廻りだすのです。そうすると、こういうところに当たって、その分だけロス馬力になります。これを詰めるかわりに、別の室を設けるというやり方もあります。
- →オイルはどの辺まで入っているのですか。
- ←オイルは、ちなみに、これから 20~30mm 下に入っています。ピストンが焼き付かないように、ここの壁に向かってオイルを掻き揚げています。掻き揚げたオイルが、ここを伝わって落ちてきたときに、そこで当たって抵抗になります。
- →本当は当たらないように、壁に沿って落ちればいいわけですね。
- ←レース車の場合は、別の壁をもう1枚立てて、当たらない方向にオイルを持っていくことが常 套手段です。量産品の場合は、これがダイキャストで出来ないということになると、実現出来な いことになります。

当然ヘッドに上がった油はこのクランク室に落ちてきます。どちらかというと、ヘッドから落ちてくる油の方が膨大です。それを如何に抵抗無くオイルパンに戻すかです。うまく早く戻るということは、循環する油の量が少なくて済むことになります。

- →その流体の解析はきっちり出来るものですか、そういうプログラムはありますか。
- ←私は見たことがないです。水(ウォータポンプからラジエターまでの流れの解析)はやりますが、オイルの方はやれないですね。
- →自由境界問題だからですね。
- ←このように今は数字で言っていますが、本当は、ここの流速がいくつでというのが本来の設計 の意図なのです。そういうところが沢山あります。
- →シミュレーションをやる上の一番のガンはなんですか。
- ←シリンダブロックのモデリングに何日かかるか、そこからシミュレーションをやっている間に モノが出来てしまう。最終的にはそこに行ってしまう。
- →10mm を変えようとすると大変な話になるわけですね。
- ←そうです。もっともらしい理由が必要になります。

2.2.4 まとめ



図I-2-21 まとめ

通常やるべきことはそこそこ出来てきました。

出来ていないことは、CAD を使うというとどうしても翻訳しないといけないところがネックになっていると思います。

CADCEUS を使っていて履歴にひっかかり、履歴のブロック化などをやってもらっいました。ブロック化してもらうことによって、一つはどこを決めているかが明確になった。もう少しいくと、設計用語、設計作業とやりたいことがもう少しマッチングできるのではないか。

- →履歴の情報と設計シーンはすごく近い概念でしょう。 1 ブロックに相当するのが 1 シーンなのでしょう。
- ←そうですね。スケッチを書いて、立体を作って、集合演算をする手前のところまでが1つのシ ーンと思います。

やって欲しいのはそういう部分と、それから、

形状と評価はグルグル回っていないと、設計のフロントローディングが出来ないのでやらなければいけない。会社としてもやらないといけない。

最後に協調設計を考えたときに、形状は目に見えてわかるからよいが、設計案は何を評価してど ういうふうに決めたか、さっきの「ストーリー」という言葉です。

決めたジオメトリだけでなく、もう少し奥も伝わったらいいですね。ここが変更可能なのかどう なのかすらわからない。昨年でいうと、「血が通っている、通っていない」という話です。

2.3 構想設計を支援する IT システム像

構想設計を支援するITシステム像

株式会社アイヴィス 大高 哲彦

図 I-2-22 構想設計を支援する IT システム像

事務局が作ってくれた議事録に沿って、一通り読んでみました。それを受けて、一案としてはこんな感じですかね、というものをパワーポイントで作ってみました。 ただし、十分検討できているわけではなく、未だイメージレベルであることを了解願いたい。

2.3.1 話に沿った要件

話に沿った要件

- (1)製品開発ステージ vs 設計シーン(部位、部分)
- (2)協調設計
- (3)設計チェック
- (4)社内規格等の扱い
- (5)CAEが回らない
- (6)設計評価の記録
- (7)設計機能4課題
- (8)デジタル検証支援
- (9)設計シーンと支援機能

図 I-2-23 話に沿った要件

米田委員のお話に沿って、言われたことを少し IT 風の要件にするとこういうことですね、という確認のためです。

(1)製品開発ステージ vs 設計シーン

最終形状に至る途中の設計状態に対応する形をどこまで出すかは検討を要する課題だが、何故こういう形にしたのかが、今の CAD ではわからないことは致命的だと思われる。 わからない限り、どう直したらいいのかわからないわけです。

(2)協調設計

米田委員のお話は自動車エンジン設計が中心だったので、あまり協調設計についての厳しい要件が出ているとは思っていない。私も途中で質問したとおり、ボディー設計などの方が切実なニーズがあると思われます。ボディー設計では数学的に1個のものを強引にN個に分けて、開発期間を短くするためにヨーイドンで並行設計が実施されます。Nグループの設計結果を集めてみたら隙があるは、重複はあるはでは成り立たない。

この問題を何とか解決したいと言われて、関連する仕組みを作ったことがあります。

つまり、自分の設計グループの周囲のグループの設計に関し進捗を確認したいものを自由に設計者が選択でき自分の画面に表示できる。周辺の設計結果の自画面での更新モードはリアルタイムだったり、ユーザが指定したときだったり、いくつかモードがあるのですが、隣の設計者の設計の完成度やどう攻めてきたかがわかる仕組みです。

そうすると、最初はここをやれと言われていたが、実は隣の設計者側から攻めた方がいい場合、 更新権交代という機能があって、この部分はあなたの方で攻めて、というのが自由に出来るよう になります。

今の CAD は、昔から議論されているように、1 人の道具でしかない。N 人でモノ作っているのに、1 人の道具でしかないのは、基本的には欠陥なのですが、それを何とかアドホックだけど、変えようという試みです。

数学的に1個のモノを強引にN個に分けたとき、N個の整合性をとりながら、皆が出来たと言ったときに本当にできていることを狙ったわけです。

その中に赤入れ機能もあって、NHKの何とかNowで、遠隔地の協同設計もここまできたということで放送もされました。アメリカと日本の某社のエンジニアが協同設計対象物の映っている画面を見ながら、ここをもう少し角にして、丸くしてなどの注釈も画面に書き込め、瞬時に相手に伝わる仕組みです。マスターデータは一箇所だけにあって、そのデータを元に自由に話せるという環境・システムを作ったことになる。私はそこそこ出来がいいと思いましたが、それでも実務に展開できませんでした。

今 CAD ベンダが一般的に出している協調設計は上記のような問題に対しては使い物にならない と思っています。その実験システムは、かなり進んでいたはずが、それでも現場が納得しなかっ たということは、協調設計は難しいなあ、という印象が強くあります。

パラメトリックについては散々やってきました。

恐らくエンジン設計の世界ではパラメトリックがもう少し良くなれば、結構協調設計に使えるのではないか、という気がします。

質問としては、現状のパラメトリック機能中心の協調設計でいいですか?ということ。現状のパラメトリック技術は数学的基盤も弱く、且つ大変固い(柔軟性の無い)機能なので、その活用で設計がうまくいくかについて伺いたい。

(3)設計チェック

何回も米田委員からも牧野委員からも出ている、効率的なチェックの仕組みです。

CAD というよりは広く IT システムとして、効率的チェックを支える仕組みとしてどんなことが考えられるか掘り下げた検討が必要です。

(4)社内規格等の扱い

米田委員らしい言い方ですが、「ポンと置いたら何とかなって欲しい」。ポンと置いたら何とかなるということは、良い言い方なのですけども、ここに書いたように、製品モデルと統合された規格の類というのは、どういう持ち方をするのが適切なのか?

というのは相手が非常に多い、大きい、先ほどのような生産要件の場合は何千件、何万件を扱う ので、どうやってインプリメントするのか、どうやって人に知らせるのかがあります。

先ほど松木委員が発言されたように、変更要件を充足するように自動変更するのは、ある部分は 可能だと思うが、全体的には可能とは言えませんので、どういう柔らかい仕組みとして作るのか なと、これは、結構大きな課題と思います。

(5)CAE が回らない

これも非常に端的な言い方ですね、お話されたことは良くわかります。

牧野委員が途中で「簡易解析」という言葉は非常にリスキーなので注意して使わなければいけないといわれたが、お話されていることは、設計のある段階(不完全な段階)で、自分が設計評価の観点でやりたいことが速やかに、適切な精度で計算/確認できて、特にシーン単位で確認したことがきちんと残せることなのでしょう。

解析は、あるときは簡単な数式計算かもしれないし、あるいは過去の実験結果を参照すれば良いかもしれないし、あるいは重長な FEM 計算が必要な場合もあろうし、いろいろなパターンがあります。

次に、何を変えるとどこに影響が出るかが正確に把握できることと話されたのですが、これはか なり難しいですね。

例えば、パラメトリックな情報を経由して、関係しているデータや影響は今でもわかるが、設計者がここ変えたらどこに影響出るかはそんな範囲のはずはない。つまり、今のパラメトリック技術中心では、位相的にあまり異ならない範囲でしかないので、実現はかなり苦しい。

(6)設計評価の記録

米田委員が何回もお話されていたが、どう評価したか、どう判断したか、評価と判断がきちんと 残るような仕組みにしないと、二重チェックの問題とかいろいろ出てくる。

それをテンプレート設計と勘違いしないで下さいと、話されました。

(7)設計支援機能4課題

・「設計評価の記録とそれを活用できる仕組み」

私が記録した情報の洗練はどうするのですかと質問したら、それは時期尚早だろうということで

した。まずは残すことから実現するのでしょう。

・「設計者が翻訳しないで設計者用語で使える CAD」

これは、本来 Feature Base CAD が狙ったものはそのはずだったのですが、今の Feature Base CAD は設計者や生産技術者の意図と形状の関係が極めて不十分なので駄目でしょう。

・「簡易解析を支えるデータベースどうするの」、「設計意図は、本来は製造で比重が上がるが、今 のところでは協調設計支援の比重が大きいですね」という話がありました。

(8)デジタル検証支援

設計のストーリー化、つまり評価の妥当性を蓄積するだけではなくうまく使える仕組み。特に熟練者の観点で新規設計においても、手順のストーリー化にチャレンジしたいということを言われた。

それから、何千件もあるチェック項目の相関。必要なところだけ抜き出せる連想性、これは牧野 委員もお話されたのではないか。毎回数千件もチェックするのは能が無い訳で、どこを変えたか らどこをチェックすべきか、システムが支援できるようにして欲しいと言われた。

それから、1 個やると次の話題が膨らんでいくような、つまり、チェックすべき項目が順次想起できるシステムが理想的なシステムということでした。

(9)設計シーンと支援機能

CADCEUS の履歴のブロック化で設計者が注目している部位への対応機能が以前よりは少し良くなった。もう一段進めて設計用語、設計作業とやりたいことのマッチングを進めたいという話でした。

設計シーンと CAD コマンドは、恐らく相当将来長い間は1対1にならないでしょう。ではどこまでならマッチングできるのかが我々の課題かなと思います。

去年の検討の結果を受けて、設計と評価はぐるぐる回っていないと設計のフロントローディングにならない。つまり、シーンごとにサクサク評価できる、設計シーン単位の手戻りが無い仕組みが重要だと話された。

話の中で、あまり強調されなかったのですが、シーンの構造はフラットなシーンが並んでいるはずはきっと無い。シーンの構造や相関の検討が必要です。構造に関しては、設計'部位'、それより小さな'部分'に対応する自然な構造が考えられる。ある設計シーンは隣接する別な設計シー

ンが煮つまらないと決まらないということは常識的にあるでしょうから、設計シーン間の相関や依存関係が必要である。複数のシーンにまたがる設計上の基本的な考えや制約を記述することも必要なので、最上位のシーンにそれを記述するか、または'設計戦略'とでも称する上位概念を設けて記述するほうが自然かもしれない。以上のようにシーンのデータモデルについては種々検討すべきことがありそうです。

協調設計で形状は目に見えるから良いが、設計案は何を評価してどう決めたかが分かりたい、ど こが変更可能かも現状は分からない、ということです。

去年一部話した「血の通った」、つまり設計のセマンティックス(意味)を内包したデータ表現に通じることですが、米田委員がやりたい世界はすごく深いので、これさえ出来ればいいのではない。

次に要件一覧 18 個を並べました。

要件一覧

- <1>形状からそれら(設計意図)が読み取れること。また形状はシーン単位に決められること。
- <2>バラメ技術中心の協調設計支援の仕掛けで十分か?
- <3>効率的な設計チェックの仕掛けは?
- <4>置いたら何とかなる、つまり、製品モデルとIntegrateされた規格の類の扱い検討要。
- <5>簡易設計(概ねFEMまで要求しない)を充実させ設計サイクルで回せる/最終CA E軽減要検討。シーン単位でちゃんと、高速(ご評価要(設計シーンーサクサク評価)。
- <6>目的にあった(8割がた当たっていればOKというような)汎用な簡易解析システムのあり方、変更の影響の表現法と影響部のみ高速変更のメカニズム
- <7>何種類かの異なる設計案を評価し、それぞれの結果がどうだったか、どういう判断である設計案にしたかという判断を残したい (Template設計との違い)。
- <8>設計評価の記録とそれを活用できる仕組みが先ず重要。記録した情報の洗練は次の課題か?
- <9>設計者が翻訳なしに設計者の用語で使えるCAD(本来のFeature Base CAD)の実現
- <10>簡易解析を支えるデータベース(ex. ネジ径ーボス径、肉厚の関係)
- <11>設計意図は本来は製造で比重が上がるが現時点では協調設計支援の比重大、デジタル支援は全般にまたがる。
- <12>設計のストーリー化(評価の妥当性蓄積/活用、IT化連携)。つまり熟練設計者の着眼のストーリー化。
- <13>チェック項目の相関、必要なところを抜き出せる連想性。
- <14>1個やると次の話題の膨らんでいくような、つまり、チェックすべき事項が順次想起できるあるいはシステムが支援するしくみ。
- <15>設計シーンとCADコマンドが1-1になる世界は未だ時期尚早。ならばどこまでマッチングが進められるか検討要。
- <16>設計シーンごとにサクサク評価でき設計シーンの手戻りが無い仕組みということ。但し設計シーンの相関はどうするのか?ある設計シーンは他の設計シーンが煮つまらないと決まらないということへの対応?</p>
- <17>昨年の'血の通った'つまり設計のセマンティックス(意味)を内包したデータ表現に通じる。
- < 18>設計から生産準備までの物作りの流れ・考え方に沿って仕組みとしてどうサポートするかが課題。そういう世界を定型設計の世界だけでなく新規設計の世界にも使える柔らかい仕掛けで作りたいーータフな課題である。現状はナレッジとCADさえくっついていない。ある程度の塊に設計の意味付けができていれば、それをどう組み合わせて使うかに設計者の差が出る。

図 I- 2-24 要件一覧

要件を集約したCADのイメージ

- 設計者がこめた意図や意味がデータに反映されていて他者にも理解可能であり、システムもそれに応じた振る舞いをするCAD
 - 設計シーン単位のデータ操作、機能体系
 - 設計者用語で使えるCAD
- 設計シーン単位のデータ構造を持ち、シーン単位の設計評価が可能であると共に評価結果と判断が蓄積・活用できるCAD
 - 効率的設計チェックの仕組み(チェック項目の相関、必要項目抽出など)
 - 変更可能部分、変更の影響がわかる仕組み
 - 高速・効率的評価を支えるデータベース
 - 設計のストーリー化
 - モノ作りの流れに沿った新規設計でも使える柔軟な意味付けされたデータ 群
- N人が共有できるCADによる抜本的協調設計支援
 - 設計部署内、設計部所間、設計から生産準備/生産現場への密連携を実現する協調製品開発環境の実現

図 I- 2-25 要件を集約した CAD のイメージ

それで米田委員がおやりになりたいことはこうなのではないかというのを書いてみました。

- (1) 設計者がこめた意図や意味がデータに反映されていて他者にも理解可能であり、システムも それに応じた振る舞いをする CAD
- (2) 設計シーン単位のデータ構造を持ち、シーン単位の設計評価が可能であると共に評価結果と 判断が蓄積・活用できる CAD

つまり、

- ・効率的設計チェックの仕組み
- ・変更が可能な部分、変更の影響がわかる仕組み
- ・高速・効率的評価を支えるデータベース
- 設計のストーリー化
- ・モノづくりの流れに沿った新規設計でも使える柔軟な意味付けされたデータ群 こういうものが、いるのでしょう。

(3) N人が共有できる CAD による抜本的協調設計支援

米田委員は、ここまではお話されたのですが、牧野委員の話も入れると、設計部署内、設計部署間、設計から生産準備/生産現場への密連携を実現するような協調製品開発環境は、この会議で議論できるかどうか、わかりませんが、明らかに大きな課題だと思う。

大きく言えば、(1)、(2)、(3)はディスクリートな問題ではない。特に(1)と(2)は非常に近い問題です。主として米田委員がお話されたのは、(1)と(2)を合わせたような世界がどう作れるのということでしょう。

2.3.3 実現性に関する考察

それではということで、イメージだけ少し書いてみました。私が思っていることの一部があまり 吟味しない形で出ています。

(1) データ表現

データ表現

従来の最終形状(形状モデル、幾何/位相/特徴)やパラメトリック情報が中核で表に見えるデータ表現は止め、設計シーン群とその相関が中核のデータ表現に変更。設計シーンを束ねる概念(ex.設計戦略)も重要。

- <設計シーンには>
- 一特に重視する設計要件
- 一設計段階情報/完成度
- 一実施した設計評価とその結果たち
- 一設計者判断

が書かれており、簡単な操作で指定シーンや全体形状を確認できる。また形状表現そのものも、今までのベタ表現は止め、完成部/半熟部、重要部/そうでない部分、生技要件部分などが色や線種で簡単に識別できるようにする。

ーチェックリスト(チェックした項目/閾値/結果)は設計シーンとその上位の設計 戦略に配分する。

図 I- 2-26 データ表現

まず、従来の最終形状(形状モデル、幾何/位相/特徴)やパラメトリック情報が中核で表に見 えるデータ表現は駄目なのだと思われる。設計シーンの集まりとその相関が中核のデータ表現に 変える。

設計シーンはそれほど深くはないが、2階層ぐらいの階層があるでしょうね。

そのシーンを束ねる概念、例えば設計戦略というとしたら、設計戦略というので全体の設計を私はこういう風にやりますよというのが、シーン群、シーンの関係によって見える。そして、細かな情報はシーンと関係付けられている。そんなものが表(おもて)に出てくるようなイメージです。それをクリックすると、それぞれ必要な情報が出てきて、クリックする1つの対象には形状もあるという感じなのではないでしょうか。

では、シーンに何があるのかということですが、

・特に重視する設計要件

設計要件は一杯あるわけですが、特に米田委員がお話されたのは、いわゆる定型設計ではなく開発設計/新規ものですが、そのときはかなり設計者の個性が出てくるはずで、設計者あるいは設計者群が重視する設計要件は何かが明示的に捉えられる必要がある。

• 設計段階情報/完成度

いま、設計はどういう段階にいるか、その完成度はどうかである。

・実施した設計評価とその結果たち

どういう設計評価を実施して、その結果はどうであったか。

• 設計者判断

設計者はどう判断しているかである。

そんなことが、みんな表に出ていて、クリックするとパッと中身がわかる。簡単な操作で指定シーンや全体形状を確認できる。

また、形状表現そのものも、今までのものをベタ表現というのが正しいのかどうかわかりませんが、一様な表現は止めて、完成部/半熟部、重要部/そうでない部分、生技要件を考慮した部分/そうでない部分などが色や線種で簡単に識別できるような、少しは血の通ったデータ表現にすべきだろう。

ですから形状表現も今のままでは駄目ということです。

特にパラメトリックとの関連で言えば、ある物についてここは設計としてはほぼ決めたというところまでパラメトリック変更の影響を受けたらまずいわけです。その意味では、完成部/半熟部というのは、データ表現だけこうすればいいと言っているのではなく、機能もそれを認識した機能群に変えるということですから、そんなに簡単なことではない。

・チェックリスト (チェックした項目/閾値/結果)

チェックリストは設計シーンレベルに書くことが多いと思いますが、その上位の設計戦略に入る チェックリストも当然あります。

(2) CAD のイメージ

絵で書くと次の図のようになります。

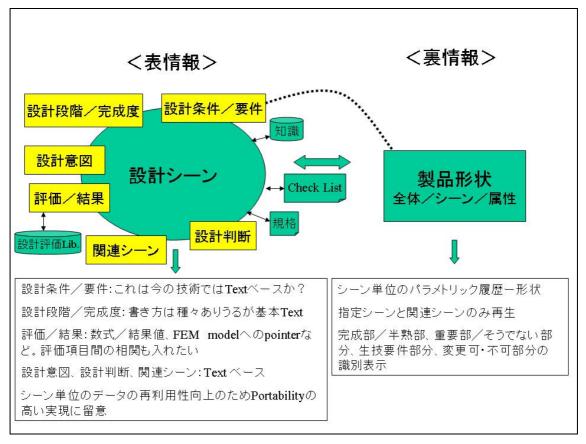


図 I- 2-27 CAD のイメージ

表(おもて)情報と裏(うら)情報があります。

①表情報

まず、最初は設計戦略が見える。戦略を見ると、どういうシーン達から成っていて、シーン間の 関係がどうなっているかが見える。「このシーンが見たい」となったらこの絵になる。

設計シーンのところに、設計条件/要件、設計段階/完成度、こういった諸々のものにピンが付いているような感じでしょうか。

設計評価ライブラリというデータベースとか、社内規格・業界規格、当該分野の設計知識とか、 あるいはチェックリストとかが必要な関係が付いて外にいます。

私の役割は、システム要件を整理するのと、有る程度実現性に関する判断をしろと言われている のですが、どう書けるかというと;

- 設計条件/要件
- 一部は数式で書けたり論理式で書けたりはあるのですが、大部分はテキストベースになるのかな。
- 設計段階/完成度

設計段階/完成度もきっとそうですね。

· 評価/結果

数式あるいは結果の値、FEM モデルへのポインタなどで表現されることになるでしょう。 評価項目間の相関も、人が最初1回は教えないといけないが、記憶しておけば活用できる。

・設計意図、設計判断、関連シーン

去年も議論したこの辺は、米田委員が満足する範囲のことを書こうとすると、殆どテキストになるのではないでしょうか。

世の中でパラメトリックをやっている人は、「パラメトリック情報は設計意図を表現している」と よく言うのですが、そんなことはほとんどないと思っていて、設計意図の内の数パーセントでし かないのではないかと感じています。

そういう意味では、米田委員がいうことをきちんと書こうとするとテキスト以外にない。ということは何を言っているかというと、そこである程度意味が失われてしまう。読んだ人しかわからない、つまりシステムが意味をわからないということになる。これが、わかる格好になるのは何十年先かわかりませんが、今のところ技術的目途が無いのではないか。

それから、シーン単位のデータの再利用性向上のためには、非常に Portability の高いシーンの実現をしないといけない。

シーン単位にデータあるいは関連情報を取っておくことはその会社の資産になる。あるいは日本 の設計技術の資産になるのだろうと思っています。

②裏情報

形状の方は良く我々はわかっているので、全体にしても、シーンにしても、属性にしても、シーン単位のパラメトリック履歴とか、形状表現とか、あるいは指定シーンと関連シーンのみの再生とか、完成部/半熟部、重要部/そうでない部分、生技要件部分、変更可・不可部分の識別表示、そういうことはできなくは無い。ただ今の CAD でこういうことがきちんと出来ているのがあるのかというと無いが、技術的には、きれいな格好ではできませんけれども、色・線種とかという形でしかないかもしれないが、実現性はあると思います。

前から言っていますように、製品形状にいろいろ属性をべたべた付けるのは間違っていると思っています。

製品形状は設計条件/要件が少し変わったら変わるのが当然です。ですから1つの製品をずっとトレースしたら、一杯形状がいる。

そういう意味からすると、形状側にそういうものを付けるのではなくて、基本的に重要なものは シーン側に付けていくことになるはずです。

だから、設計条件と要件を決めると、それに対応する形状がパッと出てきますという、形状はそ ういう一つの属性だと思う。

機械製品は、製品形状が非常に重要な属性であることは事実だが、IT システムの実装としては、 それを表面に出す今までの実装は正しくないのではないか、設計者寄りではないのではないかと 思っています。

(3) 機能体系

機能体系

- 機能体系もシーン単位にしたいが今の技術では無理。1つのシーンの設計にN個の設計/評価機能を使わざるを得ない。但し個々の機能は:
- 一 完成部/半熟部、重要部/そうでない部分、生技要件部分、変更可·不可部分への識別表示、システムの反応を可能にできる
- ー シーン単位のパラメトリック再生、評価機能も含めたパラメトリック再生、設計済み部非変更再生も技術的に可能
- 勿論形状特徴を今日のいい加減な実装でなく、設計や生産準備上のintelligenceを持ったものにすることは、やるべきだし技術的にある程度は可能。但し位相上の制約は付けざるを得ないのである特徴は設計シーンと1-1かも知れないが、一般に;設計シーン <> 形状特徴と認識
- 設計者用語での使用については呼び名の適用性の懸念(分野/設計対象)あり。簡単にカスタマイズできる仕掛けは可能性あり。
- 効率的設計評価の仕組みは設計段階情報と組み合わせ優秀な設計者の評価項目選択(順番含む)を学習し、それをTemplateとして記憶。当該設計シーンが参照されたとき、参考情報としてTemplate表示やTemplateに沿ったアドバイスは可能。それ以上高度な支援は技術的目処なし。効率的設計評価を支えるデータベース構築も実現法は同様。

図 I- 2-28 機能体系

• 機能体系

機能体系もシーン単位にしたいが今の技術では無理です。1 つのシーンの設計に N 個の設計/評価機能(コマンド)を使わざるを得ない。個々の機能については、

- 完成部/半熟部、重要部/そうでない部分、生技要件部分、変更可・不可部分への識別表示、 システムの反応を可能にできる。

スマートな格好でないかもしれないが、現場の設計者・生産準備技術者が「うん、これならいいよ」という程度ぐらいまでは出来るのではないかと思っています。つまり、充分な数学的、理論的根拠に基づいては出来ないけれども、実用的なものは出来るだろう、という意味です。

一設計シーン単位のパラメトリック再生、評価機能も含めたパラメトリック再生、設計済み部非変更再生も技術的には今でも出来る。

"一"で書いたところは、技術的には全然遠くないと思っている。やればよいのではないか。

• 形状特徴

ある CAD ベンダが、implementation を簡単にするために全ての形状要素をフィーチャと言って しまったため、元々のフィーチャから非常に乖離してしまった。

設計や生産準備上の intelligence あるいはセマンティックスを持ったものにすることは、今はある程度出来るようになったと私は思っています。ただし形状を扱うとなると位相上の制約は付けざるを得ないので、ある特徴は設計シーンと1対1かも知れないが、一般には設計シーンと特徴は1対Nになってしまうと思います。

• 設計者用語

設計者用語での使用については、私は昔から悩んでいたことで、呼び名の適用性には懸念を持っています。

分野とか設計対象によって呼び名が違う。そういうときに、ITシステムが勝手に呼び名を決めていいということではない気がしますので、分野毎に、簡単にカスタマイズできる仕掛けを作った方がいいのではないか。

• 効率的設計評価の仕組み

設計段階情報と組み合わせて、優秀な設計者の評価項目選択(順番含む)を学習し、それを Template として記憶しておいて、当該設計シーンが参照されたとき、参考情報として Template を表示し、Template に沿ったアドバイスは出来る。

それ以上高度な支援は技術的目処がない。 効率的設計評価を支えるデータベース構築も実現法は 同様。

機能体系(つづき)

- サクサク評価はFEMに関しては余り良い手が浮かばないが、設計評価Libraryにユーザが自由に簡易機能を追加する仕組みは可能。
- 評価結果と判断の蓄積/活用はこのシーン単位の情報の利活用で用を成すか?
- 変更可能部分/変更の影響範囲については、前者はデータに識別子を付けるだけで 簡単に実現できるが、後者はパラメとリックに関係ついている部分を知らせる程度しか手 が無い(これでは設計上だめでしょう)
- 設計のストーリー化要件は、設計シーン内は上記の通り。設計シーンをまたがるストーリー化についても重要と思われ'設計戦略'という'設計シーン'の上位型を作成し、効率的設計シーンの組み方にかかわる条件/留意点を書くことは可能。
- 'モノ作りの流れに沿った新規設計でも使える柔軟な意味付けされたデータ群'という 重要な要件の実現はこれまで述べた設計シーン群をPartsとしたいが不十分?
- 規格とデータに密結合要求は、規格の充足度checkerを順次開発する程度か。
- チェックの効率化要求は設計シーンや設計戦略にCheck listがついているので前段階のそれらを参照して先ずは人が判断。将来的には何らかの自動化やアドバイス機能も技術的には可能と思われる。

