

ビルドタイム/ランタイム・アプローチによる 設計プロセスの革新

製造業における開発プロセスでは、フロント・ローディングの導入による効率化が進んでいます。その一方で、設計プロセスの現状を見ると、CADツールが2次元から3次元に移行したにもかかわらず、図面や3次元モデルの清書システムの役割にとどまっているケースが少なくありません。また、設計手順ナビゲーション・システムなどを併用した設計ガイドを用い、設計を支援しているケースもありますが、そのノウハウは依然として設計者の頭の中にあり、手数を掛けてCADを操作してモデルを作っている状況です。そこで筆者のチームではCAD機能の向上に着目し、設計のノウハウをテンプレート化(ビルドタイム)し、実際の製品設計(ランタイム)の期間を大幅に短縮する手法を試みました。製品設計プロセスの分析/定義手法に、ITシステム開発手法と同様のメソドロジーを採用することで、従来あいまいだった設計プロセスを明確に分析・記述することが可能となり、設計のナレッジ化が実現しました。また、設計に要請される種々の要件をテンプレート内に含めることで、フロント・ローディングの多くが吸収でき、設計品質も大幅に向上しました。

本論文では、このビルドタイム/ランタイム・アプローチを中心に、新しいナレッジ・ベース設計のメソドロジーについて論述します。



日本アイ・ビー・エム株式会社 サービス事業
インダストリアル・ソリューション・センター
IBMディスティングイッシュト・エンジニア・ITアーキテクト、
IBMアカデミー会員
Distinguished Engineer / IT Architect
Member of the IBM Academy of Technology, Services Delivery-
Industrial, IBM Global Services Japan

長谷川 一彦 Kazuhiko Hasegawa

[プロフィール]

1969年日本アイ・ビー・エム入社。1977年から複数のCADシステム開発に従事し、CADのイネーブラーであるCADELを開発。1990年ころのホストからエンジニアリング・ワークステーションへのダウンサイジングに際しては次世代CADプロジェクトを推進し、次期CADはいかにあるべきかを研究した。その後日本アイ・ビー・エム東海SE(現日本アイ・ビー・エム中部ソリューション)出向を経て、現在はPLM分野のイノベーションに関する活動を行っている。ようやく次期CADプロジェクトで目指したことの一部分が現実となりつつあるが、CAD機能の高度化もさることながら、当時ではあまり言及されなかったメソドロジーの重要性を感じている。



日本アイ・ビー・エム中部ソリューション株式会社
第五開発部
専任ITスペシャリスト
Senior IT Specialist
PLM Solutions Group, Development No.5
IBM Global Services Japan Central Solution Company

浅羽 和弘 Kazuhiro Asaba

[プロフィール]

1999年日本アイ・ビー・エム中部ソリューション入社。前職であった“製造業における製品開発・設計”の経験を生かして、入社以来、革新的な開発プロセス・設計手法についての方法論構築と実践に取り組んでいる。現在はPLM CATIA関連のサービスビジネスを実施中である。設計者として製品開発プロセスの効率化・最適化は長年のテーマであったが、たまたま入社後のITトレーニングで学んだADSG手法にヒントを得て、その手法を製品開発プロセスの解析・再定義に適用することを発想し、今回の論文につなげることができた。

Reforms of Design Processes Employing the Build-Time, Run-Time Approach

The introduction of front-loading is increasingly bringing about higher levels of efficiency in development processes in manufacturing industries. On the other hand, looking at the current state of design processes, despite the fact that CAD tools have now moved from the second generation to the third generation, in many cases their use is restricted to the production of drawings and to a role as fair-copy production systems for three-dimensional models. There are cases where design is being supported through the use of design guides that make joint use of design procedure navigation systems, but the know-how in this regard exists only in the designer's head, and models are created at considerable effort through the use of CAD.

The writer's team focused on improvements in CAD functions. Having created templates for design know-how ("build time"), we experimented with methods aimed at drastically reducing the duration of actual product design ("run time"). By using the same methodology as that used in the development of IT systems for analysis and definition of product design processes, we became able to clearly analyze and describe design processes which had previously been vague, thereby succeeding in converting design into knowledge. Moreover, by including the various conditions required of design in the template, much of the front-loading could be absorbed and there was a consequent drastic improvement in design quality.

In this paper we describe the methodology for new knowledge-based design centering on this build time and run time approach.