図 I-2-29 機能体系(つづき)

サクサク評価

FEM は昔に比べれば良くなったと言っても、どんなにコンピュータが速くなっても、FEM という数値計算技術は明らかに限界があって、設計サイクルの中で、重長な FEM 計算がくるくる回るということはありません。

そういう意味では、余り良い手が浮かばないが、設計評価 Library にユーザが自由に簡易機能を 追加する仕組みは可能なのではないか。

・設計評価結果と判断の蓄積/活用

この紙の中ではシーン単位のデータの利活用を組み込んでいますが、それで米田委員が目指したものが実現できるのかどうか、私は疑問です。逆に何処までできれば良いのかを質問したい。

・変更可能部分/変更の影響

言葉で言うのは簡単ですが、難しいですね。

前者はデータに識別子を付けるだけで簡単に実現できるでしょうが、後者はいろんな設計を考え

たときにどの部分を変えたらどう影響するか、これを定義するのはきっと不可能な問題です。 パラメトリックに関係している部分なら今でもできています。しかしながら今のパラメトリック は非常に硬い技術ですから、しかも位相的にかなり制約のきつい技術なので、今のパラメトリッ クで関係付いている部分を知らせる程度で設計者が満足するはずはきっとないと思います。 なので、変更の影響範囲を正しく知らせるのは難しいですね。

やはり、非常にアドホックかもしれないが、認知された優秀な人がチェックした結果はある程度 信用して、何回もその分はやらなくていいようにするとか、スキッとした格好の実現にはならな いけれども、チェックの回数減らすには、そういう手段を使うしかないのかな。

・設計のストーリー化要件

設計シーン内は上記の通り。設計シーンをまたがるストーリー化についても重要と思いますし、 効率的な設計シーンの組み方にかかわる条件とか留意点を書くことはできると思います。

・モノ作りの流れに沿った新規設計でも使える柔軟な意味付けされたデータ群 大きな話ですが、ここに書いたような設計シーン群を Parts にしたいが、それでよいかを議論し たい。

・規格とデータの密結合要求

あまりうるさく考えるとキリがないので規格の充足度 checker を順次足していくのでよいのではないか。「ポンと置いたら何とかなる」という世界ではないけれども、少なくとも、データと規格がある程度意味を持ってインテグレートされているという格好にはなるので、順次開発を充実すればするほど、設計にとっての自然さとか充足度は上がるのではないでしょうか。

・チェックの効率化要求

設計シーンや設計戦略にチェックリストが付いているので前段階のそれらを参照して、先ずは人 が判断するのがよい。

将来的には何らかの自動化やアドバイス機能も技術的には可能かもしれませんが今のところ不明です。

2.3.4 付論

Appendix

先ずは純粋な設計行為を支えるCADに絞って貢献を言われた。

以下の事項も検討の候補には挙げたほうが良いと思われる。

1. 設計者間、設計者と生産技術者間の緊密な/効率的な相互連携を支えるIT技術開発(抜本的N人用のEngineering Systemの実現)

日本と米国など空間上の壁を越えて設計者が同一モデルを見ながら自由にコミュニケーションし設計を効率化する仕組みである。操作権、データ更新権を授受しながら作業する環境が不可欠なため強力なデータベース機能が中核となる。ある設計工程に限定した実験システムは作ったが、実用までには距離がある。但し従来の形状中心でなく、本書に述べた設計シーンに含まれる各種情報を共有できれば少しは実用にちかずくと思われる。

2. 情物融合化による情報モデルの信頼性向上と活用性拡大

正解を知らないDigital化推進一辺倒は危険。最近の計測技術向上を生かして計測データと数学モデルを融合させて対象を表現したり、計測データにもとずいて数学モデルの信頼性を高めることは実用上有用である(最近CADモデルに関する革新が無いことも問題だが・・)。

図 I- 2-30 Appendix

米田委員は先ずは純粋な設計行為を支える CAD に絞って要件を言われた。さらに以下の事項も 検討の候補には挙げたほうが良いと思います。

(1) 設計者間、設計者と生産技術者間の緊密な/効率的な相互連携を支える IT 技術開発 抜本的 N 人用のエンジニアリングシステムの実現ということにも絡むのですが、先ほど紹介した 某社用に作ったプロトタイプシステムなども考慮してどう実用性を上げるかは非常にチャレンジ ングな課題です。

昔に比べて実用的な操作とか実用的なパフォーマンスで、できる環境が少しずつ向上したので、 現状でどこまでできるのか、何年後にはどこまで行くのか、そういう議論はできるのではないか。

(2) 情物融合化による情報モデルの信頼性向上と活用性拡大情物とは、情報と物の意味です。

今の設計、生産準備の問題は、数学問題を解いている訳ではないから、正解がわからないまま仕事をしているわけです。

従って設計者が言ったとおりにモデリングしたつもりで、そのモデルに基づいて試作して実験してみると、実はおかしいねということがある。

情物融合によって情報モデルそのものの信頼性を上げるとか、情報モデルの活用性を拡大するのは重要です。これは日本固有ではなく、世界共通です。

正解を知らない Digital 化推進一辺倒は危険だと思っています。最近の計測技術向上を活かして 計測データと数学モデルを融合させて対象を表現し、計測データにもとづいて数学モデルの信頼 性を高めることは実用上有用だと思います。

計測関係は進歩していると思うが、CAD 関係の基礎技術は殆ど革新が最近無いので、世界の研究者の焦点はわかりませんが、鈴木先生に後から教えて頂きたいのですが、研究課題的なことは何がポイントなのかよく理解した上で検討したい。

今、自動車会社は特に大きな OEM は「世界の型同時立ち上げ」という言葉で代表される共通課題を持つ。大手各社は世界各地に型工場持っています、当然設備も違うし、技術力も違うし、仕事の進め方も違うし、文化も違う。そういう状況の中で、世界戦略上、納期をきっちり満たした品質の高いモノを作りたいという要件です。

この大変な要求を実現する手段として、型の世界では計測が1つの道具になる。

数学モデル通り、あるいは成型見込みを一応形状にして、それで型削って、何も補正しないで済む型はゼロです、必ず現場で補正する。現場で補正した結果、現場の神様がこれでいいと言った型をきちんと計測(全面計測なり、部分測定なり)して他国の工場での2番型製作に使うということは現実に見通しがある。

以上は一例ですが、情物を融合させて、情の信頼性を高めることは、今申し上げた以外にも、もっといろいろ可能性があるのではないかという感じがしています。ここで追求する議論かわからないが、将来のエンジニアリング IT を考える意味では、(1) も (2) も我々忘れてはいけない大きな課題ではないでしょうか。

非常に雑駁ですが以上です。

2.4 設計と生産サイマルの事例

設計と生産サイマルの事例

日産自動車株式会社 牧野寛幸

図 I-2-31 設計と生産サイマルの事例

設計と生産を同時に進める (サイマル) の事例についてお話します。

【背景】

- 開発プロセスのデジタル化に伴うフロントローディングにより、技術工数が不足しているという実態がある。
- グローバルでの生産地域が広がり、技術検討工数が従来以上に必要になっている。
- モノを作る前のフェーズ=デジタルでの仕事の仕方をもつと見直して、効率的にやらなければいけない。
- 技術者の声
 - (1)デジタルフェーズの仕事のやり方が、まだ整備されていない。やり方が浸透しているとはいえない。
 - (2)同じチェック、フィードバックを繰り返している。

図 I- 2-32 背景

まず、私どもが置かれている状況についてですが、

- ・開発プロセスのデジタル化に伴うフロントローディングにより、技術工数が不足しているという実態があります。
- ・グローバルでの生産地域が広がり、技術検討工数が従来以上に必要になっています。
- ・モノを作る前のフェーズ=デジタルでの仕事の仕方をもっと見直して、効率的にやらなければいけない。

技術者からの声は、次のとおりです。

- -(1)デジタルフェーズの仕事のやり方が、まだ整備されていない。やり方が浸透しているとはいえない。
 - -(2)同じチェック、フィードバックを繰り返しています。

【現状認識】

- デジタルで仕事をしているが、実際のモノを作っていたときプロセスは変わっていない。
- 本来でいえば設計の中で生産要件チェックできる仕組みを 構築したい。
 - 部分的には入れているが、全て入るわけではない。
 - 一体化した仕事の仕方をしないと非効率
- 生産要件確認・評価方法について設計側と生産技術側で、 デジタル化に合った最適なプロセス(約束事・プロセス)があ るのではないか?
- 設計から生産性検討のプロセスのシーンを変えられるような CADとかCAEからのアプローチはもう少しできないか?
- バラバラなシミュレーション等の環境を1つにしたい。いろんな検討、成形シミュレーションとか、搬送のシミュレーションとか別々のシステムになっているので分業化せざるを得ない状態になっている。

図 I- 2-33 現状認識

現状について整理すると、

- ・デジタルで仕事をしているが、実際のモノを作っていたときプロセスは変わっていない。
- ・本来でいえば設計の中で生産要件チェックできる仕組みを構築したい。
 - -部分的には入れているが、全て入るわけではない。
 - -一体化した仕事の仕方をしないと非効率
- ・生産要件確認・評価方法について設計側と生産技術側で、デジタル化に合った最適なプロセス (約束事・プロセス) があるのではないか。
- ・設計から生産性検討のプロセスのシーンを変えられるような CAD とか CAE からのアプローチはもう少しできないか。
- ・バラバラなシミュレーション等の環境を1つにしたい。いろんな検討、成形シミュレーションとか、搬送のシミュレーションとか別々のシステムになっているので分業化せざるを得ない状態になっています。

2.4.3 実例の紹介

(1) 紙ベースでの情報伝達

「実例の紹介(1):紙ベースでの情報伝達」

- 1).設計の標準書
- 2).不具合フィードバックの仕組み
 - システムは外付けのシステムで、CADのデータの絵を貼り 付けている。

昔、人が一所懸命紙で書いて電話で連絡していたのが、ちょっと便利な帳票データベースが出来てメールで飛ばしてやっているだけで、仕事のスタイルは変わっていない。

そのためフロントローディングしても、確かに電子な ので昔より速いが、工数は減らない。

図 I-2-34 紙ベースでの情報伝達

設計と生産の間における不具合のフィードバックを例に説明いたします。

昔、人が一所懸命紙で書いて電話で連絡していたのが、ちょっと便利な帳票データベースが出来てメールで飛ばしてやっているだけで、仕事のスタイルは変わっていない。

そのためフロントローディングしても、確かに電子なので昔より速いが、工数は減らない。

「実例の紹介(2):CADと周辺システムの連動」

- CADと不具合情報の連動が切れている。
 - CADとリンクしてくれないと、仕事のスタイルは変わらない。
- CADと周りのいろんなシミュレーションとかナレッジと かの情報がもう少しリンクできないものか?
- 本来CADは、モノといっしょで情報のハブであって欲しい。そこから全部の情報が手繰れてほしい。
 - 今は、CADが参照される側になってしまっている。
- 一方で、全部CADの中に持ち込もうというのは逆に CADが重くなりすぎてしまう。

図 I-2-35 CAD と周辺システムの連動

CAD システムとの関連から見ると、

- ·CAD と不具合情報の連動が切れています。
 - -CAD とリンクしてくれないと、仕事のスタイルは変わらない。
- ・CAD と周りのいろんなシミュレーションとかナレッジとかの情報がもう少しリンクできないものか。
- ・本来 CAD は、モノといっしょで情報のハブであって欲しい。そこから全部の情報が手繰れてほしい。
 - -今は、CAD が参照される側になってしまっています。
- ・一方で、全部 CAD の中に持ち込もうというのは逆に CAD が重くなりすぎてしまう。

【課題のまとめ】

- デジタル化をしたが、手順は昔のフィジカルなフローと変わっていない。
- 設計・生産技術含めてCADと品質の情報が連携していない。
- だから、品質のチェックをしてそれをフィードバックしても、次に修正されたものが、本当にそれが反映されているのかどうか分からない。結果的に、繰り返しのチェック工数が多くなっている。
- ニーズとしてはパラダイム変換をしていきたいが、ベースがこういう状態になっているため、ユーザの力だけでブレークスルーするのは難しい。もう少しツール主導でやって欲しいと思う。
- CADのシステム的な対応不足という側面の認識は必要。
 - 生産技術のフィードバックが入っているか分からない
 - 設計が変更したとき、設計側もどこまで影響が及ぶかわからない。だから、 ここはチェックしなくていいといえない。
 - 結果的にフェーズ毎に何回も繰り返しチェックがおこる。

図 I-2-36 課題のまとめ

以上をまとめます。

- ・デジタル化をしたが、手順は昔のフィジカルなフローと変わっていない。
- ・設計・生産技術含めてCADと品質の情報が連携していない。
- ・だから、品質のチェックをしてそれをフィードバックしても、次に修正されたものが、本当に それが反映されているのかどうか分からない。 結果的に、繰り返しのチェック工数が多くなって います。
- ・ニーズとしてはパラダイム変換をしていきたいが、ベースがこういう状態になっているため、ユーザの力だけでブレークスルーするのは難しい。もう少しツール主導でやって欲しいと思います。
- ・CAD のシステム的な対応不足という側面の認識は必要。
 - 生産技術のフィードバックが入っているか分からない。
- -設計が変更したとき、設計側もどこまで影響が及ぶかわからない。だから、ここはチェック しなくていいといえない。
 - 一結果的にフェーズ毎に何回も繰り返しチェックがおきます。

2.5 設計と生産サイマルのシステム要件

設計と生産サイマルのシステム要件

デジタルプロセス株式会社 森 博己

図 I-2-37 設計と生産サイマルのシステム要件

2.5.1 設計と生産サイマルの課題

(1) 狙いと主な課題

設計と生産サイマルの課題

【狙い】

- "デジタルフェーズの技術工数半減プロセスの明確化"
- (解釈) 製品設計情報に対して、生産技術の側面から、設計へ フィードバックを必要とする 要件チェック作業の効率化

・【主な課題】

- デジタル化は達成したが、業務フローは以前と変わっていない
- CAD(3D-形状)と、品質管理システム(製造要件や過去の不具合情報)が連携していない
- 以前に設計へ指摘したフィードバックが、既に折り込まれている のか否かが判らない
- 開発フェーズごとに、同じ要件チェックを繰り返している。

図 I-2-38 狙いと主な課題

"デジタルフェーズの技術工数半減プロセスの明確化"を私なりに一般的な解釈に変えてみますと、「設計部門から出てきた製品設計情報に対して、生産技術の側面から、設計へフィードバックを必要としますが、その要件のチェック作業を効率化したい。工数的な半減なのか期間的な半減なのか別にして、総論としては効率化したい」と捉えています。

→それもありますが、それだけなのかな。

つまり、半減すると言うときに、情報化してデジタル化して、設計そのものを短くしたい、生産 準備工程そのものを短くしたいということがあるはずで、「生産技術の側面から、設計へフィード バックを必要とする要件チェック作業の効率化」だけ言っていいのかという気がします。

←開発行為全体の期間短縮・効率向上ともとれるが、牧野委員の話は、殆どが生産技術の側面からの要件チェックの効率化の話でした。ということで、多少ここに焦点を当てた解釈をしています。それだけでいいとは思っていませんが、先ずは牧野委員の狙いに焦点を当てて考えました。

牧野委員から出てきた課題は、

- ・デジタル化は達成したが、業務フローは以前と変わっていない。この場合も、製造要件のフィードバックの業務フローについて説明されたと思います。
- ・CAD (3D・形状) と、品質管理システム (製造要件や過去の不具合情報) が連携していない。 具体的には品質情報システムとの連携を挙げられた。製造要件や過去の不具合情報が連携していない、全く関連性を持っていないという状況ですね。
- ・以前に設計に指摘したフィードバックが、次のフェーズで織り込まれているか否かが判らない。
- ・開発フェーズ毎に、同じ要件チェックを繰り返している。

→フィードバックという言葉が引っかかる。フィードバックとはどこかからどこかへ戻すことだと思う。牧野委員はフィードバックだけのことを言っているのではなくて、設計の時点で作れるものをきちんと設計しなさい。設計者に与えられたミッションだけやるのではなくて造れるものを設計しなさい。つまり、生産技術上の要件も考慮した設計をする、そういうことと思う。フィードバックという言葉が適切なのかという感じがします。

←今の設計と製造という役割分担を仮に是とすると、実質的にやっていることはフィードバックだと思います。設計段階から全ての製造要件を織り込んでいくべきだ、という考え方もあるが、その場合についてはここでは述べていません。

スーパエンジニアの方もいるが、一般的には、「製造要件はよくわからない、ともかく製品としてはこうありたい」ということで、今の主な自動車会社はそういう分け方になっていると思います。

→そういうこともあるかもしれないが、下流まで持っていってだめだから折り返して直すのがきっとフィードバックだと思う。そうでなくて、できるだけ折り返さないでいいような設計をしましょうということではないか。

例えば意匠のパネルの場合、後から成型しても不都合の無いような意匠を最初から設計しなさい、 という方向に帰ってきているのではないかと思っているのですが。

←私の解釈は、当たり前の製造要件はどんどん帰ってきている。具体的には、「パネルはプレスで造られる、だからインバースは絶対してはいけない」。シリンダブロックであれば、「見切りを決めて抜けないといけない」。これぐらいのことは、設計者は考えていると思います。

ただ、もっと細かな、「これを組み付けるときに、このネジはこっちから締める」、という類が一杯ある。牧野委員が言われた何千件もの要件は、製造技術寄りの人でないと把握できない。設計者がそれを全部考慮しないといけないというのは酷ではないかと思います。最終的に製造について造れるかどうか責任を持つのはあくまでも製造部門、生産技術部門という考え方でいえば、ま

だ実際のやり取りはフィードバックに近いのではないか。

→できるだけ折り返さないでいい設計をしたほうがいいわけで、その方向を狙うことが必要なのも決まっているわけだが、平均的な設計者を考えると、やっぱりフィードバックでぐるぐる回ることがある、だから効率化しますといっているわけですね。 今は絞っているが、それでthat's all じゃないよね、といっているわけです。

←自動車でなくてもっとシンプルな製品であれば、製造要件のチェックを一人でやることは有り得るわけで、それは多分業務分担上一番効率がよいと思う。ただ、あまりにも製品が大きくなったり、複雑になったりすると、どっかで分けないといけないということになる。自動車、飛行機、バイク... において、設計と生産・生産技術の間に線が引かれるというのは、何らかの理由があるのではないか。それは、細かな生産要件まで設計者が考慮できないということではないか。

(2) 設計と生産技術間の情報連携

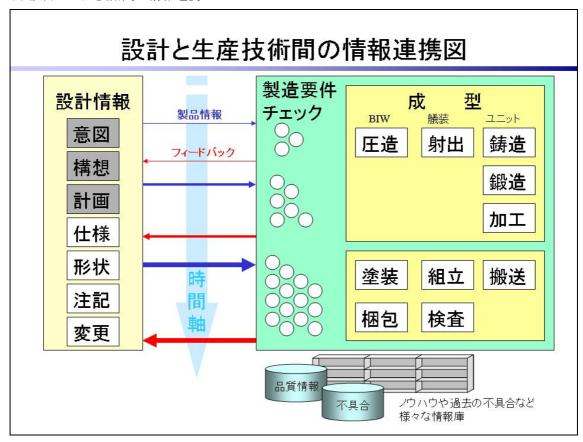


図 I-2-39 設計と生産技術間の情報連携図

設計から情報が来るわけですが、意図、構想、計画は殆ど来なくて、実際に来るのは仕様、形状、 注記、設計変更の通知書です。

製造要件は大きく分けると、形を造る(圧造、射出、鋳物、鍛造、機械加工で形を造る)ための ものと、塗装、組立、搬送、梱包(部品のノックダウン生産で必要)、検査といった生産のための 要件があります。

要件のチェックは何に基づいてするかというと

- ・品質の情報 これはデータベース化されている。
- ・不具合 記録は電子化されている。
- ・棚みたいなところに仮に入れられている情報 何ら関係付いていない情報

を元に製造要件をチェックしています。

時系列的にみると、最初は設計の意図も非常にラフなものが来ます。チェックの回数も深さもそれほど大きくなく、絶対守らないといけない重要なものだけをチェックしてフィードバックします。それに変更がかかってチェック、また、変更がかかってチェックを繰り返します。

以上が、牧野委員のお話から受け取った現状です。

(3) 課題とあるべき姿

課題と、あるべき姿

- ◆ デジタル化は達成したが、業 務フローは以前と変わってい ない
- ◆ CAD(3D-形状)と、品質管理 システム(製造要件や過去の 不具合情報)が連携していな い
- ◆ 以前に設計へ指摘したフィード バックが、既に折り込まれてい るのか否かが判らない
- ◆ 開発フェーズごとに、同じ要件 チェックを繰り返している

- 1. 電子化された情報管理のしく みをフルに活用したプロセス に従って、業務が遂行されて いる
- 2. 設計情報(CAD、PDM、BO Mに保持された部品、アセンブリ、仕様など)に、関わる製造要件が自動的(もしくは容易)に検索され、自動的に(もしくは容易に)合否判定がさせる
- 3. 設計と生産技術の間における、 適切な情報が、適切な時に、 正しく伝達され、履歴も含めて 管理されている

図 I-2-40 課題とあるべき姿

各課題について、あるべき姿を考えてみました。

- ・電子化された情報管理のしくみをフルに活用したプロセスに従って、業務が遂行されている。 先ほどの棚と製造要件が何ら連携されていない。製品のデータはデジタル化されたが、棚の 情報は Excel であったり紙であったりで整理されていない混沌とした状態で管理されている。 あるべき姿においては、それが全て電子化された情報になる。それから、プロセスでいうと、 生産技術部門だけではなく設計側も参照できる。
- ・設計で公式に管理されている情報(CAD データ、PDM 情報、BOM に保存されている部品・アセンブリ・仕様に関する情報)に関わる製造要件が自動的もしくは容易に検索され、自動的にもしくは容易に合否判定が出来る。

設計情報と製造要件とのうまい連携ができればこういうことができるのではないか。

・設計と生産技術の間における、適切な情報が、適切な時に、正しく伝達され、変更された履歴も含めて管理されている。

2.5.2 設計と生産技術間の情報伝達の課題

設計と生産技術間の情報伝達の課題

- ◆ 設計変更情報の把握が困難
 - 1. 設計変更箇所の分類
 - 2. 設計変更理由の分類
 - 3. 設計変更を生じた、問題と対策の関係
 - 4. 設計変更における魔のループ(想定例)
- ◆ 設計変更と製造要件チェックの関係図
- ◆ 設計変更と製造要件チェックの関係の課題

図 I-2-41 設計と生産技術間の情報伝達の課題

設計と生産技術の間の情報伝達の課題が大きそうだと思いましたので、整理しました。 まず、

- ・設計変更情報の把握が難しいのではないか。
 - -どこを、変えたのか(設計変更箇所)。
 - -何故、変えたのか(設計変更理由)。
 - -設計変更を生じた実際の問題と対策の関係
 - -設計者が陥りやすい魔のループ、抜け出せないループ

それから

- ・設計変更と製造要件チェックの関係
- ・設計変更と製造要件チェックの関係の課題

と整理しました。

(1) 設計変更情報の把握が困難

設計変更情報の把握が困難

設計と生産技術の間の、設計変更情報について、生産技術者が知るべき情報が把握出来ていない

- ◆何時、変えたのか?
- ◆何を、変えたのか?
- ◆ 何故、変えたのか?
- ◆どのように、変えたのか?

以前に、指摘したにも関わらず

◆ 何故、変えなかったのか?

図 I-2-42 設計変更情報の把握が困難

設計と生産技術の間で、本来生産技術者が知るべき情報が伝わっていないのではないか。

- 何時、変えたのか
- 何を、変えたのか
- 何故、変えたのか
- どのように、変えたのか

もうひとつは、前回言ったのに何故変えなかったのか、こういった問題があるのではないかと思います。

1. 設計変更箇所の分類

- ◆ 何を、変えたのか?
 - 部品構成の変更
 - 部品の差し替え
 - 部品の配置変更
 - 部品単体の変更
 - 外部形状(OEMが承認すべき形状)の変更
 - 造形形状
 - » 全体的
 - » 部分的
 - 機能形状(取り付け面形状など)
 - その他の仕様変更
 - 要求仕様(耐熱、強度····)
 - 内部形状(サプライヤーの都合)の変更
 - コーションラベル変更(←この変更は多い)
 - 設定色の変更(樹脂部品の内装色の追加や削減)

図 I-2-43 設計変更箇所の分類

一般に設計から変更が通知される内容には、次のようなものがあるのではないでしょうか。 まず、何を変えたのか、という点について、

[部品構成の変更]

部品の差し替えや配置変更がある。ユニットものは標準部品を使いながら場所を変えてみること が多い。

[部品単体の変更]

・CAD データが持っている外部形状(自動車会社が承認すべき形状)の変更

自動車会社は部品の8割は自分のところで作っていない。ですから、自動車会社が承認しているのは、大体、外側の形です。例えば、AM/FM付きラジオは、この形で納めます、と言われて承認する。自動車会社の設計部門が把握しているのは外側形状です。変わるのは主に2つあります。

- ーデザイン的に造形の都合で、全部もしくは部分的に変える
- -設計の都合で、機能的な形状(取り付け面など)を変える

・その他の仕様変更

- -要求の性能、耐久性能などの仕様 例えば、耐熱温度や要求強度。
- 内部の形状

CAD データは全く変わりません。内部の形状や構造が、部品メーカーの都合で変わりました。ですから部品としては変番になっているかもしれません。

ーコーションラベル

結構多いのが、このコーションラベルです。部品を良く見るとシールを貼っています。 シールは仕向け地で全部違う、アフリカに出すときはアフリカ用のものです。 コーションラベルの追加変更は結構多いですが、形状は直しません、記述だけ直しています。

設計変更を洗い出してみると、意外に CAD の形状には全く関係ない、設計変更が多く起こっています。

②設計変更理由の分類

2. 設計変更理由の分類

- ◆ 何故、変えたのか?
 - 設計要件
 - 機能、性能、品質、見栄……向上
 - 強度、剛性、耐久、音振……向上
 - 重量、原価・・・・・・・・・・・低減
 - 干渉………回避
 - 製造要件
 - 生産性
 - 成形性(圧造、射出、鋳物、鍛造、焼結)
 - 加工性(機械加工、放電加工、溶接)
 - 組立性、可搬性、その他作業性
 - 製造コスト低減
 - 素形材量、歩留まり、加工作業数
 - 製造(バラツキなど)品質向上、
 - 不良率向上、精度向上

図 I-2-44 設計変更理由の分類

何故変えたかです。

[設計要件]

こういう情報は、基本的には来ません、推測するだけです。

- ・機能、性能、品質、見栄 直接商品性に関わるところを向上する。
- ・強度、剛性、耐久、音振 隠れた絶対に守らなければいけない部分です。
- ・重量、原価これは低減する。
- ・干渉問題 これは起こったら回避する。

[製造要件]

• 生産性

生産技術部門からフィードバックを受けて生産性が悪いから変えて欲しいと、 例えば、

- -成形性(圧造、射出、鋳物、鍛造、焼結)
- -加工性(機械加工、放電加工、溶接)
- 一組立性、可搬性、その他作業性
- これらが悪いから製品形状を変える。

・製造コスト低減

- -素形材の量を減らす 「こういう形にしてくれたら2個取りができる。」
- 一歩留まりを上げる
- 一加工作業数

「この形にすると、加工のステップでドリル加工が3方向で済む。あなたの工法だと4回やらないといけない。」

製造のバラツキ

これなども品質向上するために、例えば精度を ± 0.5 としていたのを+0 の-1 にする。こうした表記にすることによって、不良率のバランスをうまくとる。

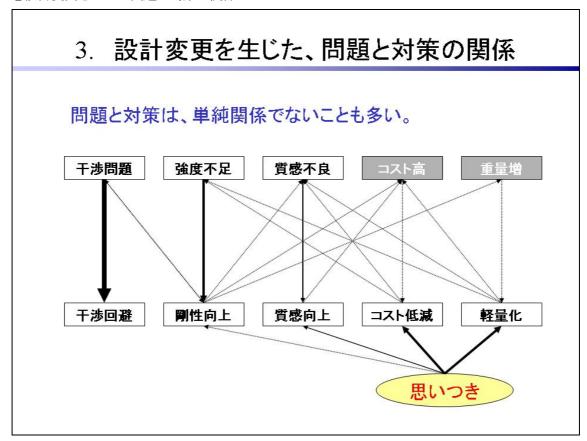


図 I-2-45 設計変更を生じた問題と対策の関係

なんらかの問題があって設計変更は起こりますが、問題と対策は単純でないことが多いのではないでしょうか。この図では、わざと上と下を対比させました。

- -干渉が起こったから、干渉を逃げた
- 一強度が足りないから、剛性を上げた
- 質感が不良だから、質感を上げる
- ーコストが高いから、コストを下げる
- 一重量が重いから、軽くする

このように対比してみて思うのですが、現実は質感が不良だからというのが発端になることはあまりなく、何らかの軽量化が主因だったり、何か問題が起こって結果的に質が悪くなった、というものが多い。最初からコストが高いということはなく、強度を上げたりするとコストが高くなった、というわけです。剛性を上げなければならなくなると、大概の場合重量なのです。いろいろコストが上がるという状況です。

コスト低減をしないといけない、軽量化しないといけない。これは大概の場合、発端は、「アッ、

こんなことだったら今度低減になるな」という思い付きから入っているはずです。決して何かやれと言われているわけではありません。コスト低減とか軽量化はずっと頭の隅に置いていますから、ちょっとした思い付きで、「アッ、これでもう少しコスト低減できるかな」という思い付きが発端になっていることが多いと思います。

④設計変更における魔のループ

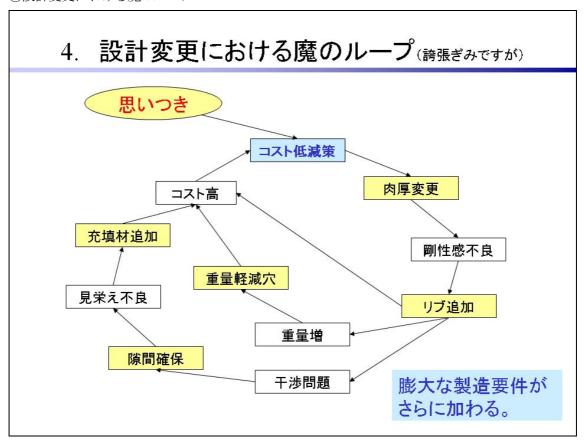


図 I-2-46 設計変更における魔のループ

これは設計要件についてだけです。

例えば、思いついて、「これでコストが下がるのではないか」と低減策を考えたところから始まり、 『肉厚を変更する→材料が安くなる→剛性感が凄く無くなってペコンペコンになる→リブを追加 →コストが上がる、重量が増す、干渉の問題が起こる→しかたがないから、インストルメントパネルに少し隙間をあける→隙間の見栄えが悪くなる→隙間に充填剤を埋める→コストが上がる→コスト低減しないといけない』に戻ります。意外とこういうループがある。

必ず何かの対策をしたら、その対策が別のものに影響する。この絵の中に、製造要件を一切入れていない。現実にはこれに製造要件が一杯加わります。

例えば、リブが加わると、そのリブがあるために、何かが取り付けにくくなるとか、手が入らな

くなるとか、となります。

私がここで言いたいのは、原因と対策は大概の場合、1 対 1 に関係するような単純なものではないということです。

(2) 設計変更と製造要件チェックの関係

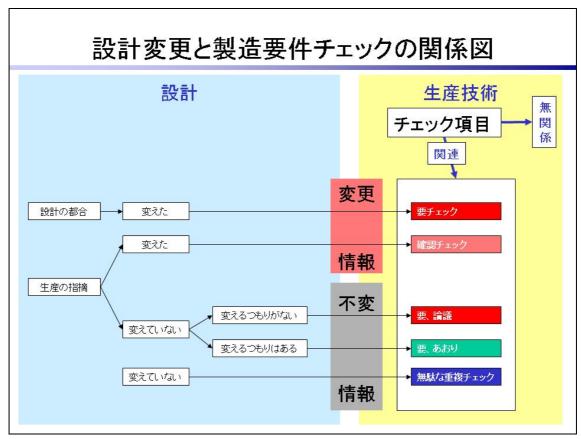


図 I-2-47 設計変更と製造要件チェックの関係図

設計と生産技術の間で情報が受け渡ってくるわけですが、

- ・設計の都合で、以前のものよりも変えた。 当然ここは変わった情報になる。
- 生産技術の指摘で設計が変えた。「これはプレスできない」と言われて変えた。
- ・生産から指摘されたのに変えてない。これには二つある。
 - 一変えるつもりがない。
 - -変えるつもりはあるが、そこまで手がまわっていない。
- ・全く何の考えも無く変えていない。 問題だと思っていないし、誰からも指摘されていない。

このように、設計から生産技術に渡る情報というのは、変わった情報と変わらなかった情報にまとめられるのですが、それを受け取った生産技術側は、

・設計の都合で変えたものは、きちんとチェックしないといけない。

- ・自分達の指摘によって変えたものは、確認のチェックをしないといけない。
- ・自分達が指摘したが、設計は変えていない、かつ、変えるつもりはない、と言っているものは、 きちんと議論するべきです。
- 変えるつもりはある、と言っているのだったら、これはあおらないといけません。
- 変えてないもの、これをチェックすると無駄なチェックになります。

生産技術側にも、チェックの項目は、まだリストベースですがあります。その中で、例えばドアパネルに関連するものか無関係なものか、を生産技術者は分けています。ここにやはり、どう分けているかという問題があるのではないかと思います。

(3) 設計変更と製造要件チェックの関係の課題

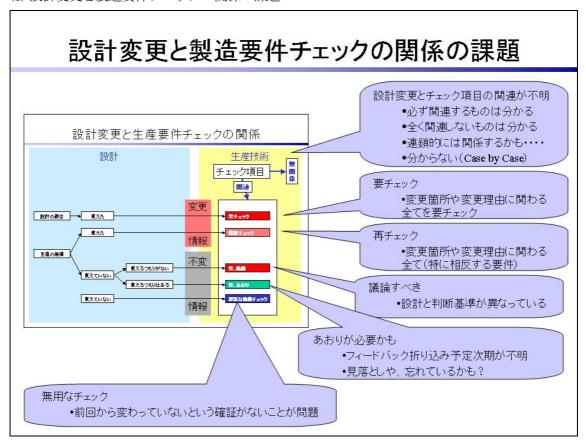


図 I-2-48 設計変更と製造要件チェックの関係の課題

前の図に注釈を付けています。

- ・仮に人間がやるとして、設計変更とチェック項目の関連がきちんと識別できるか。
 - 必ず関連するものは分かる

例えば、製造チェックの項目にドアのプレスのことが書いてあったら、当然ドアのデータが来れば関係するということがわかります。

- -全く関連しないものも逆に分り易い
- ーややこしいのは、連鎖的には関係するかもしれないものです。

例えば、「水密性のチェックをする」とあるとき、「私の担当は、バンパなんだけど、も しかしてバンパで水が漏って困る所はあるか」と、例えばターンシグナルランプを埋め 込み式の組み立て方でしていると、もしかするとそこに水が入るかもしれない。

- 一般的なバンパは関係無い、しかし連鎖的には関係があるかもしれない。
- 一全く分からない (Case by Case)

チェック要件を本当にチェックする必要があるのかどうか分らない。

この様に判断がつかないものがあると思います。

- ・設計の都合で変えて今回新しく変わったもの。 変更に関わった所、変更に関わった理由、これらについては全てをチェックする必要があります。
- ・生産技術が変えろと言ったので変えました。 変えた場所が本当に生産技術の要件をクリアしたかどうか、再チェックする必要があります。 もう 1 つは例えば、「剛性が無い」とフィードバックしたら剛性を上げてきた場合です。そ うすると相反する要件が必ずあるので、相反する要件を必ずここでチェックします。
- ・直せと言ったのに直さない。これは判断基準そのものが違っているのではないか。
- ・直せと言ったのに、まだごめんなさい直していません。 フィードバックの織り込み予定時期が未だ不明です。一番怖いのは、一旦直すつもりだった のに、見落しているとか、忘れているとか、そういうこともあるのではないか。
- ・変わったかどうかわからないから、しょうがないからチェックしている。 前回から変わっていないという確証を誰も保証してくれない。これが問題なのではないか。

システム要件

- 1. 設計変更情報の把握ができること
 - ◆ 変更箇所の把握
 - ◆ 変更理由の把握
- 2. 要件チェック項目が容易に分かること
 - ◆ 何をチェックすべきか自動的に指示
 - ◆ 関連する要件項目の類推をナビゲーション
- 3. 要件チェック作業が容易であること
 - ◆ 無駄なチェック作業をさせない
 - ◆ チェック作業の自動化
 - ◆ チェック作業の支援

図 I-2-49 システム要件

個々の課題を私なりに分析してみて、未だシステム要件のレベルに落ちていませんけれども、こ ういった要望があるのではないか。

- ・設計変更情報の把握ができること。何を変えたかはもちろんですが、変えた理由をきちんと把握できる。
- ・要件チェック項目が容易に分かること。 製造要件のチェック項目が容易に分かること。例えば、何をチェックすべきか自動的にコン ピュータが指示してくる。若しくは関連する製造要件項目の類推をナビゲーションする。
- ・要件チェック作業が容易であること。 やらなくてもよいチェック作業をさせない、若しくはチェック作業が自動化されている。せ めて、チェック作業を支援する。

次に、これらを少しブレークダウンしました。

1. 設計変更情報の把握ができること

◆ 変更箇所の把握

- A) 変更箇所の情報が容易に確実に入手できる
 - 設計者は、生産技術者が要求する変更情報を伝えてくれること
 - 適切な表現で確実に伝えてくれる、適切なしくみ↓が存在する。
 - » 業務プロセス・・・・・守れば自分の利益になるプロセスルール
 - » システム・・・・・・・やらされ感がなく、自然に使える
- B) 適切な変更情報が伝わらなくとも、何処が変更されているか自動的に確実に把握できる
 - 新旧での形状の違いや注記表記の違いだけは、完全自動抽出。
 - その他の情報は、色々な情報伝達媒体を用いた人間系で。
- C) 完全に自動でなくとも、変更箇所の把握を容易にする
 - 新旧を繰り返し映すとか(サブリミナル効果)

◆ 変更理由の把握

- A) 変更理由を容易に確実に入手できる
 - 設計変更履歴を完全にトラッキングできるしくみ?
 - 変更の理由←その原因となった事象←そのまた原因となった・・・・
- B) 変更箇所や新旧の相違から類推を容易にする

図 I-2-50 設計変更情報の把握ができること

①変更箇所の把握

A)変更箇所の情報が容易に確実に入手できる(理想)。

設計者は、生産技術者が要求する変更情報を伝えてくれる。適切な表現で確実に伝えてくれる、適切なしくみが存在する。

- ー業務プロセス的にみれば、設計者がそのルールを守れば自分の利益になるプロセスルール。 逆に、変わったと言う情報を渡し忘れると、何かの罰則がある。
- ーシステム的には、設計者にとってやらされ感がなく、コンピュータが把握してくれて自然 に使える仕組みができないか。

B)適切な変更情報が伝わらなくとも、何処が変更されているか自動的に確実に把握できる(少し 現実的)。

設計者は細かな情報を伝えないだろう、という前提に立ちます。

新旧の形状だけの違いや注記/表記の違いは、システムが完全に自動的に抽出して欲しい。 その他の細かな仕様や属性等は、今までと同じような人間系の情報伝達のやり方でカバーす る。

C)完全に自動でなくとも、何かコンピュータがナビゲーションせよ。 変更箇所の把握を容易にする。

②変更理由の把握

A)変更理由を容易に確実に入手できる(理想)。

設計変更履歴を完全にトラッキングできる仕組みがあるべきではないか。

ただこれを考えてみますと、変更された理由は必ずあるが、その原因となった事象があって、 その原因となった理由がまたある。

「リブを追加しました。これが変更理由です。」、「ちょっと待て、そのリブを追加した理由 は何なのだ。強度不足だったのか、その前に何故強度不足になったのか、それは射出成型の 充填不良があったからか。」と、いっぱい理由がある。

設計者は最後のリブを追加したしか書いていない。これでは何をチェックしたらよいかわからない。

B)変更箇所や新旧の相違から類推を容易にする(現実的)。

例えば、「この部品でここに穴を開けた、多分これは軽量化用の穴だろう。」と、設計者が書いてなくても、その理由の類推をそこそこ助けてくれるようなことができればよい。

2. 要件チェック項目が容易に分かること

- ◆ 何をチェックすべきか自動的に指示
 - チェックすべき項目が、優先順位に沿って自動的に指示される
 - 対象となる部品、材質、要求品質を考慮して
 - 過去のチェック項目を考慮して
 - 過去の問題やノウハウを考慮して
 - 今回の変更箇所や変更理由を考慮して
 - どのレベルまでチェックすると良いのか指示してくれる
 - » チェック項目と、生産不具合率の関連を自動で認識している
 - » コンピュータ: "この項目のチェックまでで、この部品の不良率はO. 05% と予測されます"
- ◆ 関連する要件項目の類推をナビゲーション
 - 部品や部位から、関連するチェック項目を類推する
 - 蓄積された過去の問題やノウハウからチェック項目を類推する
 - 設計変更理由から、関連するチェック項目を類推する
 - 設計変更理由に相反するチェック項目を類推する
 - 重量軽減←→剛性不足

図 I-2-51 要件チェック項目が容易に分かること

生産技術者が新しい設計データが来て、例えばドアのデータが来て、何をチェックすべきかそれがわからない。

それを、

- ・何をチェックすべきかを自動的にコンピュータが指示してくれるとよい。 特にチェックすべき項目が、優先順位に沿って自動的に指示されることが望ましい。 コンピュータがどうやって自動的に指示してくれるのか。
 - 一対象となる部品、材質、要求品質を考慮して決めて下さい。
 - -過去にこういった部品はどんなチェック項目を実施したのか、そういったことも考慮して下さい。
 - 一過去に起こした不具合や過去のいろいろなノウハウを考慮して決めて下さい。
 - -今回設計変更された場所や理由も考慮して下さい。

このようにコンピュータが考慮して、チェック項目を決めてくれること。

かつ、チェック項目は切りがないので、例えば、一番から百番までのチェック項目をチェックすると、「生産不具合率は既に 99.99%。品質不良はありません。」とコンピュータがナビゲーションしてくれると一番ハッピーです。

これが、自動的に指示してくれる仕組みが成り立った世界のイメージです。

それに対して現実的なものというのは、

・関連する要件項目の類推をナビゲーションしてくれるぐらいでしょう。

例えば、

- 一部品とか部位、場所から、関連するチェック項目をあくまでも類推する。ですから確定ではない。
- 蓄積された過去の問題やノウハウからチェック項目を類推する。
- 一設計の変更の理由から、関連するチェック項目を類推する。
- -設計の変更理由に相反するチェック項目を類推する。例えば重量軽減であれば、剛性不足 になっているだろうなという懸念が発生します。

そういったところから、あくまでも類推する、というのが現実的な答えだなと思っています。

3. 要件チェック作業が容易であること

要件チェック作業工数の削減ができること

- ◆ 無駄なチェック作業をさせない
 - 重複チェックをさせない
 - 設計変更されていない情報の確実な把握
- ◆ チェック作業の自動化
 - 個別の製造要件のチェック作業を自動で実施するしくみ
 - チェック結果をもとに、自動的に合否判断するしくみ
- ◆ チェック作業の支援
 - ・ チェック作業の短縮化
 - チェック作業の支援
 - » 注目すべき箇所を瞬時にクローズアップする

図 I- 2-52 要件チェック作業が容易であること

何をチェックしなければいけないか分かったとして、その製造要件のチェックを簡単にできないか、作業工数を削減できないかです。

- ・無駄なチェックをさせない。 重複したチェックをさせない。設計変更が起こってないものは何かを確実に把握する必要が あります。
- ・チェック作業をできるだけ自動化する。

個別の製造要件のチェック作業を自動的にチェックします。

チェックしただけでは駄目で、そのチェックの結果を基に、コンピュータがこれでよいかど うかを合否判定して欲しい。

チェックと合否の判定、これがもし出来れば、例えば生産技術者でなくても設計側でもできるかもしれない。

・それもできなければ、ともかくチェック作業を支援する。 これは現実に個々のチェック作業で取り組んでいる話だと思いますが、できるだけ時間を短縮する。 チェック作業を支援するやり方として、例えば、ドアの設計において、「水漏れに関する場所」というと、コンピュータがウェザーストリップ周りをクローズアップしてくれるとか。 そういったことをやると作業が楽になるのではないかと思います。

システム屋の視点からの考察

- ◆ 設計情報と製造要件を関連付けできるか?
 - ▶ 製造要件の対象物や対象作業を具体的に表現するためには、製造要件の膨大な分類定義が必要となり、困難では?
- ◆ 製造要件を自動チェックできるか?
 - » チェック作業をコンピュータが支援し、個々の要件のチェック作業の自動化を進めることは出来ても、判断を自動的に行うことは困難では?

図 I-2-53 システム屋の視点からの考察

- こういった要件があるが、本当にできるのかシステム屋の視点から考えてみました。
- ・設計情報と製造要件を関連付けできるか。 製造要件の対象物や対象作業を具体的に表現するためには、製造要件の膨大な分類定義が必要となり、これは難しいのではないか、というのが私の今の判断です。
- ・製造要件を自動チェックできるか。
 - チェック作業をコンピュータが支援し、個々の要件のチェック作業の自動化を進めることは 出来ても、判断を自動的に行うことは困難ではないかということです。

設計情報と製造要件を関連付けできるか?

関連付け易い製造要件

- 対象物が限定されている
 - トランク、フード、左右前後の ドア、バックドアにおいて。
- ・ 作業が限定されている
 - ボルト穴の座面加工において
- 判断要素が限定されている
 - 冶具が接触する断面積が・・
- 000000

関連付け難い製造要件

- 対象物が限定されていない
 - 開きものにおいて
 - Fuel-Lidは該当するのか?
- 作業が限定されていない
 - 機械加工において
- 判断要素が限定されていない
 - 冶具の取り付けに支障なき・・
- . 000000



具体的表現を多くするには、膨大な分類定義が必要となる



抽象的表現が多くなると、関連性の確度が下がる

"検索エンジン"を用いた検索支援が、現実的な解決策ではないか? (但し、キーワード⇒図や画像⇒3D検索 などに期待)

図 I-2-54 設計情報と製造要件を関連付けできるか

関連付けしやすい製造要件と関連付けしにくい製造要件を二つ並べてみました。

対象物が限定されている。

「トランクとフードとドアとバックドア...。」、これについて要件をチェックしようという わけです。

これに対して、

対象物が限定されていない。

「『開きもの』は全てこの要件をチェックする。」と言われると、燃料入れるときの蓋は開き ものですが、多分、意図は違います。

例えば、水密の基準だとか、人にぶつかって開けたときに危険だとか、指を挟むとか、そういうことだったりするので、本当は該当しないと判断しないといけないのに、『開きもの』というと Fuel-Lid も含まれてしまいます。

ですからあくまでも対象物が限定されていれば、どんどんコンピュータは判断しやすくなります。それから、

作業が限定されている。

例えば、「ボルト穴の座面の加工において」と、ここまで書いてあると相当具体性が出てきます。

ところが、

・作業が限定されていない。

「機械加工においては」と書かれていると、何のためのどこの機械加工かわかりません。

・判断する要素が限定されている。

何をもってYesかNoかの判断をすればいいのか。例えば、「冶具が接触するときの製品にあたる断面積がこれ以下であること」とか、「傷をつけずに」と書かれていると、断面積が判断要素です。

ところが、

・判断要素が限定されていない場合には、

「冶具の取り付けに支障なきこと」よくこういうことが書かれています。これでは何をもって判断するのかわからない。

図の左側に寄せて行けば行くほど、コンピュータは関連付けしやすくなる。ただ、具体的表現を どんどん多くしていくと、膨大な分類の定義が必要となります。

例えば、対象物をどんどん限定していくと、いろいろな分類をするわけですね。その分類という 定義が膨大になっていくのではないでしょうか。

一方で右側のようにまとめると、抽象度が高くなってしまうので、本当に関連したかどうかの確度が下がります。

このように相反しているので、現実的な解決策は、Google でも何でもいい、検索エンジンを用いた検索支援が現実的な解決策ではないか。

単純なキーワード検索だけではなくて、図とか画像の類似画像検索だとか、立体の 3D 検索だと かが言われているので、こういった進歩に期待できるかもしれない。

ただこういった関連付けは、非常に難しいですね。

今、当然、中間的なものですね。この対象物に関しては絶対にチェックしないといけないという ものもあれば、もしかすると関係しているかもしれない、その曖昧なものもすごく多い。

設計情報と製造要件を自動チェックできるか?

自動チェックし易い製造要件

- 判断基準が定量的である
 - 加工工程は3回以内
- 明確な数値で示されている
 - 5mm以上のクリアランス
- 判断の重要度が示されている
 - 必須、〇〇要件に優先する

自動チェックし難い製造要件

- 判断基準が定性的である
 - 加工工程は少ないこと
- 曖昧な表現で示されている
 - 指が入る程度のクリアランス
 - 判断の重要度が不明
 - 他の要件と相反したら?

さらに困難な課題は、現実の製品は妥協の産物である故に、個々の評価項目に おける判断基準は、フェーズ後半になるほど杓子定規な定量評価が困難になり、 人の判断の介入が多くなっていると想像する。

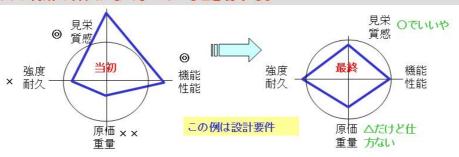


図 I-2-55 設計情報と製造要件を自動チェックできるか

自動チェックしやすい製造要件としにくい製造要件に分けてみました。 まず、

- ・判断基準が定量的である。 例えば、「加工工程は3回以内にしないといけない」。
- 判断基準が定性的である。「少ない方がいい」という記述。
- ・明確な数値で示されている。 「5mm以上のクリアランス」、これはコンピュータが識別できます。 ところが、
- ・曖昧な表現で示されている。 設計者にとっては意図が分かるからこれは意外と多い。例えば、「指が入る程度のクリアランス」。

・判断の重要度が示されている。 この要件は絶対・必須だと、それから、この要件は別のこの要件に優先する。そういうこと が示されている。

一方、

・判断の重要度が不明。他の要件とコンフリクトしたらどうするか一切書かれていない。

こういう製造要件においても、コンピュータが把握しやすい/しにくい、に差がある。

さらに困難な課題は「現実の製品は妥協の産物である。すべて理想的なものはあり得ない。故に、個々の評価項目における判断基準は、フェーズ後半になるほど杓子定規な定量評価が困難になる。」ことではないか。最終的に人の判断の介入が多くなっていると、私は想像しています。

製造要件の具体例が挙げられなくて恐縮なのですが、設計要件にしましょう。

例えば初期段階は、凄く見栄えの良い、質感の良いものを作った。機能も性能も凄く良く、誰が見ても二重丸、しかし強度が足らない、コストが大変に高い、それは誰が見てもだめです。これが何回も設計変更されていくうちに、「見栄えはこのぐらいで」という判断が入り、強度・耐久はクリアな判断ができるかもしれませんが、原価・重量になると、大概の場合、「△だけどいいか」という妥協をします。

機能・性能というのも定量的に判断できるものとできないもの、いろいろあると思いますが、ここも大体微妙な線に入っている。最終的にできる製品というのは、こういうものではないでしょうか。

コンピュータは、一番こういう線上の判断ができないですから、なかなか判断というのは、難しいのではないかと思いました。

第Ⅱ編 データ品質規格とその活用

1. データ品質の向上と規格の活用

1.1 データ品質向上の課題

大高 晢彦 委員長

製品開発への3D CAD システムを中核とした IT システム活用の普及がフロントローディングを可能にした。つまり、設計の早い段階からの3D データの共有を実現し、生産可能性など後工程の問題点をより早い段階から検討することを容易にした。フロントローディングの推進が製品開発の期間短縮に大きく貢献している。これはデジタル化推進の'光'の部分と言える。一方デジタル化推進の'影'の部分としては、第 I 編で議論されている設計品質不良や設計への過負荷の問題の他に、最近とみに顕在化している問題として製品データ品質(PDQ: Product Data Quality)問題がある。これはデータの受け手側が自身の作業を開始する前に受領データの品質を改良せざるを得ない、あるいは、改良データを発注者に要求せざるを得ないという問題である。デジタルデータは CAD データ、PDM データ、部品表データ、生産現場データなど多岐に渡り、それぞれデータ品質問題と無縁ではないが、ここでは、中核のデータであり不良品質が大きな経済ロスを生む CAD データの中の形状データに焦点を絞って議論する。

(社)日本自動車工業会(JAMA)によると、自動車メーカと部品メーカとの間の CAD データ交換時のトラブルは、最低でも年間約25万件発生し、損失金額は年間約71億円、損失リードタイムは1件あたり約1.5日と、非常に大きな経済ロスを招いているという(数値は、2001年8月試算【JAMA】)。自動車産業全体ではこの数倍の件数、経済ロスになろう。

このような形状データ品質問題の主因は、

- CAD システムごとに形状などの製品データの内部表現法(曲面式など)、表現の限界(周期性、次数など)や維持できる数値精度が異なること
- データ品質に関する要求は工程に依存する部分がある(生産技術要件の形状への織り込みなど)が、工程依存性の解決に関する合意が無いこと
- CAD システムの頑健性不足
- ユーザの操作ミス

などであるが、問題を複雑にしているのは、データを作っている本人がデータの品質の不適切さに気付かず、データを渡して始めて品質不良が判明することが多いという事情である。より本質的な課題としては、許容品質は対象製品の規模や要求設計精度に依存し、単純な基準では実用に供しないということがある。更に数年前まで表立って問題にされなかった政治的理由としては、データ品質はユーザもベンダも公表を避けたい問題であるということも挙げられよう。つまり製造業は自社のIT活用でデータ品質問題があるとは公表しにくいし、ITシステムベンダも自社製品がデータ品質問題を起こすというネガティブキャンペーンは避けたい。

このデータ品質問題に最初に光を当てたのは自動車産業であった。自動車産業はボデーメーカ を頂点に数百社がデータを媒体として関与しており、データ品質不良の被害が最も大きい産業で あることが、重い腰を上げる原因となった。日本の JAMA、(社)日本自動車部品工業会 (JAPIA)、 米国 AIAG (Automotive Industry Action Group)、仏 GALIA (Groupement pour l'Amélioration des Liaisons dans l'Industrie Automobile)、独 VDA(Verband der Automobilindustrie)、北欧 ODETT (Organisation for Data Exchange by Tele-Transmission in Europe) などが連携して データ品質評価の尺度の統一に務め、その成果である SASIG (Strategic Automotive product data Standards Industry Group) PDQ Guidelines V2.1 Rev1 は 2006 年 ISO(国際標準化機 構)/PAS(Publicly Available Specification) 26183:2006(【ISO/PAS】)としても承認されてい る。自動車産業のこの Guidelines 作成の目的は、データの受け手の再作業の撲滅による製品開発 の効率化という現実的なものである。従ってこの Guidelines は、実務で問題になるデータ品質不 具合の体系的整理の結果である。その内容は形状データを中心に CAE データ、一部の管理デー タなどに渡る。これまで隠された問題に焦点を当てたという意味で極めて意義が大きいが、'良い データ品質とは何か'を積極的に規定したものではないこと、基準が全て自然言語で書かれてい るので解釈の一意性確保が困難/コンピュータプログラムでの実装と距離があること、市販 CAD の今日現在の実力に合わせた妥協があること、などの課題も併せ持っている。

製品開発への IT 投資の効果を抜本的に高めるためには、製品データの信頼性を向上させることが不可欠である。'製品データの信頼性を向上させる'ためには製品データが設計者の意図や要求精度を十分満足しており、業界規格、社内規格他の拘束条件も満足するというだけでなく、ここで取り上げているデータ品質問題を本質的に解決して'データ品質が保証されたデジタル化'を実現する必要がある。その骨格となる品質基準は以下の要件を満たすことが求められよう。

- データ授受の種々の場面でデータの送り手の品質要求が表現でき、データの受け手が注目 する品質に関する懸念も十分含まれていること、また拡張性も豊かなこと
- 品質項目が曖昧さ無く一意に理解できること
- 品質検査の指標が示されていること
- 実務への展開の容易性が考慮されていること
- IT システムのデータ品質の観点での改良に資する内容であること

データ品質問題は産業依存性は少ないから、製造業全般に適用可能な品質項目の表現、品質検査結果の表現が望まれる。ISO 10303-59 (Part 59) は3D形状データに関してこの要件を満たすために日本主導で開発が進められている国際規格である。Part 59 の開発は SASIG PDQ Guidelines との整合性を考慮し、それが持つ課題の解決も念頭に置きながら検討が進められている。