1. はじめに

製造業におけるQ(Quality : 品質向上)、C(Cost : コスト削減)、D(Delivery : 期間短縮)は永遠の課題であり、最近では四つ目の要素として地球環境への配慮も加わっています。

QCD向上のキー・ポイントに、開発プロセスでのフロント・ローディングが挙げられます。これは生産技術・製造・保守サービスでの要件を設計の上流段階から考慮して、プロセス全体で手戻りを防ぎ、実際に物を作る試作以降のプロセスにおけるコスト最少化が狙いです。また全プロセスのデジタル化によるバーチャル・プロダクト化(コンピューター上における製品の仮想的な開発)と、デジタル・モックアップの有効利用による試作の最少化・省略が可能となります。

ところが設計プロセスの現状を見ると、CAD(Computer Aided Design : コンピューター支援設計)ツールが2次元から3次元に移行したにもかかわらず、図面や3次元モデルの清書システムの役割にとどまっているケースが少なくありません。また設計手順ナビゲーション・システムなどを併用した設計ガイドを用い、設計を支援しているケースもありますが、そのノウハウは依然として設計者の頭の中にあり、手数を掛けてCADを操作しモデルを作っている状況です。

そこで筆者のチームではCAD機能の向上に着目し、設計のノウハウをテンプレート化(ビルドタイム)して、実際の製品設計(ランタイム)の期間を大幅に短縮する手法を試みました。製品設計プロセスの分析 / 定義手法に、IT(Information Technology : 情報技術)システム開発手法と同様のメソッドロジーを採用することで、従来あいまいだった設計プロセスを明確に分析・記述することが可能となり、設計のナレッジ化が実現しました。また、設計に要請される種々の要件をテンプレート内に含めることで、フロント・ローディングの多くが吸収でき、設計品質も大幅に向上しました。

次章以降で、このビルドタイム / ランタイム・アプローチを中心に、新しいナレッジ・ベース設計のメソッドロジーについて論述します。

2. 製品設計作業とそのプロセス分析

2.1. 製品設計作業とは

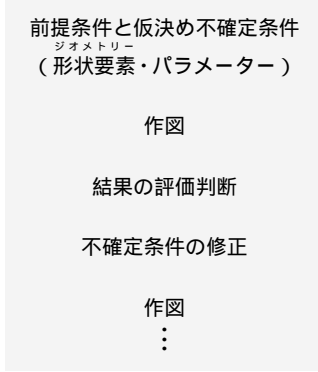
設計とは、対象プロダクトに対するさまざまな要求・制約、すなわち製品コンセプト(製品仕様・デザインなど)や、法律・規制、市場要求(品質・性能・価格など)、生産要件(工法上の条件、設備上の制約など)を満足させ、必要とされる数々の機能を

を成立させる作業です。

それぞれの要求や機能は、お互いに相反する条件となることが少なくありません。つまり製品設計とは、プロダクトに求められる数々の機能の優先順位を考慮し、そのバランスを取って成立させる作業ということもできるでしょう。

2.2. 設計プロセスの分析

従来の設計は、設計者の経験などに基づいて図面やCADモデルを取りあえず描き、その成立性やバランスを評価・判断して、必要に応じて何度でも描き直していました。トライ&エラーを繰り返しながら最適解を見つける方法が、一般的な作業の流れだったのです。



こうした一連の作業は一つの機能を成立させるために不可欠ですが、前述したように、一般に一つの対象プロダクトには複数の機能が存在します。しかも、その中の幾つかの機能はお互いに何らかの関連を持って成立します。

例えばネジ一つにしても、次のような複数の要求に対する機能が存在しています。

- ネジ頭 工具からの締め付け力を受ける機能
- ネジ頭フランジ部 締結する対象を保持する機能
- ネジ山 締め付け力を締結力に変える機能

またネジ頭のフランジ径は、ネジの呼び径や締め付け工具との関連から決定されます。

このように、すべての機能に対してそれぞれの関連性を考えながらトライ&エラーを行う必要があり、複雑なプロダクトでは、同時に膨大な検討を繰り返さねばなりません。結果的に、初期段階における設計の精度は極めて低いものとなります。もちろん検討を繰り返すことで、最終段階に近づくほど精度は急激に上がっていきませんが、その段階に至るまでの長い期間にわたり設計工数を減らすことはできません。極めて非効率なプロセスといえるでしょう。

今回の取り組みでは、こうしたプロセスを分解・分析し、論理的な形に組み立て直して、設計プロセスの効率化を目指しました。

(1) 従来のプロセスの問題点

- アウトプットを評価してインプットを修正していること
 - 不確定要素の仮決めをインプットにしていること
- この2点によりトライ&エラーの繰り返しに陥ってしまい、非効率な作業となっています。