データ品質の不良は放置すると別な不良を招くという具合に伝播する。伝播してしまったあと

では、何が不良の真因かを捉えるのは極めて困難になる。従って、現状のスナップショット的な 品質検査でなく、ユーザの1操作ごとに裏で品質検査が走り、品質上問題があればユーザに知ら せて対応を促す仕組みが理想であろう。

本委員会では Part 59 の産業界への普及を加速するために必要な諸観点からの検討を実施した。 'データ品質が保証されたデジタル化'実現のための活動がユーザ、IT ベンダ、研究者の知恵を 集めた活動として強力に推進されることが望まれる。

参考文献

【JAMA】JAMA ホームページ > クルマと情報化 > PDQ (Product Data Quality:モデルデータ品質) に関わる活動(http://www.jama.or.jp/cgi-bin/pdq/download_pdq.cgi)、JAMA、2006

【ISO/PAS】ISO ホームページ > Catalogue online > ISO/PAS 26183:2006 (http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=43436)、ISO、2006

1.2 データ品質向上の要件

相馬 淳人 副委員長

本節は"2 規格の活用法と実務適用の要件"の報告の要旨を示すと共に、それらが"1.1 データ品質向上の課題"とどのように関係するかを述べたものである。

ISO 10303 は、製品情報のコンピュータで解釈できる表現と製品データの交換に関する国際規格である。通称として STEP(Standard for the Exchange of Product model data)とも呼ばれる。この1つの規格(Part)として、現在、日本主導で国際規格 ISO 10303-59(Part 59)の開発が進められており、製品データ品質の表現を定義し利用方法を提案している。"2.1 製品データ品質規格 Part 59"では、その概要を説明している。

製品データ品質に関しては SASIG から PDQ Guidelines が先行して発行され、ISO においても ISO/PAS 26183:2006 として承認されていることは、1.1 において述べられたとおりである。 Part 59 もまた SASIG PDQ Guidelines と同じくデータ品質問題の解決を目指して開発が進められてきた。形状データ品質を考える際、最初に直面する最大の問題は「良い品質の形状データ」の定義である。この問題に対しては、未だに明快な定義は与えられていないと言って良いだろう。 そこで両者とも、形状品質項目として、実務で問題となる「不正な性質」、「不適切な性質」を列挙しそれを体系化するというアプローチをとっている。

現在、概念設計、詳細設計、生産準備等々の様々な業務プロセスにおいて CAD、CAM、CAE といった多種多様なツールが使われており、それぞれに業務上、あるいはシステム上の要件が存在する。また、それらの要件の中には、相矛盾するものも存在する。細かい形状が望ましくない CAE 用の要件と、非常に高い精度を要求する CAM 用の要件がこの矛盾の典型的な例である。このため、製品のライフサイクル全てに汎用的に通用する「良い品質」を定義することは非常に困難であり、実務上の諸問題を体系化することが現実的な解決策となるのである。

また、SASIG PDQ Guidelines と Part 59 は両者ともに、「悪い品質のデータ」を直接定義するものではないことも強調したい。形状に関して、SASIG では 64 個、Part 59 においては 85 個 (2007 年 3 月現在の国際規格原案) の品質項目が定義されているが、いずれにおいても、ユーザはこれらの中から適切な項目とそれに対応する適切な閾値を設定し、個々の目的に適した「悪い品質のデータ」を定義することが想定されている。つまり、SASIG や ISO が「良い品質のデータ」を定義しユーザを縛るために存在しているのではなく、形状データ品質を定義するための道具、あるいは共通言語という位置付けのものなのである。

また、ISO 10303 の側から見ると、Part 59 は別の意義を持っている。ISO 10303 はこれまで、 誤差を考慮しない理想的な形状表現しか特たなかった規格群であったが、Part 59 は、その中に 数値誤差を始めとする品質の概念を初めて導入する、非常に重要な意味を持った規格なのである。 開発の背景および規格の位置付けについてのより詳細な記述は、"2.1.1 開発の背景"および"2.1.2 位置付け"を参照されたい。

このような形状データ品質の要件およびそれに対する検査結果を記述するため、Part 59では、

ISO 10303 の開発を通じて培われてきた形式仕様記述言語 (EXPRESS) を用いている。その詳細については "2.1.3 概念と構成" を参照されたい。国際規格を目指すため記述から曖昧さが排除され明確であるだけでなく、形状以外の品質への拡張性、品質検査測定の外部仕様明確化とソフトウェアベンダのノウハウである形状検査アルゴリズムの隠蔽の両立などが実現されている。

"2.1.4 利用シナリオ"においては、本規格が想定している利用シナリオごとに、具体的な例を示した。利用シナリオとしては、データ品質の要求、宣言、保証、および改善の4ケースを挙げた。「要求」は、具体的なデータのやり取りに先立って当事者間で品質のルールを定めるケースを、「宣言」はデータ作成者が自分の作成するデータ品質を宣言するケースを想定しており、これらの場合には選択された幾つかの品質項目とそれに対応する閾値が、個々の製品データとは独立したデータとしてやりとりされる。「保証」では品質保証部門等が個々のデータを検査した結果が製品データに付加される。ここでは検査の合否、不正な要素の数などの情報が結果に含まれる。最後に「改善」のケースにおいては、「保証」のケースに加えて、検査で不合格となった製品データを修正するために必要な、個々の要素レベルの情報まで結果として含まれる。これら4つの利用シナリオは実務上も十分汎用的であり、Part 59 の実務への展開が容易に可能であることを示唆している。

さらに "2.1.5 インスタンス例" においては、規格の使い方を理解し易くするために、2 つの 品質項目について各利用シナリオで生成されるインスタンスを、今回考案した図式記法で示した。

Part 59 は、ISO 10303 において、汎用的な基盤技術である統合リソースとして開発が進められている。統合リソースとは、ISO 10303 の様々な応用規格から共通で使用可能な、汎用の製品モデルである。統合リソースは、何らかの応用規格を通してのみ実際に使用することが可能である。"2.2 Part 59 の活用法と拡張課題"では、Part 59 を実用に供するために考え得るいくつかの実装手段について技術的な検討を行った。

"2.2.1 既存 AP への組み込み方法"では、既存の応用規格(AP: Application Protocol)を改訂し、Part 59 を組み込む手法について検討している。この手法によると、既存の ISO 10303 の開発手法、プロセスの枠組みの中で Part 59 の実務展開が実現可能である反面、形状を扱う AP ごとに改訂のための長いプロセスを経なければならないという短所もある。また、この項には、エラーを持つ AP 203 ファイルに対する Part 59 の検査結果の具体例を、不具合形状の画像と共に示した。この手法に限らず、Part 59 全体の理解の一助となると幸いである。

"2.2.2 汎用的な組み込み方法の技術的検討"では、Part 59 を実用に供することのみを目的とした新たな仕組みを検討し、"PDQ シェル"と名づけた。"PDQ シェル"では、既存の AP は変更せず、そのデータを外側から参照することで、Part 59 の品質情報を付加することが可能になる。これは、ISO 10303 の AP では形状表現に共通の規格(Part 42)を用いているために可能である。従来に無い手法ではあるが、ISO 10303 で 3 次元形状を扱う AP の全てに対して Part 59 の品質規格を適用可能になることが、この"PDQ シェル"最大の特長である。

上記 2 項では、品質検査の対象が STEP ファイルに限定されていたが、実務展開のためには

CAD のネイティブファイルまで扱えることが重要であると考えられる。そのため、"PDQ シェル"のキーテクノロジである外部参照の仕組みを拡張することによって、非STEPファイルにまで対象を広げることを検討した。詳細は"2.2.3 非STEPファイルに適用するための検討"の考察を参照されたい。

今後は、実務展開に向けて、難易度と実現時期等を考慮しながらいずれかの方法をとって推進 することが望まれる。

最後に、"2.3 Part 59 の実務適用のための要件"では、形状データ品質問題の課題を再び振り返り、そこで見込まれる Part 59 の効果と残課題について記した。

Part 59の最大の効果は、形状品質について、ユーザ、PDQ ツールベンダ、CAD ベンダの間で利用可能な、曖昧さを排除した正確な共通言語を確立することであろう。SASIG PDQ Guidelines はあるものの、現状では、理解の一意性が保証された共通言語は存在しておらず、そのためにコミュニケーションや改善活動が大きく阻害されている。

しかし、規格を開発するだけでは品質問題は解決できない。そもそも形状品質問題を困難にしていた、品質問題を埋め込む部門(データ作成者)とその影響を被る部門(データ利用者)の不一致という構造的問題は、品質改善の活動においても改善作業を行う部門(データ作成者)と受益者(データ利用者)の不一致という形にそのまま移行しており、いずれにしてもユーザを巻き込んだ部門横断的な活動が要求される。

また、根本的な品質問題の解決には、CAD ベンダの協力も重要である。特に、基本的に最終形状のみを評価する現在の形状品質基準による品質改善を、現在主流であるフィーチャベースの CAD システムの上で推進するためには、ユーザおよび PDQ ツールベンダに加えて CAD ベンダのコミットが必須であろう。

ユーザ、CAD ベンダを巻き込んで形状データ品質の改善を進めていくために、Part 59 を基に したコミュニケーションは非常に有用であると思われるが、単に規格を開発するだけでなく、実 務で使われ効果を生み出す段階までを視野に入れた活動が今後も望まれる。

2. 規格の活用法と実務適用の要件

本章では、Part 59 の概要を説明し、製品データ品質規格として活用するための技術課題を検 討する。さらに、実務適用させるための要件について検討する。

2.1 製品データ品質規格 Part 59

2.1.1 開発の背景

ISO 10303 は、製品情報のコンピュータで解釈できる表現と製品データの交換について取り扱っており、多くの規格が発行されている。発行された規格には、ISO 10303-203 (Part 203) や ISO 1030-214 (Part 214) など、CAD システムで作成される 2D や 3D の製品データを対象とした規格がある。既に多くの CAD システムがこれらを実装し、利用者の現場ではこれらのシステムを利用した製品データの交換が当たり前となっている。同時に製品データ交換の場面では、製品データ品質の悪さを原因とした数々の問題点が認識されている。しかしながら、これまで発行された国際規格では製品データ品質をコンピュータ内で表せる規格は無かった。Part 59 は、製品データ品質を扱う枠組みを定義し、形状データに関しては具体的に品質項目や品質検査結果の表現を取り扱った初の規格であると言える。

Part 203 や Part 214 はユーザが実際に使用する応用規格であり、AP 203 や AP 214 とも呼ばれる。製品データを取り扱った既存 AP を改訂して Part 59 を組み込む、あるいは既存 AP は変更せずそのデータを外側から参照する新たな仕組みを実現することで、コンピュータを利用し、製品データ品質を要求したり、保証したり、検査結果を保持したり、それらの情報を第三者間で流通させることが可能となる。

2.1.2 位置付け

本項では、Part 59の位置付けについて説明する(【ISO】)。

ISO 10303 は、製品情報のコンピュータで解釈できる表現と製品データの交換に関する国際規格である。その目的は、製品のライフサイクル全般を通じた製品の中立的な表現メカニズムの提供である。このメカニズムは、中間ファイルでの交換ばかりでなく、製品データベースの実装と共有や保存の原理としても適している。

Part 59 は、ISO 10303 のリソースの一つであり、以下の一般的な製品データ品質に関する 2 つのスキーマと、形状データ品質に関する 2 つのスキーマより成る。

- 製品データ品質基準スキーマ
- 製品データ品質検査結果スキーマ
- 形状データ品質基準スキーマ
- 形状データ品質検査結果スキーマ

製品データ品質についての詳細な説明に入る前に、実際の製品の品質、製品モデルの品質、および製品モデルデータの品質に差異があることを理解頂きたい。ISO 9000 の用語定義を引用すると「製品品質とは、製品の外観、機能、そして、性能についての要求に対する満足の度合いである」と定義されている。概念的製品モデルはISO 10303 に適切に定義されているが、製品モデルの品質については、僅かしか定義されていない。製品モデルに関係付けされた許容差の情報のみが、製品モデルの品質に関わる情報と言えよう。製品モデルデータは、有限桁数の数値による概念的製品モデルの表現である。それは必然的に数値精度の問題を持っている。幾何要素間、位相要素間、幾何要素と位相要素間のわずかな不整合が、あるアプリケーションプログラムでは不適切である可能性がある。

全ての製品モデルデータは何らかの CAD システムにより作成されるわけであるが、数値表現 法や満足できる数値精度は CAD システムに依存することは周知の事実である。Part 59 では、製 品モデルデータの不正な性質、あるいは不適切な性質などのデータ品質を取り扱う。

仕様、検査や改良プロセスなど製品の品質について規定している他の ISO 規格も製品データ品質についての規定は含んでいない。現在の ISO 10303 に含まれている製品データ品質に関わる仕様は、モデルデータ全体の数値精度のみであり、極めて限られている。製品データ品質の表現について、特に 3D 形状データの品質の表現について詳細化して定義することは、品質保証されたデータ変換の実現に寄与するだけでなく、製品データを受け手側が利用できるレベルまで整えるための手直し作業を大幅に削減することに寄与しよう。

引用文献

【ISO】本項の文章は、ISO の許可を得て ISO 10303-59(2007 年 3 月現在の国際規格原案)から引用掲載している。(The texts taken from ISO 10303-59 in this paragraph are reproduced with the permission of the International Organization for Standardization, ISO.)

2.1.3 概念と構成

本項では、Part 59の概念と構成について説明する(【ISO】)。

(1) 概念

Part 59 は、図 II-2-1 に示すとおり、4 つの相互に関係のあるスキーマから構成される。

1番目のスキーマは、「product data quality criteria schema(製品データ品質基準スキーマ)」と呼ばれ、一般的な製品データを取扱い、品質項目と品質検査に関する要求をどのように表すかを定義している。2番目のスキーマは、「product data quality inspection result schema(製品データ品質検査結果スキーマ)」と呼ばれ、同じく、一般的な製品データを取扱い、品質検査結果をどのように表すかについて定義している。これらのスキーマに定義された仕様は、十分に汎用的である。従って形状データ以外のタイプの製品データ品質を将来扱う場合も有用であろう。

残りの2つのスキーマ「shape data quality criteria schema(形状データ品質基準スキーマ)」と「shape data quality inspection result schema(形状データ品質検査結果スキーマ)」は、前述の製品データ品質基準スキーマと製品データ品質検査結果スキーマの各々を形状データに特化したものである。

製品形状データの品質は、正常なデータ交換を阻害する不正なデータや不適切なデータ、あるいはその他の要因を原因とする、実務の問題項目を分類することにより定義した。これらの問題項目は、主に、基礎となる数学モデルの不適切な数値表現によって引き起こされる。低い品質の製品形状データが引き起こす既知の問題を整理することで、大きく「不正な(erroneous)データ」と「不適切な(inapt)データ」の2つに分類した。問題項目は、さらに、幾何の問題、位相の問題、幾何と位相の不整合の問題、そして、形状モデルの問題に分類した。

Part 59 における製品形状データの表現は、将来的にはネイティブな CAD データなど他の表現 も可能とすることを視野に入れているが、現在のところは ISO 10303 による表現に限定している。

各々の品質項目では、品質の検査要求についても定義している。検査要求では、検査基準がどのように検査されるべきかを自然言語で記述している。また、検査方法や検査対象要素を制御するための属性やルールも記述している。

補足:図II-2-1において、三角形で囲んだ数字は、Part 59 のスキーマ内の別の箇所から参照されていることを表す。EXPRESS-Gの形式に則っていないが、参照の線が煩雑になるのを防ぐため利用した。

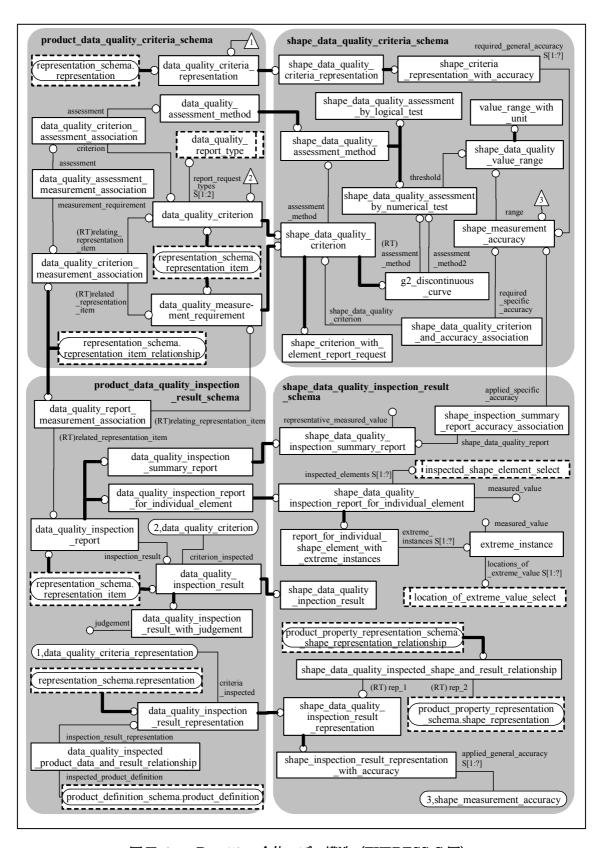


図 II-2-1 Part 59 の全体モデル構造 (EXPRESS-G 図)

(2) アルゴリズムについての考え方

Part 59 では、検査のアルゴリズムには立ち入っていない。

PDQ チェックプログラムのベンダが、検査方法について検査要求に述べられた要求を満たすようにアルゴリズムを開発することを期待している。計測のアルゴリズムは、Part 59 の対象外である。それは、アルゴリズム自身が技術系システムベンダのノウハウであり、国際規格として標準化できないからである。

(3) 閾値についての考え方

形状データ品質を数学的に評価するためには、利用者が AP を利用し定義できる閾値がキーとなる。典型的な閾値の例は、曲面をトリミングする場合の、曲線と曲面間の間隙を評価するための距離の閾値である。その距離の閾値は、曲線と曲面間隔が最も離れた場所でも、閾値より小さい値であることを要求している。適切な閾値は、例えば、製品の大きさ、設計要件、数値的な精度に関する設計システムの制限などの、多くの要素が関係している。従ってどのような閾値の値を利用するかは、各々のビジネスの場面で、ビジネスパートナとの合意に基づき、慎重に定義するべきである。

(4) 精度についての考え方

殆どの場合では、自由形状要素が存在する場合、近似計算を行う。先述の例では、曲線と曲面の最大距離を測定するためには曲線上の無限個の点に対して計算を行う必要がある。しかし、無限回の計算は不可能であるため、どんなアルゴリズムも、その中から有限個の点をサンプリングして近似計算する。近似解と厳密解(それが不明な場合も含め)の差をある期待値以下に抑えるために使われる概念として精度がある。精度に関しては、ある近似計算の解は、他のどのように細かな分割点により導き出された近似計算との差が与えられた精度より小さいならば、精度の要求を満たしているものとする。

精度は2つに分類される。1つ目は、一般的な精度であり、全ての基準に適用される。2つ目は、個別の精度であり、個々の品質項目に対して固有の値を設定できる。

補足: Part 45 には、精度の観点から実物の計測結果の仕様を定義した

「qualified_measure_schema」の規定がある。Part 45 において定義された概念とPart 59 において精度について定義された概念では、いくつかの類似点があるが、Part 59 は、数値的な精度計算の問題に焦点を当てている。そのため、Part 59 は、Part 45 には定義されていないパラメタ値の精度のような単位系を含まない精度の値を定義している。Part 59 での精度の情報は、数値的な計算精度を、対象となる製品データが定義されていない段階において、要求する際にも有効

である。以上の理由から、Part 59 では、Part 45 の qualified_measure_schema を使用していない。

(5) 品質検査結果についての考え方

形状データ品質検査結果スキーマは、指定された品質要求の定義に対する、特定の製品形状データの品質検査結果の表現を取扱う。検査結果は、製品形状データが、事前に定義された品質要求を満足するかどうかを表せる。また、どんな品質項目が、どの位、どんな形状要素において、どこで問題を引き起こしているかという、詳細な情報も含んでいる。この情報は、悪い品質の形状データに対する修正工程への活用や、製品形状データの品質チェック工程と修正工程の協業の効率化などに役立つと予測される。

(6) 利用シナリオ

Part 59 を利用するにあたり図 II- 2-1 で示されたエンティティの中で必須の情報は、品質情報の利用シナリオに依存する。Part 59 で予想している利用シナリオは、次のとおり。詳細は、2.1.4 で説明する。

① データ品質の要求

例えば発注元の会社は、発注先の会社に、所定の品質要件を満足する製品データを生成することを要求する。品質要件とは、与えた許容差より小さな微小幾何を含まないこと、製品形状の表現に寄与しない冗長な幾何を含まないこと、などである。従ってこの利用シナリオでは、閾値を伴う必要な品質項目だけが必要となる。これらの情報は、発注書類と共に送付される。

② データ品質の宣言

例えば製品データの生成者は、そのモデルが満足する品質レベルを宣言するために品質情報を使う。設計手法や利用する CAD システムにより、製品データの品質を検査無しでも明確に宣言できる。従ってこの利用シナリオでは、選択した品質項目とそれを満足する閾値が必要で、データ品質の要求の利用シナリオと同様である。これらの情報は、対応する製品データと共に送付しても良い。

③ データ品質の保証

例えば品質保証組織は、特定の製品データに対する品質検査の結果を表すのに品質情報を用いる。従ってこの利用シナリオでは、検査した品質項目、それに用いた閾値、使った計測手法、および得られた検査結果が必要である。これらの情報は、対応する製品データと共に送られる。

④ データ品質改善に使う情報の提供

例えば PDQ チェッカで品質上の不具合が検出された場合、問題データの修正などの対策が必

要になる。そのためには、どの品質項目が検出され、程度はどうかといった情報が必要である。 従ってこの利用シナリオでは、幾何エンティティインスタンスレベルの詳細な検査結果レポート を出力する。これらの情報は対応する製品データと共に送られる。

(7) スキーマ構成

Part 59 を構成するスキーマは以下のとおりである。

① 製品データ品質基準スキーマ

製品データに関する、品質項目や品質の測定要求および品質の評価方法の表現についての一般的な仕様を定義している。

② 製品データ品質検査結果スキーマ

実際の製品データに対する品質検査結果の表現についての一般的な仕様を定義している。

③ 形状データ品質基準スキーマ

計測要求や個々の品質項目の満足/不満足の判断のための閾値、および製品形状データの評価 方法と共に形状データの品質項目の表現について定義している。

④ 形状データ品質検査結果スキーマ

品質検査結果の表現について定義している。また、もし品質項目に違反する問題があれば、問題が発生した形状要素データを特定する品質項目と共に品質検査結果は表現できる。

図 II- 2-2 に、Part 59 と ISO 10303 のリソースとして定義された他のスキーマとの関係を EXPRESS-G 表記で示す。EXPRESS-G については、ISO 10303 の Part 11 の Annex D に記載されている。図 II- 2-1 に表したスキーマは、ISO 10303 のリソースから構成され、下記のリソースを利用し定義している。

date_time_schema	ISO 10303-41
measure_schema	ISO 10303-41
person_organization_schema	ISO 10303-41
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
support_resource_schema	ISO 10303-41
geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42

topology_schema ISO 10303-42 representation_schema ISO 10303-43 mathematical_functions_schema ISO 10303-50

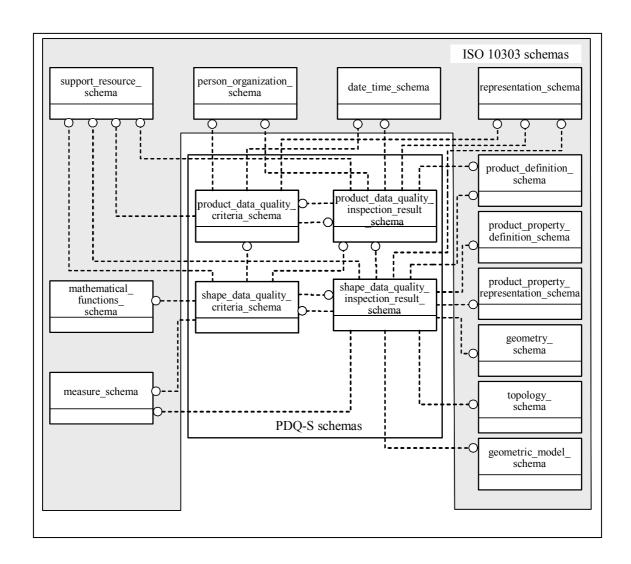


図 II-2-2 Part 59 (PDQ-S schemas) と他のリソーススキーマとの関係図

(8) 品質項目の分類

Part 59 では、製品形状データの品質は、正常なデータ交換を阻害する実務の問題項目を整理した品質項目で表されている。これら品質項目は、大きく「(1)不正な (erroneous) データ」と「(2)不適切な (inapt) データ」の2つに分類されており、各々さらに、①位相の問題、②幾何の問題、③幾何と位相の不整合の問題、そして、④形状モデルの問題に分類されている。

付録1に、品質項目の分類と一覧を示す。

引用文献

【ISO】本項(1)~(7)の文章や図は、ISO の許可を得て ISO 10303-59 (2007年3月現在の国際規格原案) から引用掲載している。(The texts and figures taken from ISO 10303-59 in this paragraph (1)-(7) are reproduced with the permission of the International Organization for Standardization, ISO.)

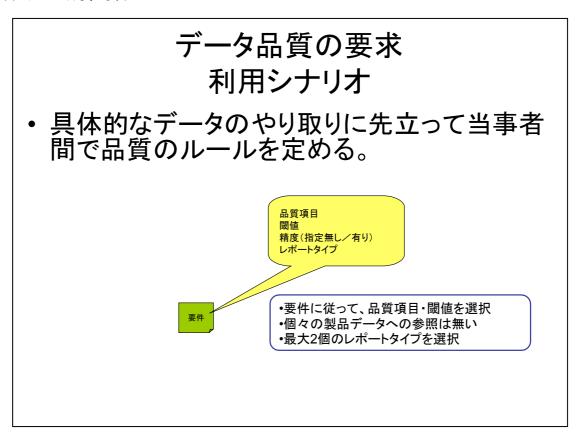
2.1.4 利用シナリオ

本項では、各利用シナリオについて、Part 59の使い方や扱う情報を示す。

- データ品質の要求
- データ品質の宣言
- データ品質の保証
- データ品質改善に使う情報の提供

図 II-2-1 の Part 59 全体モデル構造の内、必要な情報は利用シナリオによって異なる。

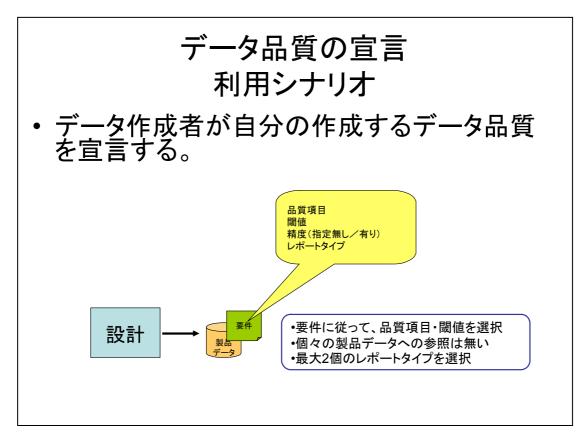
(1) データ品質の要求



図Ⅱ-2-3 データ品質の要求の利用シナリオ

具体的なデータのやり取りに先立って当事者間で品質のルールを定める。例えば製品データ発注元の会社は、所定の品質要件を満足することを発注先に要求する。この場合、閾値を伴う必要な品質項目だけが必要となる。内部計算の精度にも言及されるかもしれない。また、品質検査されることを想定し、品質結果のレポートタイプにも言及する。個々の製品データへの参照は無い。

(2) データ品質の宣言



図Ⅱ-2-4 データ品質の宣言の利用シナリオ

データ作成者が自分の作成するデータ品質を宣言する。製品データの品質を検査無しでも明確に宣言できる。Part 59で用いる品質情報は、データ品質の要求の場合と同様である。

(3) データ品質の保証

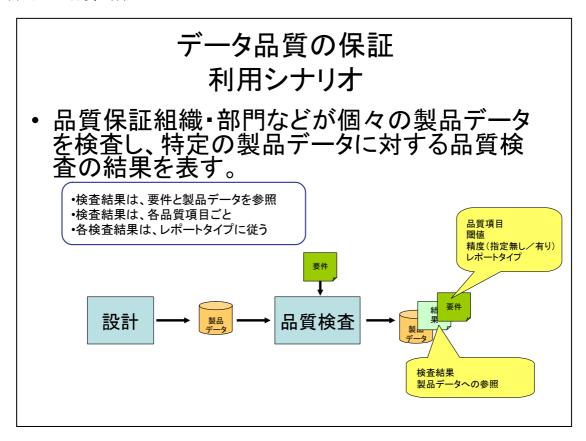


図 II-2-5 データ品質の保証の利用シナリオ

品質保証組織・部門などが個々の製品データを検査し、特定の製品データに対する品質検査の 結果を表す。検査結果は、検査総数、不合格数、不合格の代表値などで表現される。

(4) データ品質改善に使う情報の提供

データ品質改善に使う情報の提供 利用シナリオ

・検査で不合格となった製品データを修正するために必要な、個々の要素レベルの情報までを 検査結果として出力する。

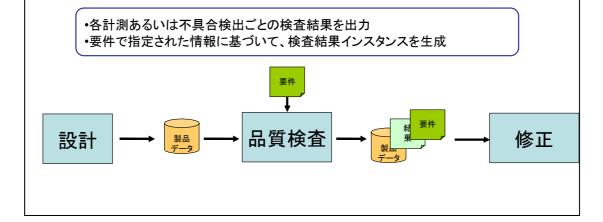


図 II-2-6 データ品質改善に使う情報の提供の利用シナリオ

検査で不合格となった製品データを修正するために必要な、個々の要素レベルの情報までを検 査結果として出力する。

これまでは PDQ チェッカと PDQ 修正ツールの間のインタフェースが無いために修正ツール側でも最初に検査が必要で、その検査結果がチェッカの検査結果と一致しないということが十分あり得た。本規格を利用すればユーザが認定した PDQ チェッカの結果がそのまま PDQ 修正ツールで使われるので、修正の効率、信頼性が上がる。

2.1.5 インスタンス例

本項では、Part 59 の 2 つの品質項目「short_length_edge (短い稜線)」と「gap_between_edge_and_base_surface (稜線と基底曲面との間隙)」についてインスタンス例を示す(【ISO】)。これらのエンティティの定義は、Part 59 の規格を参照されたい。前述の利用シナリオ、すなわち、データ品質の要求/宣言(二つの表現形式が同様となるため一緒に記述する)、データ品質の保証、およびデータ品質改善に使う情報の提供について、形状データ品質のテストデータに基づいて生成した。

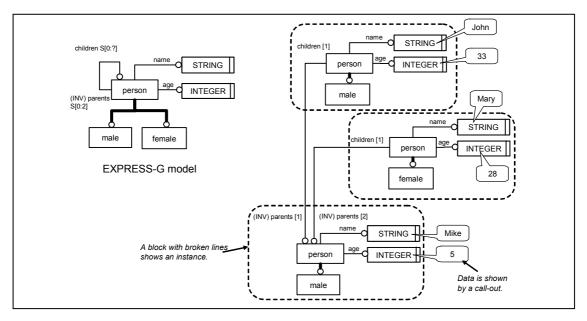
まず、理解が容易なように EXPRESS-G に基づいて作ったインスタンスの表現のための図式記法について説明する。

(1) インスタンスの図式表記

今回、インスタンスを分り易く示すため、EXPRESS-G 記法を基にインスタンス表示の図式記法を考案した。インスタンスを表すために、EXPRESS-G 記法に次の変更を加えた。

- 1つのモデルエンティティに対して複数のインスタンスを表記できる。
- 実際の値は、吹き出しを使って別に表す。
- 一続きの継承関係にあるエンティティ群に対するインスタンスは、破線で囲んで表す。本項では、この記法で示すインスタンスは、Part 59 で生成される。

図 II- 2-7 に、この記法の使用例を示す。図の左の EXPRESS-G に対して生成したインスタンスを図の右に示す。左側の EXPRESS-G モデルは人の簡単なモデルであり、右側は人の 3 つのインスタンス、John、Mary とその息子の Mike を示す。



図Ⅱ-2-7 考案した図式記法で描いたインスタンス例

(2) short_length_edge のインスタンス

1つ目の品質項目の例は short_length_edge である。この品質項目が要件に指定されると、PDQ チェッカは対象形状データを検査し、与えられた閾値より長さの短い edge を検出する。短いエッジの典型例を図 II-2-8 に示す。

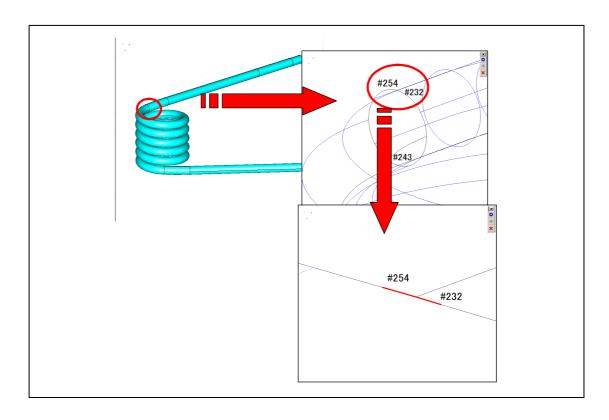


図 II-2-8 short_length_edge の典型例【エリジオン】

① short_length_edge の要求/宣言

Part 59 でこの利用シナリオの品質情報を生成するには、要件を構成する一組の品質項目を選択する。各品質項目に対して閾値の値と、必要であれば精度を設定する。個別の製品データは参照しない。必要なレポートの種類を示すために、2 つまでの data_quality_report_type を設定できる。

この利用シナリオに対して想定した例は、次のとおりである。

- short_length_edge を検出する閾値は、0.01mm とする。
- 2 種類のレポートを要求する。すなわち、検査数と検出インスタンス数を示す概要レポート、および閾値より短い長さの edge に関するレポートである。

生成されるインスタンス例を図 II- 2-9 に示す。mathematical_functions_schema のインスタンス以外に 4 つのインスタンスが生成されることがわかる。

注:あるデータ品質に関わる要件を表現するために、複数の品質項目が必要な場合は(通常そうであるが)、図中右の shape_data_quality_criteria_representation のインスタンスは、複数のitems を持ち、それぞれが要件に対応した個別の shape_data_quality_criterion のインスタンスである。

さらに、検査の精度の指定が必要な場合には、そのためのインスタンスを図 II-2-10 のように 生成する。この例では、汎用の精度を 10^{-3} mm、品質項目個別の精度を 10^{-5} mm としている。

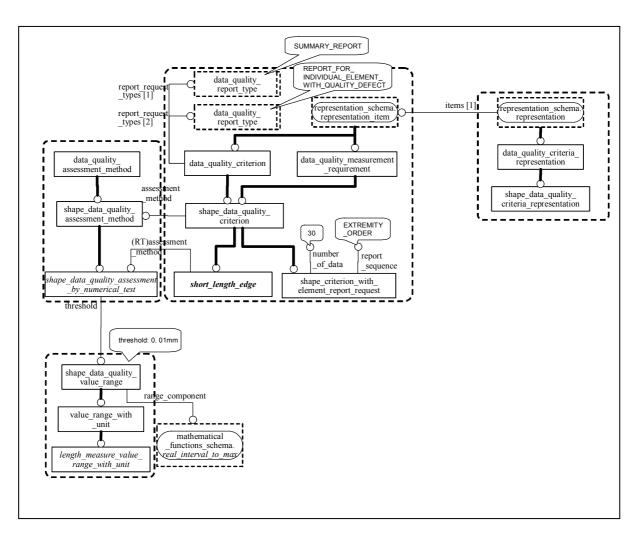


図 II- 2-9 short_length_edge の要求/宣言(精度指定無し)で用いる品質情報のインスタンス 例

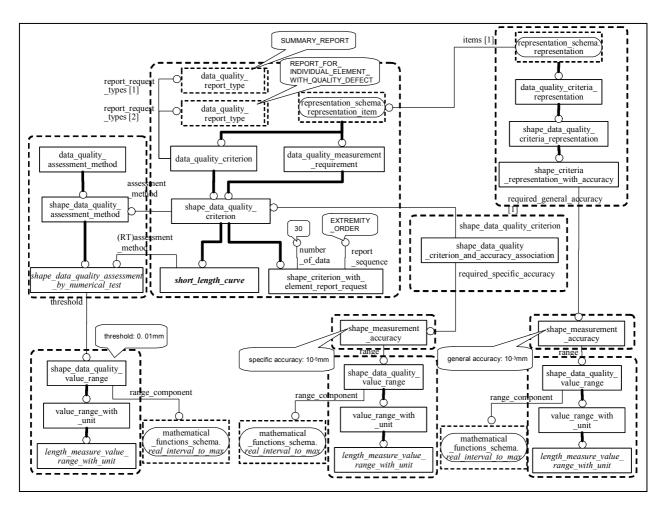


図 II- 2-10 short_length_edge の要求/宣言 (精度指定付き) で用いる品質情報のインスタンス例

② short_length_edge の保証

Part 59 でこの利用シナリオの品質情報を生成するには、要件と検査対象の製品データとを参照する検査結果を作る。検査結果は、指定された品質項目ごとに結果を生成する。各検査結果は、data_quality_report_type で指定された種類のレポートを付けて生成する。

この利用シナリオに対して想定した例は、次のとおりである。

- 要件は、10⁻⁵mm の精度で 0.01mm の閾値より短い edge を検出し、検査の概要レポートを生成する。
- 製品モデルの形状データに対して検査を行う。製品モデルの ID は#7449、形状表現 (shape_representation) の ID は#7434 である。
- 検査結果として、71 本の edge を検査し、閾値より短い 3 本の edge が見つかった。検出

された最も短い edge の長さは、0.0023mm である。

生成されるインスタンス例を図 II-2-11 と図 II-2-12 に示す。

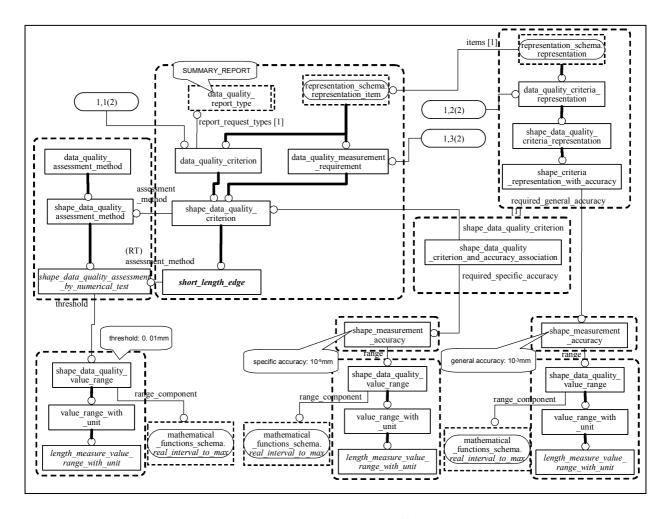


図 II-2-11 short_length_edge の保証に用いる品質情報のインスタンス例 (1/2)

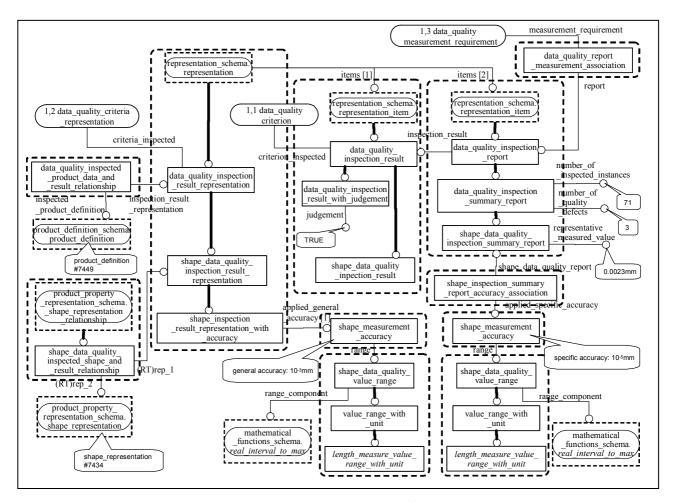


図 II-2-12 short_length_edge の保証に用いる品質情報のインスタンス例 (2/2)

③ short length edge の改善に使う情報の提供

PDQ チェッカによる short_length_edge の検出で得られる詳細情報を、後の工程で問題データの修正に用いることを前提としている。PDQ チェッカを開発する場合は、どの品質項目が満足されていないか、その程度はどのくらいかの情報を生成し、それを幾何エンティティレベルの詳細検査結果レポートとして提供する。検査結果は、計測ごとあるいは不具合検出ごとに生成する。検査結果のインスタンスは、要件で指定された情報に基づいて生成する。

この利用シナリオに対して想定した例は、次のとおりである。

- 要件は、精度 10⁻⁵mm で閾値 0.01mm より短い edge の検出である。
- この利用シナリオのためには、長さの短い edge とその計測された長さを示すレポートを生成しなければならない。レポートは、値の極端な順に最大30生成する。
- 概要レポートは、②の例と同じである。
- short_length_edge であると検出された 3本の edge の詳細情報は、次のとおりである。

- ・ edge_curve #254 長さ 0.0029mm。
- ・ edge_curve #232 長さ 0.0023mm。
- ・ edge_curve #243 長さ 0.0079mm。

生成されるインスタンスを図 II-2-13、図 II-2-14、図 II-2-15 に示す。

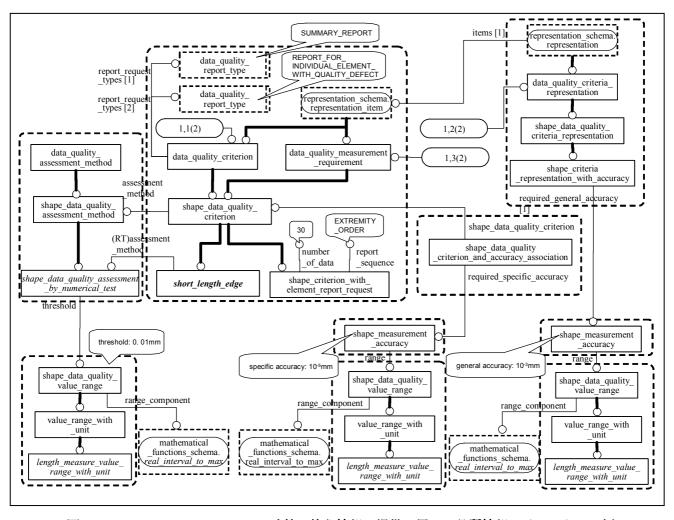


図 II-2-13 short_length_edge の改善に使う情報の提供に用いる品質情報のインスタンス例 (1/3)

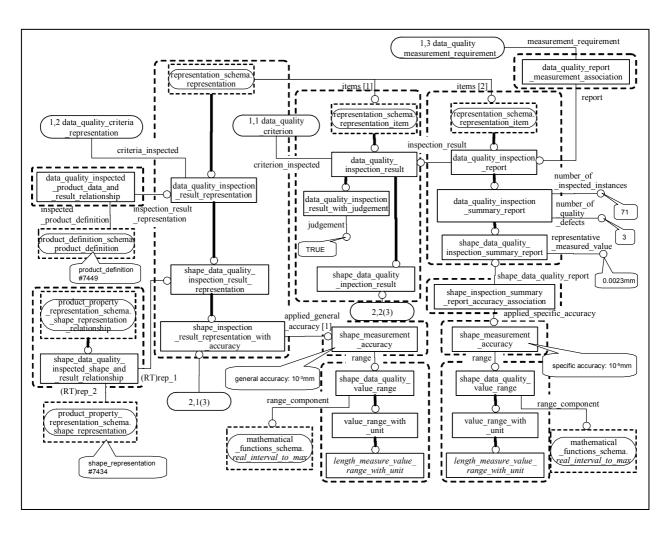


図 II- 2-14 short_length_edge の改善に使う情報の提供に用いる品質情報のインスタンス例 (2/3)

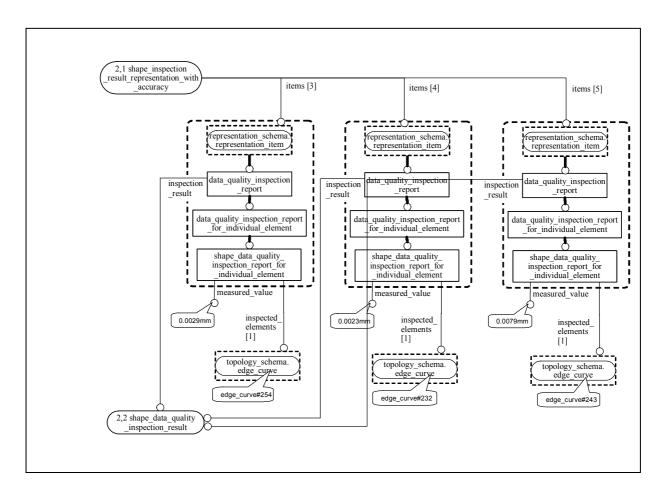


図 II- 2-15 short_length_edge の改善に使う情報の提供に用いる品質情報のインスタンス例 (3/3)

(3) gap_between_edge_and_base_surface のインスタンス

次の品質項目の例は gap_between_edge_and_base_surface である。この品質項目が要件に指定されると、PDQチェッカは、対象形状データの face_surface の境界の各 edge_curve を検査し、検査対象の edge_curve の任意の点から基底曲面への最小距離の最大値を計算し、その計算値が指定された閾値より大きな値を持つすべての edge_curve を検出する。

データ品質の要求/宣言およびデータ品質の保証については、前の品質項目 short_length_edge と同様なので、例を省略する。データ品質の改善に使う情報の提供に用いるデータ品質情報についてのみ、インスタンス例を示す。

例は、次の想定に基づいて作った。

● 要件は、精度 10⁻⁵mm で閾値 0.01mm より大きな間隙 (gap) を検出することとする。

- レポートへの要件は、1) 境界の edge と閾値より大きな間隙を持つ face を示し、2) 間の 距離が閾値より大きい、edge 上の点と曲面上の点の対を示し、3) 値の極端な順に、最大 30 件まで報告する。
- 検査は、製品モデルの形状データに対して行った。製品モデルの ID は#116、形状表現 (shape_representation) の ID は#77 である。
- 概要レポートは、検査結果として次の内容を持つものとする。66 枚の face_surface を検査し、閾値より大きな間隙を持つ face_surface 1 枚を検出した。間隙の大きさは 0.013mm である。
- 境界の edge_curve との間隙が大きいと検出された face_surface の詳細情報は、間隙は face_surface #115 で、point_on_edge_curve #37 と point_on_face_surface #217 の間で検出され、その値は 0.013mm である。

この利用シナリオで生成されたインスタンスを、図 II-2-16、図 II-2-17、図 II-2-18 に示す。

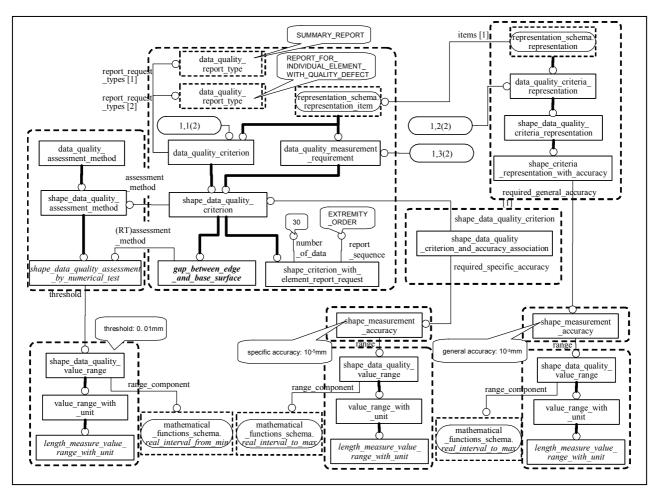


図 II- 2-16 gap_between_edge_and_base_surface の改善に使う情報の提供に用いる品質情報 のインスタンス例 (1/3)

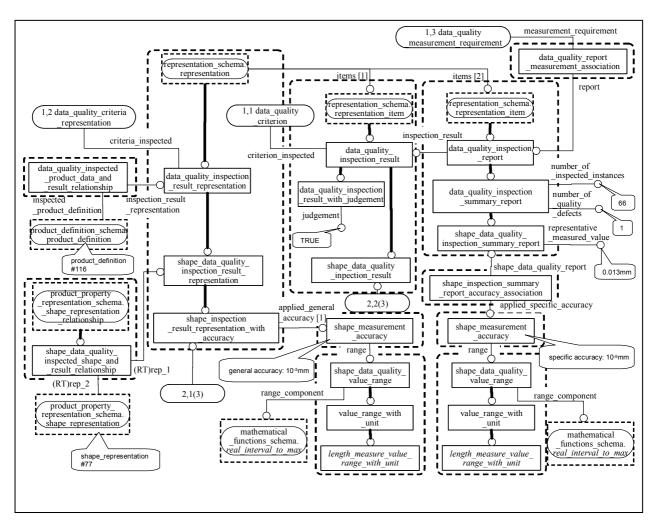


図 II- 2-17 gap_between_edge_and_base_surface の改善に使う情報の提供に用いる品質情報 のインスタンス例 (2/3)

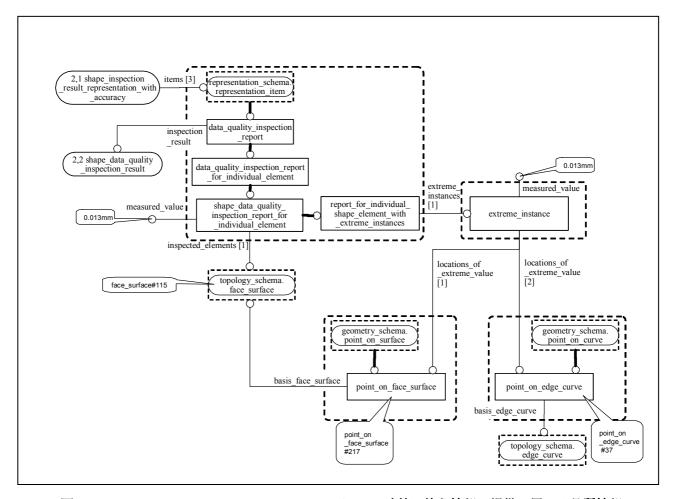


図 II- 2-18 gap_between_edge_and_base_surface の改善に使う情報の提供に用いる品質情報 のインスタンス例 (3/3)

引用文献

【ISO】本項の文章や図は、ISO の許可を得て ISO 10303-59(2007年3月現在の国際規格原案)から引用掲載している。(The texts and figures taken from ISO 10303-59 in this paragraph are reproduced with the permission of the International Organization for Standardization, ISO.)

提供データ

【エリジオン】データは、株式会社エリジオン殿ご提供

2.2 Part 59 の活用法と拡張課題

本節では、Part 59 を実務で活用し、適用範囲を拡張するための技術課題を検討する。

Part 59 は、ISO 10303 の統合リソースとして開発されているため、その成果を実務に適用するためには、何らかの形で応用規格(AP: Application Protocol)を介する必要がある。そのためには、大きく分けて、従来の AP を改訂する方法と、新しい AP (もしくはそれに準ずるもの)を定義する方法の2種類が考えられる。以下、2.2.1 では前者を、2.2.2 では後者を検討する。

また、適切な方法のAPを介してPart 59を使用できたとしても、製品データ品質の検査対象 データがSTEP ファイルのみに限定されることは実務展開上大きな問題である。現在、CAD データの流通では、自社内においても取引先とのやり取りにおいても、STEP のみではなく、個々の CAD に依存するデータベースの外部ファイルを使用する場合が増加している。このため対象がSTEP ファイルのみであるという制限は、Part 59の展開上大きな障害となる可能性が高い。2.2.3 では、2.2.2 の検討を基に PDQ 検査の対象を非 STEP データに拡張するための技術課題を検討する。

2.2.1 既存 AP への組み込み方法

本項では、既存APへの組み込み手順について説明する。

(1) 組み込み手順概要

AP203 や AP214 といった既存の AP に、Part 59 を組み込む場合について述べる。つまり、既存 AP+Part 59 の Long Form を生成するものである。ここで Long Form とは、この合体したスキーマのことを意味している。図 II・2・19 は、その構成と流れを示したもので、既存 AP の Short Form と Part 59 に対して、Part 41 や Part 42 などの統合リソースを結合することで、既存 AP に Part 59 の機能を含む所要のスキーマ(の Long Form)を生成している。

以上が、既存 AP に Part 59 を組み込む場合の基本的な手順であるが、通常は既に存在する AP の (Short ではなく Long Form)スキーマに対して、Part 59 とそれらが呼ぶ他の統合リソースを結合するのが現実的である。またこの各種のスキーマを結合する作業は、当然手作業で行うことも出来るが、それをより効率的かつ正確に行う為に現存する各種の STEP ツールを用いるのが一般的である。

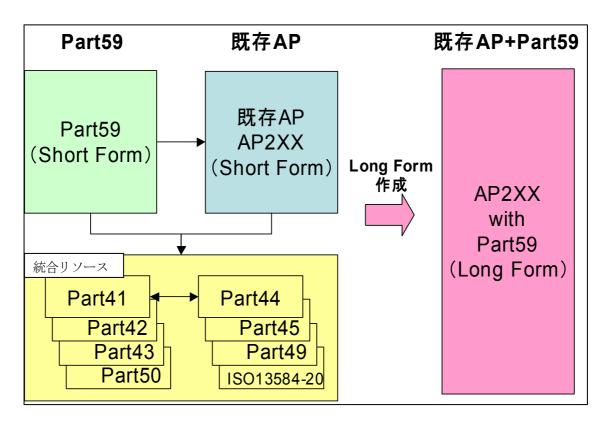


図 II-2-19 既存 AP への組み込み手順概要

付録2に、AP 203 を例にとり Part 59 を組み込んだスキーマ (Long Form) を示す。

(2) 組み込み例 (インスタンスの EXPRESS-G)

本項では、既に取り上げている 2 つの品質項目 short_length_edge と gap_between_edge_and_base_surface について、Part 59 を組み込んだ AP 203 スキーマで表現したインスタンス例を示す。

具体的に、2つの品質項目が検出される AP 203 形式の Part 21 ファイルを対象とし、2.1.4 で述べた "(4) データ品質改善に使う情報の提供"に基づき生成した。

- ① short length edge の AP 203+Part 59 インスタンス例
 - (a) 検査対象データについて
 - ファイル名: TorsionSpringLeft.stp【エリジオン】
 - モデル内に edge (稜線) が 71 個存在

付録3に、ファイル「TorsionSpringLeft.stp」を示す。

(b) 閾値と検査結果について

例では、閾値 0.01mm 以下の稜線を検査すると、以下の 3 つの edge (稜線) が検出される。

検出されるインスタンスNo.と計測値は以下のとおり。

#254:0.0029mm#232:0.0023mm#243:0.0079mm

(c) 検査対象のモデル

図 II-2-20 に、検査対象のモデルを示す。

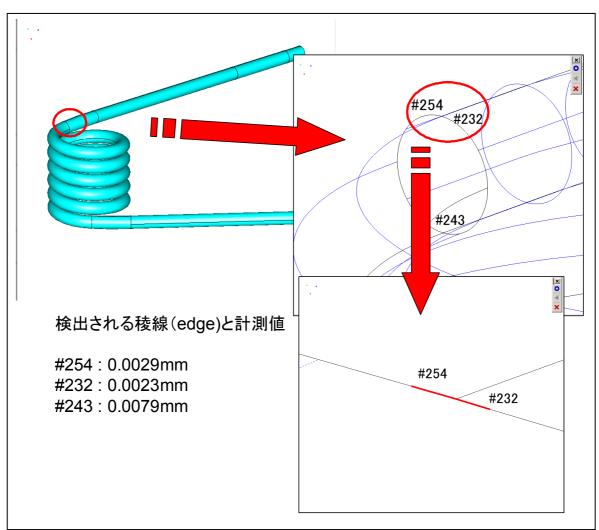


図 II- 2-20 short_length_edge が検出されるモデル【エリジオン】

(d) インスタンス例

ここでは、Part 59 の品質項目 short_length_edge について、Part 59 を組み込んだ AP 203 スキーマで表現したインスタンス例を示す。

図 II- 2-21 は AP 203 のインスタンス例、図 II- 2-22 は Part 59 から AP 203 を参照するインスタンス例である。

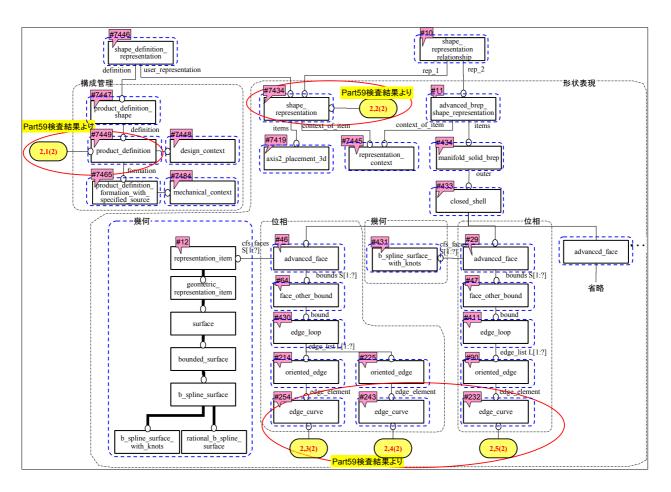


図 II- 2-21 short_length_edge の AP 203+Part 59 インスタンス例 (1/2)

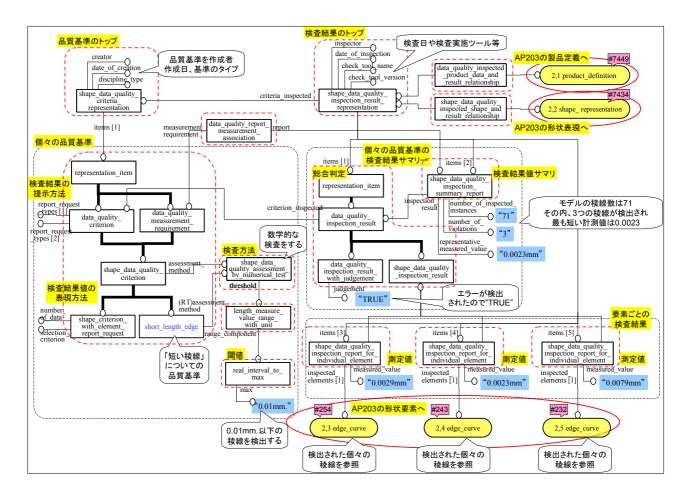


図 II- 2-22 short_length_edge の AP 203+Part 59 インスタンス例 (2/2)

- ② gap_between_edge_and_base_surface のAP 203+Part 59 インスタンス例
 - (a) 検査対象データについて
 - ファイル名: zone1.stp【エリジオン】
 - モデル内に face (基底曲面) は、35 個存在

付録4に、ファイル「zone1.stp」を示す。

(b) 閾値と検査結果について

例では、閾値は 1.0mm 以上の離れを検査すると、以下の 2 組の face (基底曲面) と edge (稜線) が検出される。

検出されるインスタンスNo.と計測値は以下のとおり。

• #93 \mathcal{O} face \geq #443 \mathcal{O} edge : 1.088566mm.