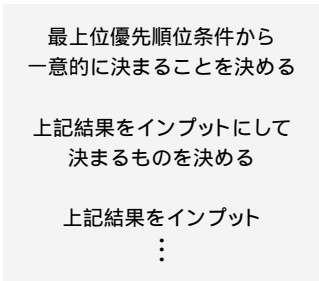


(2) 設計プロセスのあるべき姿
 左に示した作業の流れを積み上げていけば、同一作業の繰り返しがなくなり、効率的な設計プロセスとなります。

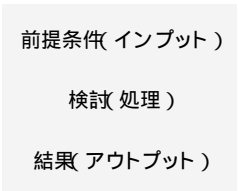
(3) あるべき姿を実現するには
 ・各検討(処理)におけるアウトプットは、必ず成立したものであること

ある機能を成立させる設計ロジックは、評価過程の中に必ず存在するはずで、ロジックを処理そのものに組み込めば、機能の成立が必ず導かれることとなります。その結果、処理の評価は不要となり、条件の修正に立ち戻る作業は最少化されます。

- ・インプットが必ず確定条件であること
- 設計対象に対する条件を漏れなく羅列し、優先順位を付けて右に示した決定連鎖を確立します。



これは、一つひとつの決定作業を単位化し、それをインプットとアウトプットに結び付けることで実現できます。“決定作業”



とは、ある機能を成立させることであり、これを“機能単位”と定義します。機能単位は左に示したプロセスで構成されます。

こうしたプロセスを経ることで、設計プロセスの形態はストレート・フォワードとなり、設計戦略が同じである限り、機能単位内の最適化ループはあっても、設計プロセス全体のループはほとんど起こりません。設計作業は効率化し、また設計の初期段階から品質が高くなります。

さらにこのプロセスは、アプリケーションとしてシステム化が可能であり、後述するCADの機能を活用して設計の自動化を図ることができます。

3. アソシアティブ・モデリング

設計プロセスの効率化の背景として、CADの機能向上があ

ることを1章で論述しました。

具体的には、CADの持つモデリング表現能力がパラメトリック関連機能を強化しており、パラメーターの変更により、モデルが表現している形状(ジオメトリー)を自律的に変更できることを意味します。

パラメーターとしては以下のものがあります。

- ・形状を定義するための数値
- ・形状そのもの
- ・判断の基準値など

また単独の形状モデルだけではなく、アセンブリーされたモデルの集合の中で、パラメーターの参照関係を持つことが可能です。あるパラメーターの変更により、関連する部分はその影響を受けて自動的に変更されるという動きが次々に伝播し、最終的に新しい形状を生成することが可能となっています。

筆者のチームでは、この機能を積極的に利用したモデリング手法を“アソシアティブ・モデリング”と呼んでいます。

最新のCATIA/V5を例に取ると、以下の機能がアソシアティブ・モデリングに利用できます(図1参照)。

- ・パブリッシング機能
- ・パワー・コピー機能

3.1. パブリッシング機能

CATIA/V5では、モデリングの単位を次のように呼んでいます。

- ・CATPart
 ファイリング単位の単体モデル。
- ・CATProduct
 CATPartの集合をアセンブリー状態で扱うための単位。
 CATProductは、CATPartだけではなくCATProductもその配下に持つことができます。

パブリッシング機能は、CATProduct下に定義されたCATPart間でジオメトリーを参照する機能です。

CATProduct下のあるCATPartで、他CATPartに参照させたジオメトリーをパブリッシュしておく、そのジオメトリーは、同一CATProduct下にあるすべてのCATPartから外部参照要素として利用できます。

ほかのCATPartのジオメトリーを参照できるという点ではCATIA/V5のリンク付きコピー機能と同じですが、外部参照は、パブリッシュされている要素をCATProduct下の階層パスとジオメトリーの同一インスタンス名称同士でリンクします。