• #89 ∅ face と#433 ∅ edge : 1.075071mm.

(c) 検査対象のモデル

図 II-2-23 と図 II-2-24 に、検査対象のモデルを示す。

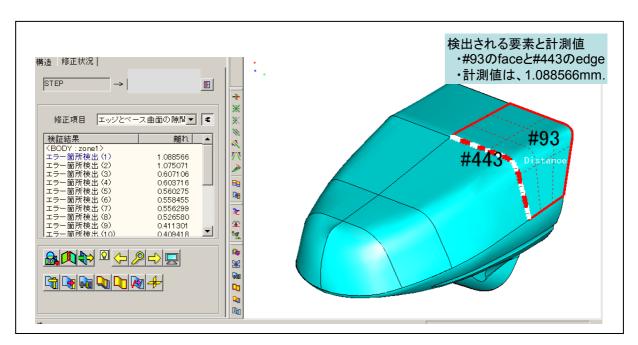


図 II- 2-23 gap_between_edge_and_base_surface が検出されるモデル(1/2) 【エリジオン】

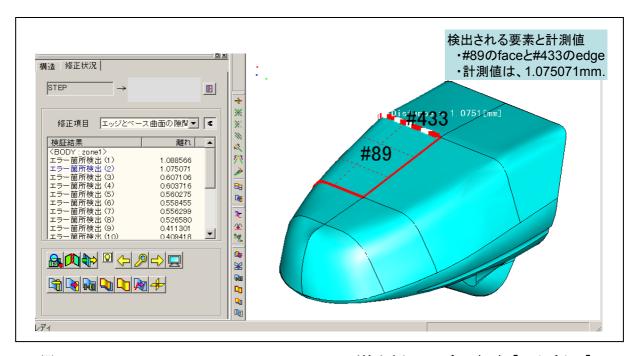


図 II- 2-24 gap_between_edge_and_base_surface が検出されるモデル (2/2) 【エリジオン】

(d) インスタンス例

ここでは、Part 59 の品質項目 gap_between_edge_and_base_surface について、Part 59 を組み込んだ AP 203 スキーマで表現したインスタンス例を示す。

図 II- 2-25 は AP 203 のインスタンス例、図 II- 2-26 は Part 59 から AP 203 を参照するインスタンス例である。

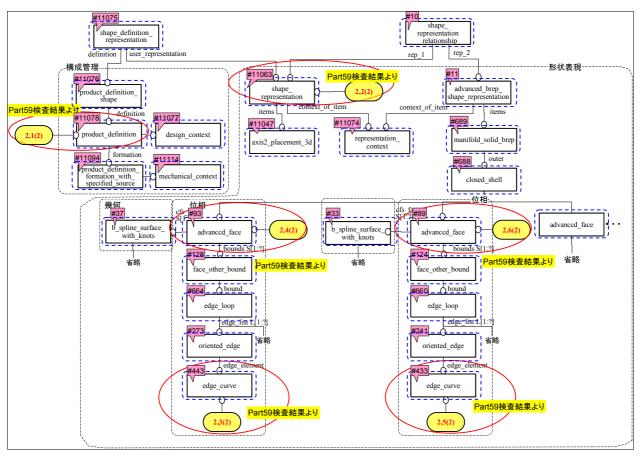


図 II- 2-25 gap_between_edge_and_base_surface の AP 203+Part 59 インスタンス例(1/2)

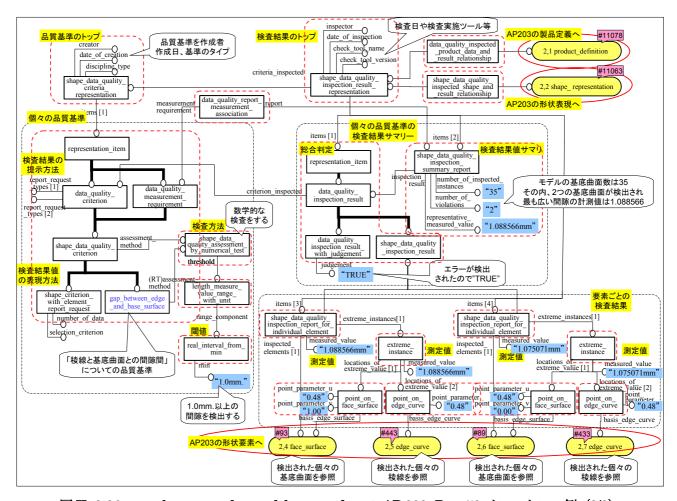


図 II- 2-26 gap_between_edge_and_base_surface の AP 203+Part 59 インスタンス例(2/2)

提供データ

【エリジオン】データや検査結果画面は、株式会社エリジオン殿ご提供

2.2.2 汎用的な組み込み方法の技術的検討

本項では、製品データ品質規格を汎用的に組み込むための技術的検討を記す。

Part 59 は、形状モデルとして Part 42 を参照しているため、技術的には AP 203、AP 214 など Part 42 を利用する全ての AP から利用可能である。従って前項のような手続きにより、既存の AP に Part 59 の機能を追加することで、利用環境は構築できる。しかし、これは個々の AP ごとに準備する必要があることや、誰でも簡単にこの結合されたスキーマを生成できる訳ではないなど、現実としては問題が多い。この問題を解決するために、既存 AP ファイルに対して外側から PDQ 情報を付加する機能を持つ、新しい仕組みの技術的検討を行った。我々はこの仕組みを、リソースを包んで利用できる環境を提供するという意味から、PDQ シェルと呼んでいる。PDQ シェルの利用環境が整えば、Part 42 を使用している、つまり幾何形状を扱っている全ての既存 AP はオリジナルの形のままで Part 59 の機能を利用することが可能になる。

(1) PDQ シェルの概要

図 II- 2-27 に、本手法を適用する場合の PDQ 検査の想定フローを示す。

PDQ チェッカは、PDQ 検査仕様と形状データを受け取り、PDQ シェルの仕様に従って、検査結果ファイルを出力する。本手法では、既存 AP の定義は修正しないため、既存 AP ファイルとは別に PDQ シェル独自のファイルが必要である。つまり、既存 AP ファイル中に存在する検査対象の形状表現や詳細レポートにおける個々の形状要素を、PDQ シェルファイルから要素レベルで外部参照することが必要となる。これが本手法で解決しなければならない最大の課題である。

本手法で外部参照の課題を解決することにより、Part 59 の適用対象をSTEP 以外の独自のデータ構造を有するCAD にも容易に拡張可能である。この点については2.2.3 に詳細を記す。

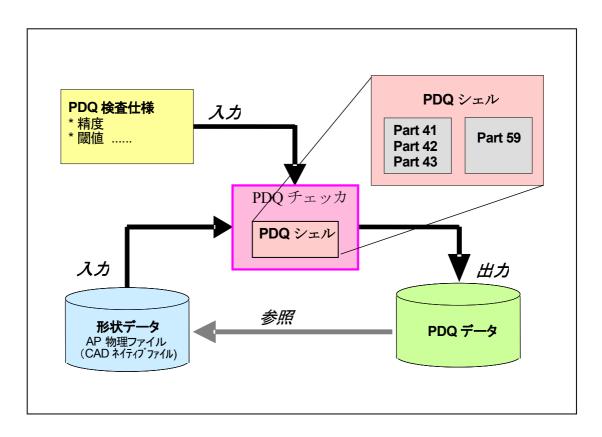


図 II- 2-27 PDQ シェル適用時の PDQ 検査想定フロー

(2) 外部参照の方法

図 II-2-28 に、今回検討した外部参照の仕組みを図示する。

PDQ シェルは簡単な外部参照機構によって形状データ内にある要素を指す。この外部参照機構はファイル名(URI)、要素タイプ、要素 ID を文字列として保持することで実現する。図中、要素の参照に加えて context の参照を示している。これは、Part 59 の PDQ 検査の詳細レポートは、単に形状データ中に存在する要素を指すだけでなく、独自に 3 次元データを持つからである。例として稜線と基底曲面の隙間(gap_between_edge_and_base_surface)では、実際に隙間が検出された edge 上の点とそれに対応する基底曲面上の点を詳細レポートとして出力する。これらの点は元の形状データ中には存在しないため、Part 59 側で表現する必要がある。

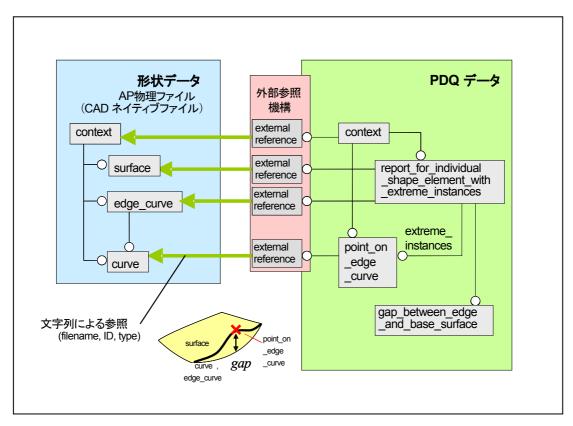


図 II-2-28 PDQ シェルにおける外部参照の仕組み

図 II- 2-29 に、外部参照を用いた PDQ シェルの実装案を、具体的な事例に基づいて示す。例 として gap_between_edge_and_base_surface を用いている。図中、上から、形状データ(既存 AP の物理ファイル)、外部参照のためのインスタンス案、そして外部参照のための定義案を示している。

本手法は、STEP ファイルに対して汎用的に適用できるだけでなく、後述のとおり独自のデータ構造を有する CAD に対しても容易に拡張可能と考えられる。Part 59 の実用化に向けて非常に重要な技術であり、来年度以降、実装に向けてより詳細な検討が期待される。

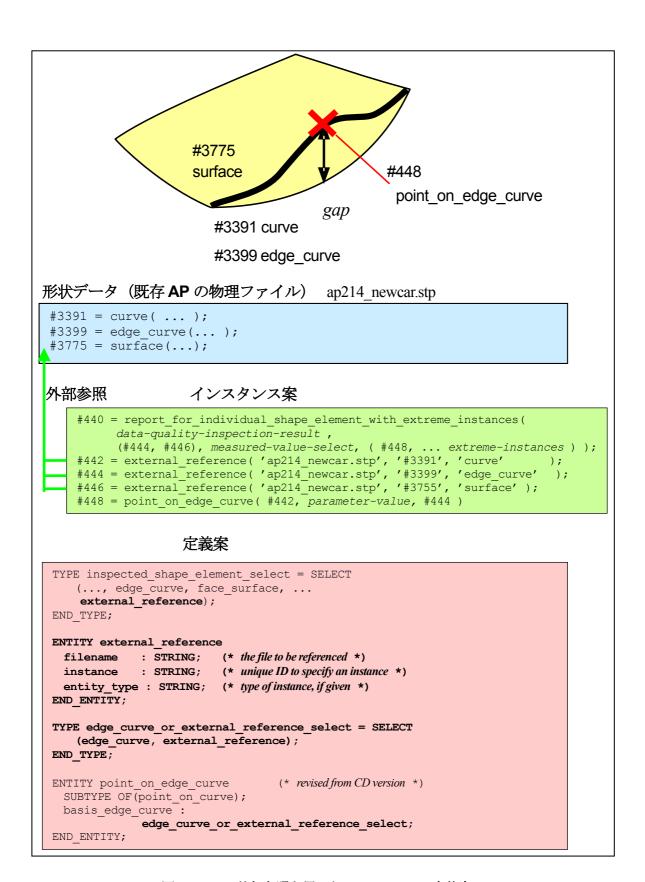


図 II- 2-29 外部参照を用いた PDQ シェルの実装案

2.2.3 非 STEP ファイルに適用するための検討

本項では、Part 59 を非 STEP ファイルに適用するための技術的検討を以下の観点から行う。

- 品質項目が適用可能であるか
- 検査結果が表現可能であるか

なお、AP としての実装方法は、2.2.2 で検討された、PDQ シェルによるものとする。

(1) 品質項目について

ここでは、非 STEP ファイルに対して、Part 59 の品質項目による品質検査が可能であることを確認する。

まず検査の対象となる形状データについて検討する。先行規格である SASIG PDQ Guidelines においては、前提となる形状モデルが曖昧であったために、ある形状要素が検査の対象となるか 否かというレベルにおいてもツールベンダ間で解釈の差異が見られている。

これに対し、Part 59では、形状モデルとしてPart 42を用いている。現在、ほとんど全てのCADシステムは3次元形状データをBrepとしてAP 203(およびAP 214)のファイルに書き出す機能を備えており、自システムの要素とPart 42の要素の対応は既に取れていると考えるのが自然である。Part 59では各品質項目の対象となるPart 42要素をruleによって厳密に指定しているので、各品質項目の対象要素は、STEP変換における要素の対応を通じて、CADファイルに対して明確に定めることができる。

また、対象となる形状要素に対する適切な検査のための外部仕様も、アルゴリズムの詳細に立ち入らない範囲で明確に示されている。アルゴリズムの詳細に立ち入らない記述であるため、この仕様も汎用的に様々な CAD に対して適用可能である。SASIG PDQ Guidelines では、検査の外部仕様も実装から離れた曖昧な記述であった。このため、実装者の解釈に依存し、CAD ごと、PDQ ツールベンダごとに検査の結果が大きく異なっていた。

Part 59 の品質項目は全ての CAD データに対して汎用的に適用可能で、且つ解釈の曖昧性が極力排除された規格となることが可能である。

要素の参照

- 要素を一意に特定できるIDが存在すれば良い
 - CATIA, Pro/E, I-DEASILOK
 - 代表的ソリッドモデリングカーネルであるParasolid、 ACISにも存在する
- 上記のIDは、基本的にシステムの内部IDである。エンドユーザに公表された情報ではない場合が多いが、この情報を用いて様々なアプリケーションを実装することが可能である
 - PDQチェッカ PDQ検査ツールの連携、CAD上のPDQ 機能開発、CADベンダへのバグレポートなど

図 II- 2-30 要素の参照

検査結果を表現するためには、2.2.2で検討した外部参照の手法を用いることが必要である。

現在使用されている主要な CAD (CATIA、Pro/Engineer、NX) および 3 次元ソリッドモデリングカーネル (Parasolid、ACIS) では、各システムの API を介して要素を一意に特定することが出来る ID にアクセス可能であることが分かっている。よって、図 II-2-28 における個別要素の参照は実現可能である。

Context@mapping

- Unit
 - 各CADで様々なunitが存在するが、Part 41 の"conversion_based_unit"によって吸収可能なため問題なし。
- 座標系について
 - 一部のCADでは左手系、および1.0以外のスケーリングをサポートする。(CATIA)
 - Part 42の"axis2 placement 3d"は右手系
- assembly中のpartの特定方法
 - システムによって異なる
 - ・ 部品名が一意(重複を許さない)
 - 部品名+部品番号が一意
 - アセンブリツリー中の経路が一意
 - 最後の方法で特定できれば十分

図 II-2-31 Context の mapping

また、図 II-2-29 中に見られるように context の mapping も必要である。context には単位系 および座標系の情報が含まれる。単位系は、Part 41 に"conversion_based_unit"という汎用的な 要素が存在するので、全ての CAD の情報を問題なく表現することができる。

座標系に関して、CAD 間で仕様が異なるのは左手系の存在とスケーリングの有無であるが、この点についても現在の STEP の表現力の範疇で十分な効果が出るものと考えられる。

(3) 結論

図 II-2-32 に検討内容をまとめた。

まとめ

- 品質基準と計測手法の定義は汎用的で全CADに適用可能。summary reportも適用可能。詳細なreportは外部から参照可能なIDを持つCADに は適用可能。
- 対策
 - 参照可能なIDを持つかどうか、持つ場合はその仕様を、主要なCAD毎に明らかにする
 - 左手系の座標系を持つ主要なCADを明らかにする
 - 主要CADの各形状要素とPart 42要素のmappingを明らかにする
 - 上記条件の下にnative CAD用のinspected product definition, summary report, report for individual elementの外部参照方法を検討する

• 残課題

- 左手系の配置マトリックスを許すCADではassemblyをそのまま検証できない (part毎のcheckなら可能)
- native CAD -> Part 42のmappingが近似的にしか行われない場合がある。
 - CADデータとしては問題ないが、Part 42として表現するために問題となる場合がある
 - 但し、このケースでも、外部へのデータ交換では必ず問題となる

図II-2-32 まとめ

結論として、外部参照による PDQ シェルを用いることで、Part 59 は非 STEP データに対して汎用的に適用することが可能であると推測できる。

今後、2.2.2 の PDQ シェルの詳細な検討を進めることが、非 STEP データへの適用においても 非常に重要である。

2.3 Part 59 の実務適用のための要件

本節では、製品データ品質に関する現状の問題に対し、Part 59 適用による効果と残課題が何であるかを検討することを通して、Part 59 を実務に広く適用するための要件を明らかにする。

2001年にJAMAによってデータ品質に起因する損失が日本の自動車産業で年間71億円であるという見積りが公表され、また、2000年にJAMA/JAPIAから、2001年にSASIGからPDQGuidelinesが発行された結果、自動車業界を中心に3次元CAD形状データの品質問題に対する関心が高まり各社で様々な取り組みが行われてきた。このような事例の一部はJAMAのホームページからダウンロード可能である。例えば、日本の大手自動車メーカA社においては、3次元形状について15の品質項目と対応する閾値を策定し運用した結果、CADサポート部隊によるデータ調査/修復の件数が95%減少したという結果が公開されている(【JAMA】)。

一方で、実務環境でのデータ品質向上の推進には障害も多く、着手できない、活動は開始した が思うような成果が上がらないという事例も数多く存在する。

以下ではまずデータ品質向上の阻害要因を述べ、Part 59 の効果そして残課題を記す。

2.3.1 データ品質向上の阻害要因

ここでは業務プロセスに関する問題と技術的な問題に分けて阻害要因を記す。

(1) 業務プロセス上の問題

形状のデータ品質問題は、CAD データを作成する設計のプロセスにおいて、問題となる形状が作成されることによって埋め込まれる。データ品質を高めるための仕組みや基準の無い場合には、生産準備、金型設計、あるいは解析など様々なプロセスで設計データを利用する際に初めてこのデータ品質問題が顕在化する。そのため、これらデータ利用部門の担当者が多大な工数をかけてデータ整備を行うことが業務として常態化している場合が多い。

データ品質向上の取り組みにおいては、通常、適切な品質項目と閾値を選定し、それを満たさない設計データの流通が禁じられる。つまり、基本的には、データ流用部門の多大な工数を、設計部門が工数をかけて未然に防ぐ構造となる。そのため、データ品質向上のための活動は一般的に以下の様な問題をはらんでいる。

1-i) 部門ごとにデータ品質に対する要求が異なる。例えば、設計部門では問題無い品質の CAD データが生産準備では使えないというケースも非常に多い。往々にして 2 次元データしか無 い場合も多い概念設計の段階では、形状が表現できていれば精度などのデータ品質は特に問われないが、詳細設計においては、CAD システムが要求する精度が必要となる。CAE にお

いては基本的に細かいデータが悪影響を及ぼすため、工数をかけて簡略化した解析モデルを作成するケースが多く見られる。生産準備においては、抜き勾配、スプリングバックやオーバクラウンなどの見込み変形などの特有の要件が多く、多くの工数を必要とするデータの再作成が必須となっている。また、CAMではデータに 0.001mm レベルの精度が要求されるといった明らかに相反する条件も含まれるため、上記の要件を全て単一のデータで満足することは不可能である。

1-ii) PDQ 改善のために工数を費やす部門(データ作成部門)とこの活動による受益者(データ利用部門)が異なる場合が多いため、部門横断的な効果の測定とプロジェクト推進が必要となる。言い換えれば、データ品質問題の損失や改善効果が単一の部門で算出できず非常に困難であるため、活動の動機付けに支障を来たす。

(2) 技術的な問題

データ品質向上には、ソフトウェアとして CAD、PDQ チェッカ、PDQ 修正ツールが使われており、それぞれの機能に問題を抱えている。また、製品データ品質を推進する人材の面でも技術的な問題が存在する。

- 2-i) 各社、各部門のデータ品質問題に対し、適切な品質項目を対応付けるためには非常に高度な専門知識を要する。特に形状に関する深い理解が必要である。また、品質項目の選定作業に際しては、ブラックボックス化している各 CAx ツールの仕様を、多くの不具合事例から推定する作業が必要であり、多大な工数が必要となる。
- 2-ii)データ品質問題が CAD の不具合であるか、ユーザのオペレーションの問題であるか、切り分けが困難であるため、CAD のコマンドが生成する形状の品質向上がなかなか進まない。また、悪形状に関して統一した明確な技術的な定義が無いため、ユーザと CAD ベンダ、PDQ ツールベンダと CAD ベンダの間のデータ品質に関するコミュニケーションが円滑に進まない場合がある。
- 2-iii) 同じ品質項目、同じ閾値に対する PDQ チェックの結果が、ツールごとに異なる。これは、現在の SASIG PDQ Guidelines において各項目の記述が曖昧であり、検査方法や検査結果の精度に対する制限もほとんど存在しないからである。
- 2-iv) 現在主流のソリッドモデラにおいては、形状データは、パラメタ付きのフィーチャとその 履歴から、CAD ソフトウェアの内部で自動的に計算される。この自動計算で問題形状が作られる場合には、本質的には CAD ベンダの協力無しに問題を解決できない。しかし、2-ii)で述べた原因切り分けの問題や、CAD の改善とそれを織り込んだソフトウェアのバージョ

ンアップに年単位の時間がかかるという問題などから、ユーザ側で対策を講じている場合も多い。しかし、この原因によって発生する問題形状を、CAD の改善以外の手段で修正しようとする場合、履歴情報が落ちてしまうことがほとんどである。そのため、例えば、設計変更や流用品設計など履歴を利用する作業を PDQ 整備前のデータから開始しなければならないなどといった悪影響が生まれている。

2.3.2 Part 59 によって見込まれる効果

Part 59 は、3 次元データの品質上問題となる形状に関して、曖昧性を極力排除した定義を与えることが技術面での大きな特徴である。

この特徴により、2.3.1 で述べた諸問題の内、技術的な問題の 2-ii)、2-iii)に対しては非常に大きな効果が見込まれる。2-ii)に対しては、悪形状に関する様々なコミュニケーションを行う際の技術的共通言語の形成が可能となる。また、2-iii)については、Part 59 に準拠している PDQ ツール間では、検査結果の差異は SASIG PDQ Guidelines 準拠の場合と比較して大きく低減されることが見込まれる。特に、Part 59 では検査精度の概念を導入しているので、外部仕様上の差異を排除するだけでなく、近似計算結果の精度まで制御することが可能となる。

また、CAD ベンダが、Part 59 を用いて、自社製品が作成するデータや自社製品に受け入れ可能なデータの品質を宣言することを可能にすれば、品質項目選定の労力が軽減され、CAD システムと操作の間の問題切り分けが明確になるなど、問題 2-i)に関しても非常に効果が大きいと思われる。

さらに、業務プロセス上の問題 1-i)に対しても各部門の品質要求を明確に表現可能になるという効果は期待できる。1-ii)に対しては、Part 59 の様々なレポート機能を使用することで、改善活動中の3次元データの定量的評価や分析が容易となる。但し、いずれにしてもプロセス上の問題を根本的に解決するには規格だけでなく、組織的な取り組みが必要である。

2.3.3 今後の課題

Part 59 によって形状データ品質向上のための技術的な基盤が形成されるとしても、様々な課題は残る。

まず、技術的には、問題 2-iv)が目前の最大の課題である。現在、履歴付きフィーチャからの形状の計算は CAD 毎にブラックボックス化されており、ユーザがその結果生まれる形状の品質を制御することはほとんどできない。ユーザ、PDQ ツールベンダと比較して CAD ベンダを巻き込んだ形状データ品質向上の活動が望まれる。その際のコミュニケーションツールとして、Part 59は有効に活用できるであろう。

この問題が解決されたとしても、問題 1-i)の背後に存在する、より深い技術的な問題は残存する。それは、ある製品モデルのデータは、ライフサイクル中に様々なツール上で様々な用途に利用されるため、適当な要件に応じて様々に形を変える必要があるという問題である。金型用の見込み変形、簡略化された解析モデル作成、あるいは概念設計と詳細設計の間のデータの受け渡しなど、現在ユーザは工数をかけて、用途に応じたデータを作り直しているのが実情である。これは、この分野の問題は定式化された解決方法が確立されていないからである。同じ理由により、「使えるデータ」であるかどうかという意味では非常に重要な問題であるにも関わらず、この問題はPart 59 の対象からも外れている。今後、研究者、開発者の今後の取り組みにより技術的な進歩が遂げられることを期待したい。

最後に、プロセス上の問題 1·i)、1·ii)に対しては組織的な取り組みが必須であり、これは規格やツールだけでは根本的に解決できない。上述したとおり、製品データ品質向上に取り組むためには、製品データのライフサイクル全体を視野に入れた組織横断的な取り組みが必須となる。データ品質向上のためには、利害の相反する部署間の調整、特にデータ作成者の、品質向上に対する意識やモチベーションの向上、仕組みやルール作りなどが必要であり、使用する規格や推進者の技術力に加えて、高度な組織的行動力も必須となることを強調しておきたい。

参考文献

【JAMA】JAMA ホームページ > クルマと情報化 > PDQ (Product Data Quality: モデルデータ品質) に関わる活動 普及資料 先進活動事例

(http://www.jama.or.jp/cgi-bin/pdq/download_pdq.cgi), JAMA, 2006

付録 1. Part 59 の品質項目の分類と一覧

表 II- 付録 1-1 Part 59 の品質項目の分類と一覧

(1)erroneous_da	(1)erroneous_data (不正なデータ)				
①erroneo	①erroneous_topology (不正な位相)				
	open_edge_loop (開いている稜線ループ)				
	open_closed_shell (開いている閉シェル)				
	inconsistent_adjacent_face_normals (隣接した面の法線の不整合)				
2erroneo	②erroneous_geometry (不正な幾何)				
	inconsistent_curve_transition_code(曲線遷移コードの不整合)				
	erroneous_b_spline_curve_definition(不正な b-spline 曲線の定義)				
	inconsistent_surface_transition_code (曲面遷移コードの不整合)				
	erroneous_b_spline_surface_definition(不正な b-spline 曲面の定義)				
③erroneous_topology_and_geometry_relationship(不正な位相と幾何の関係)					
	inconsistent_edge_and_curve_directions (稜線と曲線の方向の不整合)				
	inconsistent_face_and_surface_normals(面と曲面の法線の不整合)				
	inconsistent_face_and_closed_shell_normals(面と閉シェルの法線の不整合)				
	intersecting_loops_in_face (干渉している面内ループ)				
	wrongly_placed_loop (ループ位置の不正)				
4erroneo	us_manifold_solid_brep(不正な多様体立体)				
	wrongly_oriented_void (間違った向きの空洞)				
	wrongly_placed_void(空洞位置の不正)				
	intersecting_shells_in_solid (干渉している立体中シェル)				
(2)inapt_data(不適切なデータ)				
①inapt_to	ppology(不適切な位相)				
	non_manifold_at_vertex(頂点における非多様体)				
	non_manifold_at_edge (稜線における非多様体)				
	free_edge(未接合の稜線)				
	disconnected_face_set (非連結な面群)				
	over_used_vertex (過剰に使用されている頂点)				
②inapt_g	eometry(不適切な幾何)				
near	nearly_degenerate_geometry (近縮退している幾何)				
	short_length_curve(短い曲線)				
	small_area_surface(面積が小さい曲面)				
	entirely_narrow_width_surface(全域に幅が狭い曲面)				

disc	ontinuous_geometry(不連続な幾何)	
	g1_discontinuous_curve(曲線が g1 不連続)	
	g2_discontinuous_curve(曲線が g2 不連続)	
	g1_discontinuous_surface (曲面が g1 不連続)	
	g2_discontinuous_surface (曲面が g2 不連続)	
self	_intersecting_geometry(自己干渉している幾何)	
	self_intersecting_curve(自己干渉している曲線)	
	self_intersecting_surface (自己干渉している曲面)	
geor	geometry_with_local_near_degeneracy (局所的に近縮退している幾何)	
	curve_with_small_curvature_radius(小さな曲率半径をもつ曲線)	
	surface_with_small_curvature_radius(小さな曲率半径をもつ曲面)	
	short_length_curve_segment (短い曲線セグメント)	
	small_area_surface_patch (面積が小さい曲面パッチ)	
	narrow_width_surface_patch (幅が狭い曲面パッチ)	
	indistinct_curve_knots (識別不能な程に近接している曲線ノット)	
	indistinct_surface_knots (識別不能な程に近接している曲面ノット)	
	nearly_degenerate_surface_boundary (近縮退している曲面境界)	
	nearly_degenerate_surface_patch (近縮退している曲面パッチ)	
geor	metry_with_local_irregularity(局所的に不規則な幾何)	
	zero_surface_normal(大きさゼロの法線を持つ曲面)	
	abrupt_change_of_surface_normal (法線が急変する曲面)	
	wavy_planar_curve(波打っている平面曲線)	
	wavy_surface (波打っている曲面)	
	extreme_patch_width_variation (パッチ幅の極端な変化)	
over	rlapping_geometry(部分的に重複している幾何)	
	partly_overlapping_curves (部分的に重複している曲線)	
	partly_overlapping_surface (部分的に重複している曲面)	
mul	tiply_defined_geometry (重複定義された幾何)	
	multiply_defined_points(重複定義された点)	
	multiply_defined_curves(重複定義された曲線)	
	multiply_defined_surfaces(重複定義された曲面)	
over	r_complex_geometry(幾何が過剰に複雑)	
	excessively_high_degree_curve(過剰に高次な曲線)	
	excessively_high_degree_surface(過剰に高次な曲面)	
	curve_with_excessive_segments (過剰なセグメントをもつ曲線)	

	surface_with_excessive_patches_in_one_direction(1 方向に過剰なパッチをもつ曲面)
	high_degree_linear_curve(高次に定義された直線)
	high_degree_conic(高次に定義された円錐曲線)
	high_degree_planar_surface(高次に定義された平面)
	high_degree_axi_symmetric_surface (高次に定義された軸対称曲面)
3inapt	z_topology_and_geometry_relationship(不適切な位相と幾何の関係)
to	ppology_related_to_nearly_degenerate_geometry(近縮退した幾何に関係した位相)
	short_length_edge (短い稜線)
	small_area_face (面積が小さい面)
	entirely_narrow_width_face(全域に幅が狭い面)
ge	eometric_gap_in_topology(位相と関連する幾何の間隙)
	gap_between_vertex_and_edge(頂点と稜線との間隙)
	gap_between_adjacent_edges_in_loop(ループ上で隣り合う稜線間の間隙)
	gap_between_vertex_and_base_surface(頂点と基底曲面との間隙)
	gap_between_edge_andbase_surface(稜線と基底曲面との間隙)
	gap_between_pcurves_related_to_an_edge (単一の稜線に関係する pcurve 間の間隙)
	gap_between_faces_related_to_an_edge(単一の稜線に関係する面間の間隙)
tc 相	ppology_related_to_self_intersection_geometry(自己干渉している幾何に関係する位
	self_intersecting_loop (自己干渉しているループ)
	self_intersecting_shell (自己干渉しているシェル)
	intersecting_shells (干渉しているシェル)
n	on_smooth_geometry_transition_across_edge (滑らかで無い稜線越えの幾何遷移)
	g1_discontinuity_between_adjacent_faces(隣り合う面が g1 不連続)
	g2_discontinuity_between_adjacent_faces(隣り合う面が g2 不連続)
st	teep_geometry_transition_across_edge(急変している稜線越えの幾何遷移)
	steep_angle_between_adjacent_edges (急角度で隣り合う稜線)
	steep_angle_between_adjacent_faces(急角度で隣り合う面)
tc 相	ppology_related_to_overlapping_geometry(部分的に重複している幾何に関係する位
	partly_overlapping_edges (部分的に重複している稜線)
	partly_overlapping_faces (部分的に重複している面)
to	ppology_related_to_multiply_defined_geometry (重複定義された幾何に関係する位相)

		multiply_defined_vertices (重複定義された頂点)
		multiply_defined_edges (重複定義された稜線)
		multiply_defined_faces (重複定義された面)
over_complex_topology_and_geometry_relationship(位相と幾何の関係が過剰に		
		edge_with_excessive_segments (過剰な数のセグメントをもつ稜線)
		face_surface_with_excessive_patches (過剰な数のパッチをもつ面)
		unused_patches (使用されていないパッチ)
④inapt_geometric_model(不適切な形状モデル)		
		small_volume_solid (体積が小さい立体)
		entirely_narrow_width_solid(全域に幅が狭い立体)
		partly_overlapping_solids (部分的に重複している立体)
		multiply_defined_solids (重複定義された立体)
		solid_with_excessive_number_of_voids(過剰な数の空洞を持つ立体)

付録 2. AP 203 with Part 59 の統合スキーマ(Long Form)

以下に、AP 203 を例にとり Part 59 を組み込んだ4つのスキーマを統合した LongForm を示す。統合スキーマの名前は AP203_WITH_PDQ_SCHEMA とした。

```
(* Long form schema generated by The EXPRESS Data Manager compiler
                                                                               *)
                                                        version 9.7.03B.20060802*)
                                                                                *)
(* Fri Mar 02 21:01:17 2007
SCHEMA AP203_WITH_PDQ_SCHEMA;
CONSTANT
(* Implicit interfaced from: topology_schema *)
    dummy tri: topological representation item ≔ representation item(") | |
                  topological_representation_item();
(* Implicit interfaced from: mathematical functions schema *)
schema_prefix : STRING := 'AP203_WITH_PDQ_SCHEMA.';
(* Implicit interfaced from: geometry_schema *)
 dummy_gri: geometric_representation_item ≔ representation_item(") | |
                                  geometric_representation_item();
END_CONSTANT;
(* Implicit interfaced from: geometry_schema *)
TYPE vector or direction = SELECT
  (vector.
   direction);
 END TYPE;
(* REFERENCE FROM (support resource schema); *)
 TYPE label = STRING;
 END TYPE;
(* REFERENCE FROM (measure schema); *)
```

```
TYPE parameter_value = REAL;
 END TYPE;
                               ····· 中略 ·····
(* Implicit interfaced from: PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA *)
 TYPE data quality report type = ENUMERATION OF
   (NO_REPORT_REQUIRED,
    SUMMARY_REPORT,
    REPORT_FOR_INDIVIDUAL_ELEMENT_MEASURED,
    REPORT_FOR_INDIVIDUAL_ELEMENT_VIOLATED);
 END_TYPE;
(* Implicit interfaced from: SHAPE_DATA_QUALITY_CRITERIA_SCHEMA *)
 TYPE interval select = SELECT
   (real_interval_to_max,
    real_interval_from_min,
    finite real interval,
    integer_interval_from_min,
    integer interval to max,
    finite integer interval);
 END_TYPE;
(* Implicit interfaced from: SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA *)
(* This SELECT set has been pruned. *)
TYPE inspected_shape_element_select = SELECT (
  edge_loop,
  edge_curve,
  face surface,
  curve,
  surface,
  vertex_point,
  point,
  connected_face_set);
END TYPE;
(* Implicit interfaced from: SHAPE DATA QUALITY INSPECTION RESULT SCHEMA*)
```

```
(* This SELECT set has been pruned. *)
TYPE measured_value_select = SELECT (
  parameter_value,
  measure_with_unit);
END TYPE;
(* Implicit interfaced from: SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA *)
(* This SELECT set has been pruned. *)
TYPE location_of_extreme_value_select = SELECT (
  inspected_shape_element_select,
  point_on_surface,
  point_on_curve,
  oriented edge,
  face_bound);
END_TYPE;
                               •••• 中略 ••••
(* REFERENCE FROM (PRODUCT DATA QUALITY CRITERIA SCHEMA); *)
 ENTITY data_quality_criteria_representation
   SUBTYPE OF(representation);
                    : person;
     creator
     date_of_creation: date_and_time;
     discipline_type : text;
   WHERE
     WR1: SIZEOF(QUERY(q <* SELF¥representation.items |
'PRODUCT_DATA_QUALITY_REPRESENTATION_SCHEMA.' +
                             'DATA_QUALITY_CRITERION' IN TYPEOF(q))) > 0;
 END ENTITY;
(*\ REFERENCE\ FROM\ (PRODUCT\_DATA\_QUALITY\_CRITERIA\_SCHEMA);\ *)
  ENTITY data_quality_criterion
   SUBTYPE OF(representation item);
     report_request_types: SET [1:2] OF data_quality_report_type;
   WHERE
     WR1: SIZEOF(USEDIN(SELF,
```

```
'AP203 WITH PDQ SCHEMA.'+
          'DATA QUALITY INSPECTION RESULT REPRESENTATION.ITEMS')
          )=0;
  END ENTITY;
(* REFERENCE FROM (PRODUCT DATA QUALITY CRITERIA SCHEMA); *)
  ENTITY data_quality_measurement_requirement
   SUBTYPE OF(representation_item);
   WHERE
     WR1: SIZEOF(USEDIN(SELF.
          'AP203 WITH PDQ SCHEMA.'+
          'DATA_QUALITY_INSPECTION_RESULT_REPRESENTATION.ITEMS')
  END ENTITY;
(* REFERENCE FROM (PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA); *)
  ENTITY data_quality_assessment_method;
     description: text;
 END ENTITY;
(*
    REFERENCE
                    FROM
                             (PRODUCT DATA QUALITY INSPECTION RESULT
SCHEMA); *)
  ENTITY data_quality_inspection_result_representation
   SUBTYPE OF(representation);
     check tool name
                      : label;
     check tool version: label;
     criteria_inspected : data_quality_criteria_representation;
     date_of_inspection: date_and_time;
     inspector
                     : person;
   WHERE
     WR1: SIZEOF( QUERY( q <* SELF\( \) representation.items |
                      'AP203 WITH PDQ SCHEMA.' +
                      'DATA_QUALITY_INSPECTION_RESULT' IN TYPEOF(q))) > 0;
  END ENTITY;
    REFERENCE
                    FROM
                            (PRODUCT DATA QUALITY INSPECTION RESULT
SCHEMA); *)
 ENTITY data_quality_inspection_result
   SUBTYPE OF(representation item);
```

```
criterion_inspected : data_quality_criterion;
   WHERE
     WR1: SIZEOF(USEDIN(SELF,
           'AP203 WITH PDQ SCHEMA.'+
          'DATA QUALITY CRITERIA REPRESENTATION.ITEMS')
          )=0;
 END ENTITY;
(*
    REFERENCE
                             (PRODUCT_DATA_QUALITY_INSPECTION_RESULT_
                    FROM
SCHEMA); *)
  ENTITY data_quality_inspection_summary_report
   SUBTYPE OF(data_quality_inspection_report);
     number_of_inspected_instances : INTEGER;
     number of violations
                                : INTEGER;
  END ENTITY;
    REFERENCE
                    FROM
                             (PRODUCT_DATA_QUALITY_INSPECTION_RESULT_
SCHEMA); *)
  ENTITY data_quality_inspection_report_for_individual_element
   SUBTYPE OF(data_quality_inspection_report);
 END ENTITY;
    REFERENCE
                             (PRODUCT_DATA_QUALITY_INSPECTION_RESULT_
                    FROM
SCHEMA); *)
  ENTITY data_quality_inspected_product_data_and_result_relationship;
     inspected product definition
                                  : product definition;
     inspection_result_representation: data_quality_inspection_result_representation;
  END_ENTITY;
(* REFERENCE FROM (SHAPE_DATA_QUALITY_CRITERIA_SCHEMA); *)
  ENTITY shape measurement accuracy;
     description: text;
                : shape_data_quality_value_range;
     range
  END_ENTITY;
(* REFERENCE FROM (SHAPE_DATA_QUALITY_CRITERIA_SCHEMA); *)
(* Pruned SUPERTYPE *)
  ENTITY shape data quality criterion
 SUBTYPE OF(data_quality_criterion, data_quality_measurement_requirement);
     assessment method: shape data quality assessment method;
```

```
END ENTITY;
(* REFERENCE FROM (SHAPE DATA QUALITY INSPECTION RESULT SCHEMA); *)
  ENTITY shape_data_quality_inspection_report_for_individual_element
   SUBTYPE OF(data_quality_inspection_report_for_individual_element);
     inspected elements: SET [1:?] OF inspected shape element select;
                         : measured_value_select;
     measured value
  END ENTITY;
(* REFERENCE FROM (SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA); *)
  ENTITY report_for_individual_shape_element_with_extreme_instances
   SUBTYPE OF(shape_data_quality_inspection_report_for_individual_element);
     extreme_instances : SET [1:?] OF extreme_instance;
  END ENTITY;
(* REFERENCE FROM (SHAPE DATA QUALITY INSPECTION RESULT SCHEMA); *)
  ENTITY shape_data_quality_inspection_summary_report
   SUBTYPE OF(data_quality_inspection_summary_report);
     representative measured value: measured value select;
  END ENTITY;
(* REFERENCE FROM (SHAPE DATA QUALITY INSPECTION RESULT SCHEMA); *)
  ENTITY curvature unit
   SUBTYPE OF(named unit);
   WHERE
     WR1: (SELF\(\preceq\)named_unit.dimensions.length_exponent = -1.0) AND
           (SELFYnamed unit.dimensions.mass exponent = 0.0) AND
           (SELFYnamed unit.dimensions.time exponent = 0.0) AND
           (SELF\(\preceivage)\) named_unit.dimensions.electric_current_exponent = 0.0) AND
           (SELF\(\frac{1}{2}\) named_unit.dimensions.thermodynamic_temperature_exponent = 0.0)
AND
           (SELF\(\frac{1}{2}\) named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
           (SELF\(\preceivage)\) named_unit.dimensions.luminous_intensity_exponent = 0.0);
  END ENTITY;
                                 •••• 中略 •••••
(* REFERENCE FROM (aic geometrically bounded surface); *)
FUNCTION gbsf check point (pnt:point): BOOLEAN;
(* This function varifies the validity of a point in the context of a
```

geometrically bounded surface model.

```
*)
 (* a cartesian point is valid *)
 IF 'AP203_WITH_PDQ_SCHEMA.CARTESIAN_POINT'
   IN TYPEOF (pnt) THEN RETURN(TRUE);
  ELSE
  (* a point on curve shall reference a valid curve*)
   IF 'AP203_WITH_PDQ_SCHEMA.POINT_ON_CURVE'
     IN TYPEOF (pnt) THEN RETURN
     (gbsf_check_curve(pnt\print_on_curve.basis_curve));
   ELSE
   (* a point on surface shall reference a valid surface*)
     IF 'AP203_WITH_PDQ_SCHEMA.POINT_ON_SURFACE'
       IN TYPEOF (pnt) THEN RETURN (gbsf_check_surface
       (pnt\proptypoint on surface.basis surface));
     ELSE
     (* a degenerate pourve shall reference a valid curve and
        a valid surface*)
       IF 'AP203 WITH PDQ SCHEMA.DEGENERATE PCURVE'
         IN TYPEOF (pnt) THEN RETURN
         ((gbsf check curve
         (pnt\degenerate pcurve.reference to curve\delta
                                            representation.items[1]))
         AND (gbsf_check_surface (pnt\def{ptydegenerate_pcurve.basis_surface)));
       END IF;
     END_IF;
   END_IF;
 END IF;
 RETURN(FALSE);
END_FUNCTION;
(* REFERENCE FROM (aic_geometrically_bounded_surface); *)
FUNCTION gbsf_check_curve (cv:representation_item):BOOLEAN;
(* This function varifies the validity of a curve in the context of a
```

```
geometrically bounded surface model. Representation items are
  valid input, however, they are supposed to be curves; otherwise
  this function will return false.
*)
(* complex subtypes of curve that are both bounded curve and one of
   conic, curve_replica, line, or offset_curve_3d are not valid
*)
IF SIZEOF (['AP203_WITH_PDQ_SCHEMA.BOUNDED_CURVE',
  'AP203_WITH_PDQ_SCHEMA.CONIC',
  'AP203_WITH_PDQ_SCHEMA.CURVE_REPLICA',
  'AP203_WITH_PDQ_SCHEMA.LINE',
  'AP203_WITH_PDQ_SCHEMA.OFFSET_CURVE_3D']
  * TYPEOF(cv)) > 1 THEN RETURN(FALSE);
 END IF;
(* circle, ellipse, and trimmed curve are valid curves; they
   are bounded per definition
*)
IF SIZEOF (['AP203 WITH PDQ SCHEMA.CIRCLE',
'AP203_WITH_PDQ_SCHEMA.ELLIPSE',
'AP203 WITH PDQ SCHEMA.TRIMMED CURVE']
* TYPEOF(cv)) = 1
THEN RETURN(TRUE);
 ELSE
  (* b_spline_curves shall not self-intersect
  IF (('AP203_WITH_PDQ_SCHEMA.B_SPLINE_CURVE'
    IN TYPEOF(cv)) AND
    (cv\(\frac{1}{2}\)b spline curve.self intersect = FALSE) OR
    (cv\(\frac{1}{2}\)b_spline_curve.self_intersect = UNKNOWN))
  THEN RETURN(TRUE);
  ELSE
    (* a composite curve shall not self-intersect and all of
```

its segments shall reference valid curves

```
*)
IF (('AP203 WITH PDQ SCHEMA.COMPOSITE CURVE'
IN TYPEOF(cv)) AND
  (cv\(\frac{1}{2}\)composite curve.self intersect = FALSE) OR
  (cv\(\frac{1}{2}\)composite curve.self intersect = UNKNOWN))
THEN
  RETURN (SIZEOF (QUERY (seg <* cv\u224\u222composite_curve.segments |
  NOT (gbsf_check_curve(seg.parent_curve)))) = 0);
ELSE
  (* a curve replica shall reference a valid curve
  IF 'AP203 WITH PDQ SCHEMA.CURVE REPLICA'
    IN TYPEOF(cv) THEN
    RETURN (gbsf_check_curve (cv\u214curve_replica.parent_curve));
  ELSE
    (* an offset curve 3d shall not self-intersect and shall
       reference a valid curve; a polyline is not a valid
       basis_curve
    *)
    IF (('AP203_WITH_PDQ_SCHEMA.OFFSET_CURVE_3D'
      IN TYPEOF(cv))
      AND
      ((cv\subset_curve_3d.self_intersect = FALSE) OR
      (cv\subset_curve_3d.self_intersect = UNKNOWN))
      AND
      (NOT ('AP203_WITH_PDQ_SCHEMA.POLYLINE'
      IN TYPEOF(cv\subseteq offset_curve_3d.basis_curve)))) THEN RETURN
      (gbsf check curve (cv¥offset curve 3d.basis curve));
    ELSE
      (* a pcurve shall reference a valid curve and a valid
         basis_surface
      *)
      IF 'AP203 WITH PDQ SCHEMA.PCURVE'
```

```
IN TYPEOF(cv) THEN RETURN ((gbsf check curve
 (cv\pcurve.reference_to_curve\presentation.items[1]))
 AND
 (gbsf check surface (cv\pcurve.basis surface)));
ELSE
 (* a polyline shall have at least 3 points
 *)
 IF 'AP203_WITH_PDQ_SCHEMA.POLYLINE'
    IN TYPEOF(cv) THEN
   IF (SIZEOF (cv\polyline.points) >= 3)
     THEN RETURN (TRUE);
    END IF;
 ELSE
    (* a surface curve references a curve 3d and one or two
      pcurves or one or two surfaces or one of each;
       all of these references shall be valid
    *)
    IF 'AP203_WITH_PDQ_SCHEMA.SURFACE_CURVE'
     IN TYPEOF(cv) THEN
     (* if the curve reference is correct, check also the rest
     *)
     IF gbsf_check_curve (cv\surface_curve.curve_3d) THEN
       REPEAT i = 1 TO SIZEOF
       (cv\surface_curve.associated_geometry);
       (* do for one or two associated_geometrys:
       *)
       IF 'AP203_WITH_PDQ_SCHEMA.SURFACE' IN TYPEOF
         (cv\surface_curve.associated_geometry[i]) THEN
         IF NOT gbsf check surface
           (cv\surface_curve.associated_geometry[i]) THEN
           RETURN(FALSE);
         END IF;
```

```
ELSE
                 IF \ 'AP203\_WITH\_PDQ\_SCHEMA.PCURVE' \ IN \ TYPEOF
                  IF\ NOT\ gbsf\_check\_curve
                    (cv\surface_curve.associated_geometry[i]) THEN
                    RETURN(FALSE);
                  END_IF;
                 END_IF;
               END_IF;
               END_REPEAT;
               RETURN(TRUE);
              END_IF;
            END_IF;
          END_IF;
        END_IF;
       END_IF;
     END_IF;
    END_IF;
  END_IF;
END_IF;
(* FALSE is returned if the input parameter cv is not a valid curve.
RETURN(FALSE);
END_FUNCTION;
                           •••• 中略 ••••
END_SCHEMA;
```

付録 3. 検査対象ファイル「TorsionSpringLeft.stp」

以下に、品質項目 short_length_edge の検査で使用した Part 21 ファイル「TorsionSpringLeft.stp」(【エリジオン】) の抜粋を示す。

```
ISO-10303-21;
HEADER;
 * Created by Elysium, Inc.
 * www.elysiuminc.com
 * CADporter Product Family
/* OPTION: using older ISO 10303-21:1994 (E) spec */
FILE_DESCRIPTION(
/* description */ ('AP203 STEP file'),
/* implementation_level */ '1');
FILE_NAME(
/* name */ 'TorsionSpringLeft',
/* time_stamp */ '2007-02-20T20:28:19+09:00',
/* author */ ("),
/* organization */ ("),
/* preprocessor_version */ 'CADporter STEP Translator v1.1.0',
/* originating_system */ 'STEP',
/* authorisation */ ");
FILE_SCHEMA (('CONFIG_CONTROL_DESIGN'));
ENDSEC;
DATA;
#10=SHAPE_REPRESENTATION_RELATIONSHIP($,$,#7434,#11);
#11=ADVANCED_BREP_SHAPE_REPRESENTATION($,(#434),#7445);
#12=(
BOUNDED_SURFACE()
B_SPLINE_SURFACE(3,3,(#5038,#5039,#5040,#5041,#5042,#5043,#5044,#5045,#504
```

6,

 $\#5047,\#5048,\#5049,\#5050,\#5051,\#5052,\#5053,\#5054,\#5055,\#5056,\#5057,\#5058,\\ \#5059,\#5060,\#5061,\#5062,\#5063,\#5064,\#5065,\#5066,\#5067,\#5068,\#5069,\#5070,$

•••• 中略 •••••

#6979, #6980, #6981, #6982, #6983, #6984, #6985, #6986, #6987)),. UNSPECIFIED.,. T., .F.,. U.)

····· 中略 ·····

 $0.973959183584058, 0.979165558166278, 0.9843727349794, 0.989585835284034, 0.9948\\ 03851115241,$

1.)..UNSPECIFIED.)

GEOMETRIC_REPRESENTATION_ITEM()

•••• 中略 •••••

);

#13=FACE_BOUND(\$,#423,.T.);

#14=FACE_BOUND(\$,#425,.T.);

#15=PLANE(",#7429);

#16=PLANE(",#7430);

#17=PLANE(",#7431);

#18=PLANE(",#7432);

#19=CIRCLE(",#7421,2.38760000000006);

#20=CIRCLE(",#7423,2.40665);

```
#21=CIRCLE(",#7424,2.40665);
#22=CIRCLE(",#7425,2.38760000000001);
#23=CIRCLE(",#7426,2.40665);
#24=CIRCLE(",#7427,2.40665);
#25=CYLINDRICAL_SURFACE(",#7420,2.38760000000000);
#26=CYLINDRICAL_SURFACE(",#7422,2.40665);
#27=CYLINDRICAL_SURFACE(",#7428,2.3876);
#28=CYLINDRICAL SURFACE(",#7433,2.3876);
#29=ADVANCED_FACE('1',(#47),#431,.T.);
#30=ADVANCED_FACE('2',(#48),#431,.T.);
#31=ADVANCED FACE('3',(#49),#432,.T.);
#32=ADVANCED_FACE('4',(#50),#432,.T.);
#33=ADVANCED_FACE('5',(#51),#25,.T.);
#34=ADVANCED_FACE('6',(#52),#26,.T.);
#35=ADVANCED_FACE('7',(#53),#25,.T.);
#36=ADVANCED FACE('8',(#54),#26,.T.);
#37=ADVANCED_FACE('9',(#55),#27,.T.);
#38=ADVANCED_FACE('10',(#56),#27,.T.);
#39=ADVANCED_FACE('11',(#57),#15,.T.);
#40=ADVANCED_FACE('12',(#58,#13),#16,.T.);
#41=ADVANCED_FACE('13',(#59,#14),#17,.T.);
#42=ADVANCED_FACE('14',(#60),#18,.T.);
#43=ADVANCED_FACE('15',(#61),#28,.T.);
#44=ADVANCED_FACE('16',(#62),#12,.T.);
#45=ADVANCED_FACE('17',(#63),#28,.T.);
#46=ADVANCED_FACE('18',(#64),#12,.T.);
#47=FACE_OUTER_BOUND($,#411,.T.);
#48=FACE OUTER BOUND($,#412,.T.);
#49=FACE_OUTER_BOUND($,#413,.T.);
#50=FACE_OUTER_BOUND($,#414,.T.);
#51=FACE_OUTER_BOUND($,#415,.T.);
#52=FACE_OUTER_BOUND($,#416,.T.);
#53=FACE OUTER BOUND($,#417,.T.);
#54=FACE_OUTER_BOUND($,#418,.T.);
#55=FACE_OUTER_BOUND($,#419,.T.);
#56=FACE OUTER BOUND($,#420,.T.);
```

#57=FACE_OUTER_BOUND(\$,#421,.T.); #58=FACE_OUTER_BOUND(\$,#422,.T.); #59=FACE_OUTER_BOUND(\$,#424,.T.); #60=FACE_OUTER_BOUND(\$,#426,.T.); #61=FACE_OUTER_BOUND(\$,#427,.T.); #62=FACE_OUTER_BOUND(\$,#428,.T.); #63=FACE_OUTER_BOUND(\$,#429,.T.); #64=FACE_OUTER_BOUND(\$,#430,.T.);

•••• 中略 ••••

#87=ORIENTED_EDGE(\$,*,*,#229,.T.);
#88=ORIENTED_EDGE(\$,*,*,#230,.T.);
#89=ORIENTED_EDGE(\$,*,*,#231,.T.);
#90=ORIENTED_EDGE(\$,*,*,#232,.T.);
#91=ORIENTED_EDGE(\$,*,*,#233,.F.);
#92=ORIENTED_EDGE(\$,*,*,#234,.F.);
#93=ORIENTED_EDGE(\$,*,*,#235,.T.);
#94=ORIENTED_EDGE(\$,*,*,#236,.T.);
#95=ORIENTED_EDGE(\$,*,*,#236,.T.);
#95=ORIENTED_EDGE(\$,*,*,#237,.T.);
#96=ORIENTED_EDGE(\$,*,*,#238,.T.);
#97=ORIENTED_EDGE(\$,*,*,#239,.T.);
#98=ORIENTED_EDGE(\$,*,*,#240,.F.);
#99=ORIENTED_EDGE(\$,*,*,#241,.F.);
#100=ORIENTED_EDGE(\$,*,*,#242,.T.);

•••• 中略 ••••

#207=ORIENTED_EDGE(\$,*,*,#298,.F.); #208=ORIENTED_EDGE(\$,*,*,#237,.F.); #209=ORIENTED_EDGE(\$,*,*,#236,.F.); #210=ORIENTED_EDGE(\$,*,*,#235,.F.); #211=ORIENTED_EDGE(\$,*,*,#234,.T.); #212=ORIENTED_EDGE(\$,*,*,#233,.T.); #213=ORIENTED_EDGE(\$,*,*,#232,.F.); #214=ORIENTED_EDGE(\$,*,*,#254,.F.); #215=ORIENTED_EDGE(\$,*,*,#253,.T.);
#216=ORIENTED_EDGE(\$,*,*,#252,.T.);
#217=ORIENTED_EDGE(\$,*,*,#251,.F.);
#218=ORIENTED_EDGE(\$,*,*,#250,.F.);
#219=ORIENTED_EDGE(\$,*,*,#249,.F.);
#220=ORIENTED_EDGE(\$,*,*,#248,.F.);
#221=ORIENTED_EDGE(\$,*,*,#247,.F.);
#222=ORIENTED_EDGE(\$,*,*,#246,.F.);
#223=ORIENTED_EDGE(\$,*,*,#245,.T.);
#224=ORIENTED_EDGE(\$,*,*,#244,.T.);
#224=ORIENTED_EDGE(\$,*,*,#244,.T.);
#225=ORIENTED_EDGE(\$,*,*,#243,.F.);
#226=ORIENTED_EDGE(\$,*,*,#243,.F.);
#227=ORIENTED_EDGE(\$,*,*,#261,.F.);
#227=ORIENTED_EDGE(\$,*,*,#261,.F.);
#228=ORIENTED_EDGE(\$,*,*,#277,.F.);

····· 中略 ·····

#231=EDGE_CURVE('3',#356,#357,#302,.T.); #232=EDGE_CURVE('4',#357,#358,#303,.T.); #233=EDGE_CURVE('5',#359,#358,#304,.T.);

•••• 中略 •••••

#242=EDGE_CURVE('14',#367,#354,#76,.T.); #243=EDGE_CURVE('15',#354,#368,#77,.T.); #244=EDGE_CURVE('16',#369,#368,#313,.T.);

•••• 中略 •••••

#253=EDGE_CURVE('25',#378,#377,#322,.T.); #254=EDGE_CURVE('26',#378,#357,#78,.T.); #255=EDGE_CURVE('27',#356,#355,#323,.T.);

•••• 中略 •••••

#411=EDGE LOOP(\$.(#87,#88,#89,#90,#91,#92,#93,#94,#95,#96,#97,#98,#99,#100));

#412=EDGE_LOOP(\$,(#101,#102,#103,#104,#105,#106,#107,#108,#109,#110,#111, #112,#113,#114,#115));

•••• 中略 •••••

#429=EDGE_LOOP(\$,(#203,#204,#205,#206));

#430=EDGE_LOOP(\$,(#207,#208,#209,#210,#211,#212,#213,#214,#215,#216,#217, #218,#219,#220,#221,#222,#223,#224,#225,#226,#227,#228));

#431=B_SPLINE_SURFACE_WITH_KNOTS(",3,3,((#436,#437,#438,#439,#440,#441, #442,#443,#444,#445,#446,#447,#448,#449,#450,#451,#452,#453,#454,#455,#456,

•••• 中略 •••••

 $0.709011477025894, 0.757509564188245, 0.806007651350596, 0.854505738512947, \\0.903003825675298, 0.951501912837649, 1.),. UNSPECIFIED.); \\#432=B_SPLINE_SURFACE_WITH_KNOTS(",3,3,((#2666,#2667,#2668,#2669,#2670, #2671,#2672,#2673,#2674,#2675,#2676,#2677,#2678,#2679,#2680,#2681,#2682,$

•••• 中略 •••••

 $0.709011477025895, 0.757509564188246, 0.806007651350597, 0.854505738512948, \\ 0.903003825675298, 0.951501912837649, 1.), UNSPECIFIED.);$

#433=CLOSED_SHELL(\$,(#29,#30,#31,#32,#33,#34,#35,#36,#37,#38,#39,#40,#41, #42,#43,#44,#45,#46));

#434=MANIFOLD_SOLID_BREP(\$,#433);

#435=CARTESIAN_POINT(\$,(0.,0.,0.));

#436=CARTESIAN_POINT(\$,(12.2883469020702,-281.040858190387,-1039.57636399 833));

····· 中略 ·····

#7419=AXIS2_PLACEMENT_3D(",#435,#7379,#7378); #7420=AXIS2_PLACEMENT_3D(",#4872,#7384,#7383);

```
#7434=SHAPE_REPRESENTATION($,(#7419),#7445);
 #7435=(
 NAMED_UNIT(*)
 SI_UNIT($,.STERADIAN.)
 SOLID_ANGLE_UNIT()
 );
 #7436=(
 NAMED_UNIT(*)
 PLANE_ANGLE_UNIT()
 SI_UNIT($,.RADIAN.)
 );
 #7437=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.01
74532925199433),
 #7436);
 #7438=(
 CONVERSION_BASED_UNIT('DEGREE',#7437)
 NAMED_UNIT(#7441)
 PLANE_ANGLE_UNIT()
 );
 #7439=(
 LENGTH_UNIT()
 NAMED_UNIT(*)
 SI_UNIT(.MILLI.,.METRE.)
 );
 #7440=DIMENSIONAL_EXPONENTS(1.,0.,0.,0.,0.,0.,0.,0.);
 #7441=DIMENSIONAL_EXPONENTS(0.,0.,0.,0.,0.,0.,0.,0.);
 #7442=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(1.),#7439);
 #7443=(
 CONVERSION_BASED_UNIT('MM',#7442)
 LENGTH_UNIT()
 NAMED_UNIT(#7440)
 #7444=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#7443
  'point coincidence tolerance',
  'maximum model space distance between geometric entities at asserted co
```

```
nnectivities');
 #7445=(
 GEOMETRIC_REPRESENTATION_CONTEXT(3)
 GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#7444))
 GLOBAL_UNIT_ASSIGNED_CONTEXT((#7443,#7438,#7435))
 REPRESENTATION_CONTEXT('3','3D')
 #7446=SHAPE_DEFINITION_REPRESENTATION(#7447,#7434);
 #7447=PRODUCT_DEFINITION_SHAPE($,'Torsion Spring Left Shape',#7449);
 #7448=DESIGN_CONTEXT($,#7487,'design');
 #7449=PRODUCT DEFINITION('design', $, #7465, #7448);
 #7450=APPROVAL_DATE_TIME(#7477,#7455);
 #7451=APPROVAL_ROLE('approver');
 #7452=APPROVAL PERSON ORGANIZATION(#7474,#7455,#7451);
 #7453=CC_DESIGN_APPROVAL(#7455,(#7464,#7465,#7449));
 #7454=APPROVAL_STATUS('approved');
 #7455=APPROVAL(#7454,'approved for release');
 #7456=DATE_TIME_ROLE('classification_date');
 #7457=DATE_TIME_ROLE('sign_off_date');
 #7458=DATE_TIME_ROLE('creation_date');
 #7459=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#7477,#7456,(#7464));
 #7460=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#7477,#7457,(#7452));
 #7461=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#7477,#7458,(#7449));
 #7462=CC_DESIGN_SECURITY_CLASSIFICATION(#7464,(#7465));
 #7463=SECURITY_CLASSIFICATION_LEVEL('unclassified');
 #7464=SECURITY_CLASSIFICATION(",",#7463);
 #7465=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE(",",#7
485..MADE.);
 #7466=PERSON_AND_ORGANIZATION_ROLE('design_owner');
 #7467=PERSON_AND_ORGANIZATION_ROLE('creator');
 #7468=PERSON_AND_ORGANIZATION_ROLE('design_supplier');
 #7469=PERSON_AND_ORGANIZATION_ROLE('classification_officer');
 #7470=CC DESIGN PERSON AND ORGANIZATION ASSIGNMENT(#7474.#7466
,(#7485));
 #7471=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#7474,#7467
,(#7465,#7449));
```

```
#7472=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#7474,#7468
,(#7465));
 #7473=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#7474,#7469
,(#7464));
 #7474=PERSON_AND_ORGANIZATION(#7476,#7475);
 #7475=ORGANIZATION($,",");
 #7476=PERSON($,",",$,$,$);
 #7477=DATE_AND_TIME(#7480,#7478);
 #7478=LOCAL_TIME(20,28,19.,#7479);
 #7479=COORDINATED_UNIVERSAL_TIME_OFFSET(9,0,.AHEAD.);
 #7480=CALENDAR_DATE(2007,20,2);
 #7481=PRODUCT_CATEGORY_RELATIONSHIP($,$,#7482,#7483);
 #7482=PRODUCT_CATEGORY('part',$);
 #7483=PRODUCT_RELATED_PRODUCT_CATEGORY('detail',$,(#7485));
 \#7484 = \texttt{MECHANICAL\_CONTEXT(",\#7487,'mechanical');}
 #7485=PRODUCT('Torsion Spring Left', 'Torsion Spring Left', $, (#7484));
 #7486=APPLICATION_PROTOCOL_DEFINITION('international standard',
 'config_control_design',1994,#7487);
 #7487=APPLICATION_CONTEXT(
  'configuration controlled 3D designs of mechanical parts and assemblies
 ');
  ENDSEC;
 END-ISO-10303-21;
```

提供データ

【エリジオン】データは、株式会社エリジオン殿ご提供

付録 4. 検査対象ファイル「zone1.stp」

以下に、品質項目 gap_between_edge_and_base_surface の検査で使用した Part 21 ファイル「zone1.stp」(【エリジオン】) の抜粋を示す。

```
ISO-10303-21;
HEADER;
 * Created by Elysium, Inc.
 * www.elysiuminc.com
 * CADporter Product Family
/* OPTION: using older ISO 10303-21:1994 (E) spec */
FILE_DESCRIPTION(
/* description */ ('AP203 STEP file'),
/* implementation_level */ '1');
FILE_NAME(
/* name */ 'zone1',
/* time_stamp */ '2007-02-20T19:59:34+09:00',
/* author */ ("),
/* organization */ ("),
/* preprocessor_version */ 'CADporter STEP Translator v1.1.0',
/* originating_system */ 'STEP',
/* authorisation */ ");
FILE_SCHEMA (('CONFIG_CONTROL_DESIGN'));
ENDSEC;
DATA;
#10=SHAPE_REPRESENTATION_RELATIONSHIP($,$,#11063,#11);
#11=ADVANCED_BREP_SHAPE_REPRESENTATION($,(#689),#11074);
```

#32=B_SPLINE_SURFACE_WITH_KNOTS(",2,3,((#5564,#5565,#5566,#5567,#5568,#5569,

#5570,#5571,#5572,#5573,#5574),(#5575,#5576,#5577,#5578,#5579,#5580,#5581, #5582,#5583,#5584,#5585),(#5586,#5587,#5588,#5589,#5590,#5591,#5592,#5593, #5594,#5595,#5596),(#5597,#5598,#5599,#5600,#5601,#5602,#5603,#5604,#5605, #5606,#5607),(#5608,#5609,#5610,#5611,#5612,#5613,#5614,#5615,#5616,#5617, #5618)),.UNSPECIFIED.,.F.,.F.,.U.,(3,2,3),(4,1,1,1,1,1,1,1,1,4),(0.,0.994035721844947, 1.),(0.,0.125,0.25,0.375,0.5,0.625,0.75,0.875,1.),.UNSPECIFIED.);

#33=B_SPLINE_SURFACE_WITH_KNOTS(",3,3,((#5637,#5638,#5639,#5640,#5641,#5642,

#5643,#5644,#5645,#5646,#5647,#5648,#5649,#5650),(#5651,#5652,#5653,#5654, #5655,#5656,#5657,#5658,#5659,#5660,#5661,#5662,#5663,#5664),(#5665,#5666, #5667,#5668,#5669,#5670,#5671,#5672,#5673,#5674,#5675,#5676,#5677,#5678),

•••• 中略 ••••

- 0.111801272667498, 0.14709204438191, 0.182382816096322, 0.217673587810733, 0.252964359525146,
 - 0.288255131239558, 0.323545902953969, 0.358836674668381, 0.394127446382793,
 - 0.429418218097205, 0.464708989811617, 0.499999761526029, 0.535290533240441,
 - 0.570581304954853, 0.605872076669264, 0.641162848383677, 0.676453620098088,
- $0.7117443918125, 0.747035163526912, 0.782325935241324, 0.817616706955736, 0.8529\\0.7478670148,$

•••• 中略 •••••

#37=B_SPLINE_SURFACE_WITH_KNOTS(",3,3,((#8671,#8672,#8673,#8674,#8675,#8676,

#8677,#8678,#8679,#8680,#8681,#8682,#8683,#8684,#8685,#8686,#8687),(#8688, #8689,#8690,#8691,#8692,#8693,#8694,#8695,#8696,#8697,#8698,#8699,#8700, #8701,#8702,#8703,#8704),(#8705,#8706,#8707,#8708,#8709,#8710,#8711,#8712,

•••• 中略 •••••

- 1.),(0.,0.0714285714285714,0.142857142857143,0.214285714285715,0.2857142857142 86,
 - 0.357142857142857,0.428571428571428,0.5,0.571428571428571,0.642857142857143, 0.714285714285714,0.785714285714286,0.857142857142857,0.928571428571429, 1.),.UNSPECIFIED.);

#38=B_SPLINE_SURFACE_WITH_KNOTS(",3,3,((#9286,#9287,#9288,#9289,#9290,#9291,

•••• 中略 •••••

#70=ADVANCED FACE('1',(#105),#677,.T.);

```
#71=ADVANCED_FACE('2',(#106),#64,.T.);
#72=ADVANCED FACE('3',(#107),#42,.T.);
#73=ADVANCED_FACE('4',(#108),#65,.T.);
#74=ADVANCED_FACE('5',(#109),#66,.T.);
#75=ADVANCED_FACE('6',(#110),#26,.T.);
#76=ADVANCED_FACE('7',(#111,#24),#27,.T.);
#77=ADVANCED_FACE('8',(#112),#28,.T.);
#78=ADVANCED_FACE('9',(#113),#29,.T.);
#79=ADVANCED_FACE('10',(#114),#678,.T.);
#80=ADVANCED_FACE('11',(#115),#679,.T.);
#81=ADVANCED FACE('12',(#116),#67,.T.);
#82=ADVANCED_FACE('13',(#117),#30,.T.);
#83=ADVANCED_FACE('14',(#118),#43,.T.);
#84=ADVANCED_FACE('15',(#119),#68,.T.);
#85=ADVANCED_FACE('16',(#120),#69,.T.);
#86=ADVANCED_FACE('17',(#121),#31,.T.);
#87=ADVANCED_FACE('18',(#122),#680,.T.);
#88=ADVANCED_FACE('19',(#123),#32,.T.);
#89=ADVANCED_FACE('20',(#124),#33,.T.);
#90=ADVANCED_FACE('21',(#125),#34,.T.);
#91=ADVANCED_FACE('22',(#126),#35,.T.);
#92=ADVANCED_FACE('23',(#127),#36,.T.);
#93=ADVANCED_FACE('24',(#128),#37,.T.);
#94=ADVANCED_FACE('25',(#129,#25),#38,.T.);
#95=ADVANCED_FACE('26',(#130),#681,.T.);
#96=ADVANCED_FACE('27',(#131),#39,.T.);
#97=ADVANCED_FACE('28',(#132),#40,.T.);
#98=ADVANCED FACE('29',(#133),#682,.T.);
#99=ADVANCED_FACE('30',(#134),#683,.T.);
#100=ADVANCED_FACE('31',(#135),#684,.T.);
#101=ADVANCED_FACE('32',(#136),#685,.T.);
#102=ADVANCED_FACE('33',(#137),#686,.T.);
#103=ADVANCED FACE('34',(#138),#687,.T.);
#104=ADVANCED_FACE('35',(#139),#41,.T.);
#105=FACE_OUTER_BOUND($,#640,.T.);
#106=FACE_OUTER_BOUND($,#641,.T.);
```

#107=FACE_OUTER_BOUND(\$,#642,.T.); #108=FACE_OUTER_BOUND(\$,#643,.T.); #109=FACE_OUTER_BOUND(\$,#644,.T.); #110=FACE_OUTER_BOUND(\$,#645,.T.); #111=FACE_OUTER_BOUND(\$,#646,.T.); #112=FACE_OUTER_BOUND(\$,#648,.T.); #113=FACE_OUTER_BOUND(\$,#649,.T.); #114=FACE_OUTER_BOUND(\$,#650,.T.); #115=FACE_OUTER_BOUND(\$,#651,.T.); #116=FACE_OUTER_BOUND(\$,#652,.T.); #117=FACE_OUTER_BOUND(\$,#653,.T.); #118=FACE_OUTER_BOUND(\$,#654,.T.); #119=FACE_OUTER_BOUND(\$,#655,.T.); #120=FACE_OUTER_BOUND(\$,#656,.T.); #121=FACE_OUTER_BOUND(\$,#657,.T.); #122=FACE_OUTER_BOUND(\$,#658,.T.); #123=FACE_OUTER_BOUND(\$,#659,.T.); #124=FACE_OUTER_BOUND(\$,#660,.T.); #125=FACE_OUTER_BOUND(\$,#661,.T.); #126=FACE_OUTER_BOUND(\$,#662,.T.); #127=FACE_OUTER_BOUND(\$,#663,.T.); #128=FACE_OUTER_BOUND(\$,#664,.T.); #129=FACE_OUTER_BOUND(\$,#665,.T.); #130=FACE_OUTER_BOUND(\$,#667,.T.); #131=FACE_OUTER_BOUND(\$,#668,.T.); #132=FACE_OUTER_BOUND(\$,#669,.T.); #133=FACE_OUTER_BOUND(\$,#670,.T.); #134=FACE OUTER BOUND(\$,#671,.T.); #135=FACE_OUTER_BOUND(\$,#672,.T.); #136=FACE_OUTER_BOUND(\$,#673,.T.); #137=FACE_OUTER_BOUND(\$,#674,.T.); #138=FACE_OUTER_BOUND(\$,#675,.T.); #139=FACE OUTER BOUND(\$,#676,.T.);

#238=ORIENTED_EDGE(\$,*,*,#431,.F.); #239=ORIENTED_EDGE(\$,*,*,#430,.T.); #240=ORIENTED_EDGE(\$,*,*,#432,.F.); #241=ORIENTED_EDGE(\$,*,*,#433,.F.); #242=ORIENTED_EDGE(\$,*,*,#434,.F.); #243=ORIENTED_EDGE(\$,*,*,#435,.F.); #244=ORIENTED_EDGE(\$,*,*,#436,.F.); #245=ORIENTED_EDGE(\$,*,*,#437,.F.); #246=ORIENTED_EDGE(\$,*,*,#438,.F.); #247=ORIENTED_EDGE(\$,*,*,#439,.F.); #248=ORIENTED_EDGE(\$,*,*,#439,.F.); #248=ORIENTED_EDGE(\$,*,*,#440,.F.);

•••• 中略 ••••

#265=ORIENTED_EDGE(\$,*,*,#442,F.);
#266=ORIENTED_EDGE(\$,*,*,#383,T.);
#267=ORIENTED_EDGE(\$,*,*,#382,T.);
#268=ORIENTED_EDGE(\$,*,*,#381,.T.);
#269=ORIENTED_EDGE(\$,*,*,#380,.T.);
#270=ORIENTED_EDGE(\$,*,*,#379,.T.);
#271=ORIENTED_EDGE(\$,*,*,#444,F.);
#272=ORIENTED_EDGE(\$,*,*,#445,.T.);
#273=ORIENTED_EDGE(\$,*,*,#443,F.);

····· 中略 ·····

#432=EDGE_CURVE('75',#622,#620,#530,.T.); #433=EDGE_CURVE('76',#623,#622,#531,.T.); #434=EDGE_CURVE('77',#624,#623,#532,.T.);

•••• 中略 ••••

#442=EDGE_CURVE('85',#564,#620,#540,.T.); #443=EDGE_CURVE('86',#586,#622,#541,.T.); #444=EDGE_CURVE('87',#630,#580,#542,.T.);

•••• 中略 ••••

#659=EDGE_LOOP(\$,(#234,#235,#236,#237));

#660=EDGE LOOP(\$,(#238,#239,#240,#241,#242,#243,#244,#245,#246,#247,#248));

#661=EDGE_LOOP(\$,(#249,#250,#251,#252));

#662=EDGE_LOOP(\$,(#253,#254,#255,#256));

#663=EDGE_LOOP(\$,(#257,#258,#259,#260,#261,#262,#263,#264,#265));

#664=EDGE LOOP(\$.(#266,#267,#268,#269,#270,#271,#272,#273));

#665=EDGE_LOOP(\$,(#274,#275,#276,#277,#278,#279,#280,#281,#282,#283,#284, #285,#286,#287,#288,#289,#290,#291,#292,#293,#294));

•••• 中略 •••••

#688=CLOSED_SHELL(\$,(#70,#71,#72,#73,#74,#75,#76,#77,#78,#79,#80,#81,#82, #83,#84,#85,#86,#87,#88,#89,#90,#91,#92,#93,#94,#95,#96,#97,#98,#99,#100, #101,#102,#103,#104));

#689=MANIFOLD_SOLID_BREP(\$,#688);

#690=CARTESIAN_POINT(\$,(0.,0.,0.));

#691=CARTESIAN_POINT(\$,(-390.261926941289,47.5374614553585,523.445241240781));

•••• 中略 •••••

#10999=CARTESIAN_POINT(\$,(410.109597775882,33.8381513474656,-165.546069818374));

#11000=CARTESIAN_POINT(\$,(411.486265426848,47.5374614553585,-157.63676943 9869));

#11001=DIRECTION(",(1.,0.,0.));

#11002=DIRECTION(",(0.,0.,1.));

•••• 中略 •••••

#11044=DIRECTION(",(0.,0.,1.));

#11046=DIRECTION(",(-1.96039722621979E-015,0.49999999999997,-0.86602540378 444));

#11047=AXIS2_PLACEMENT_3D(",#690,#11002,#11001);

```
#11063=SHAPE_REPRESENTATION($,(#11047),#11074);
 #11064=(
 NAMED_UNIT(*)
 SI_UNIT($,.STERADIAN.)
 SOLID_ANGLE_UNIT()
 );
 #11065=(
 NAMED_UNIT(*)
 PLANE_ANGLE_UNIT()
 SI_UNIT($,.RADIAN.)
 );
 #11066=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.0
174532925199433),
 #11065);
 #11067=(
 CONVERSION_BASED_UNIT('DEGREE',#11066)
 NAMED_UNIT(#11070)
 PLANE_ANGLE_UNIT()
 );
 #11068=(
 LENGTH_UNIT()
 NAMED_UNIT(*)
 SI_UNIT(.MILLI.,.METRE.)
 );
 #11069=DIMENSIONAL_EXPONENTS(1,,0,,0,,0,,0,,0,,0,,0,);
 #11070=DIMENSIONAL_EXPONENTS(0.,0.,0.,0.,0.,0.,0.);
 #11071=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(1.),#11068);
 #11072=(
 CONVERSION_BASED_UNIT('MM',#11071)
 LENGTH UNIT()
 NAMED_UNIT(#11069)
 );
 #11073=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#110
```

```
72,
 'point coincidence tolerance',
 'maximum model space distance between geometric entities at asserted co
 nnectivities');
 #11074=(
 GEOMETRIC_REPRESENTATION_CONTEXT(3)
 GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#11073))
 GLOBAL UNIT ASSIGNED CONTEXT((#11072,#11067,#11064))
 REPRESENTATION_CONTEXT('3','3D')
 );
 #11075=SHAPE DEFINITION REPRESENTATION(#11076,#11063);
 #11076=PRODUCT_DEFINITION_SHAPE($,'zone1 Shape',#11078);
 #11077=DESIGN_CONTEXT($,#11116,'design');
 #11078=PRODUCT DEFINITION('design', $, #11094, #11077);
 #11079=APPROVAL_DATE_TIME(#11106,#11084);
 #11080=APPROVAL_ROLE('approver');
 #11081=APPROVAL_PERSON_ORGANIZATION(#11103,#11084,#11080);
 #11082=CC_DESIGN_APPROVAL(#11084,(#11093,#11094,#11078));
 #11083=APPROVAL_STATUS('approved');
 #11084=APPROVAL(#11083,'approved for release');
 #11085=DATE_TIME_ROLE('classification_date');
 #11086=DATE_TIME_ROLE('sign_off_date');
 #11087=DATE_TIME_ROLE('creation_date');
 #11088=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#11106,#11085,(#11093));
 #11089=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#11106,#11086,(#11081));
 #11090=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#11106,#11087,(#11078));
 #11091=CC_DESIGN_SECURITY_CLASSIFICATION(#11093,(#11094));
 #11092=SECURITY CLASSIFICATION LEVEL('unclassified');
 #11093=SECURITY_CLASSIFICATION(",",#11092);
 #11094=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE(",",#
11114,.MADE.);
 #11095=PERSON_AND_ORGANIZATION_ROLE('design_owner');
 #11096=PERSON_AND_ORGANIZATION_ROLE('creator');
 #11097=PERSON_AND_ORGANIZATION_ROLE('design_supplier');
 #11098=PERSON_AND_ORGANIZATION_ROLE('classification_officer');
 #11099=CC DESIGN PERSON AND ORGANIZATION ASSIGNMENT(#11103.#11
```

```
095,(#11114));
  #11100=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#11103,#11
096,(#11094,
 #11078));
 #11101=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#11103,#11
097,(#11094));
  #11102=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#11103,#11
098,(#11093));
 #11103=PERSON_AND_ORGANIZATION(#11105,#11104);
 #11104=ORGANIZATION($,",");
 #11105=PERSON($,",",$,$,$);
 #11106=DATE_AND_TIME(#11109,#11107);
 #11107=LOCAL_TIME(19,59,34.,#11108);
 #11108=COORDINATED_UNIVERSAL_TIME_OFFSET(9,0,.AHEAD.);
 #11109=CALENDAR_DATE(2007,20,2);
 #11110=PRODUCT_CATEGORY_RELATIONSHIP($,$,#11111,#11112);
 #11111=PRODUCT_CATEGORY('part',$);
 #11112=PRODUCT_RELATED_PRODUCT_CATEGORY('detail', $, (#11114));
 #11113=MECHANICAL_CONTEXT(",#11116,'mechanical');
 #11114=PRODUCT('zone1','zone1',$,(#11113));
 #11115=APPLICATION_PROTOCOL_DEFINITION('international standard',
  'config_control_design',1994,#11116);
 #11116=APPLICATION_CONTEXT(
  'configuration controlled 3D designs of mechanical parts and assemblies
  ');
  ENDSEC;
  END-ISO-10303-21;
```

提供データ

【エリジオン】データは、株式会社エリジオン殿ご提供

禁無断転載

平成 18 年度 EC における国際連携の推進に関する調査研究 設計製造データの国際標準化に関する調査研究報告書 平成 1 9年 3月 発行

発行所 財団法人 日本情報処理開発協会

電子商取引推進センター

東京都港区芝公園三丁目5番8号

機械振興会館3階

TEL: 03 (3436) 7500

印刷所 ㈱三菱電機ドキュメンテクス

東京都中央区湊三丁目5番10号

TEL: 03 (5566) 0681

(本報告書は再生紙を使用しています。)

18-E005