これにより同一の参照関係を持つCATPartであれば、CATPartの置換を行ってもリンク関係が保持されるというメリットが生じ

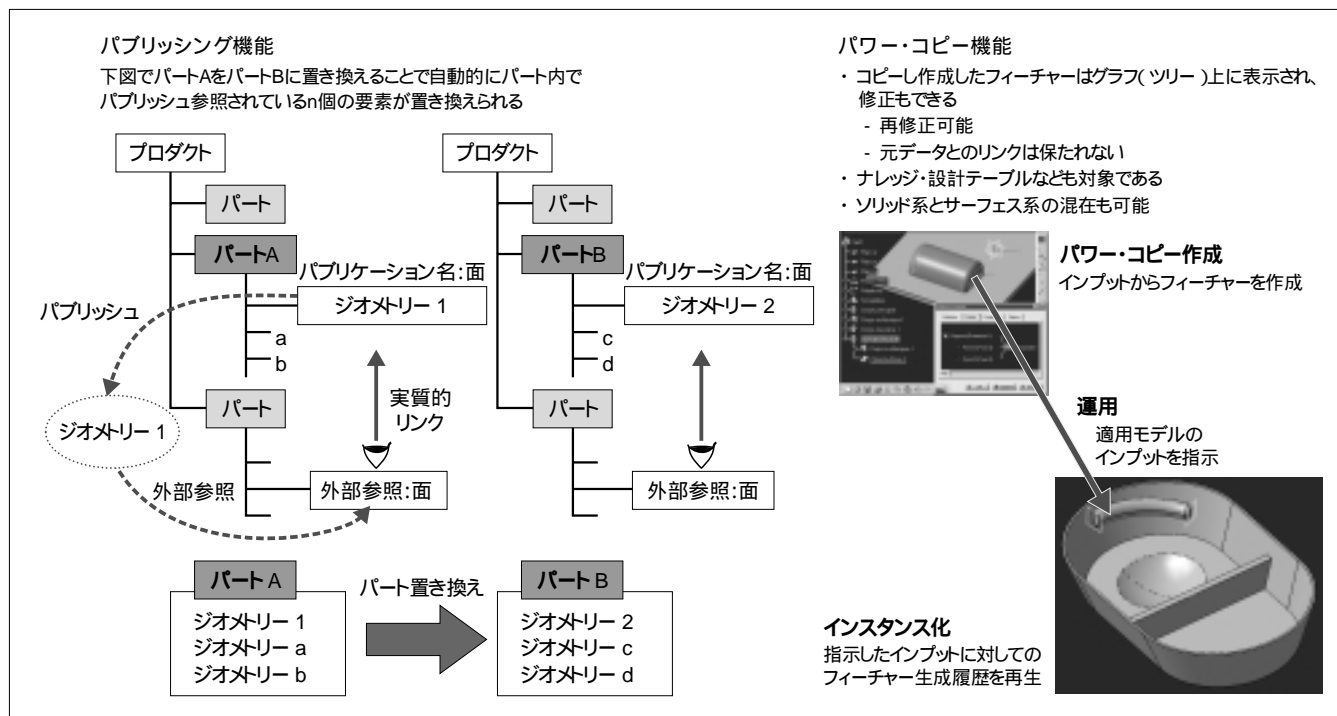


図1. パブリッシング機能とパワー・コピー機能

ます。CATPartの置換を前提にしたアソシアティブ・モデリングにおいては有効なリンク機能といえるでしょう。

3.2. パワー・コピー機能

あらかじめインプットが定義されたフィーチャー(形状履歴・文字・式・拘束)を参照しているCATPartのインプット要素に対してインスタンス化する機能です。インスタンス化された後では、元のパワー・コピーとのリンク関係を持たないため、適用されたCATPart内で個別に編集(変更)できます。

アソシアティブ・モデリングにおいては、プロセスの末端(他パートへのアウトプットなし)の形状で、複数個配置されるようなフィーチャーを生成する場合などに活用できます。

3.3. アソシアティブ・モデリングの特徴

CADモデルのアセンブリであるCATProductを使用することを3.1節で論述しました。従来の手法におけるCADモデルのアセンブリは、使用される各モデル(CATPart)が物理的な部品単位となります。実際の部品のアセンブリまたはサブアセンブリなどを表現する場合に使われますが、アソシアティブ・モデリングにおける各モデルは、基本的には設計の検討(処理)単位となり、この点が従来のモデリング手法と大きく異なります。

使用されるモデル・アセンブリは、設計対象の物理構成から成るものではなく、設計対象の機能単位の論理構成、つまり機能オブジェクト構成を表します。従って物理構成を表す場合

は、さらに上位でのアセンブリが必要となります。

4. ナレッジ機能の活用

モデリングにおける形状定義の途中では、それまでの結果を基に設計ノウハウを駆使したり、種々の制約をクリアしながら複雑な設計解を求める場合が少なくありません。

こうした要求に対応するために、CATIA/V5には数々のナレッジ関連機能が用意されています。

- ・ ナレッジ・アドバイザー2、ナレッジ・エキスパート2

設計ノウハウや製造上の制約をあらかじめ条件化し、モデリング履歴に組み込んでおくことで、より高度な自動設計が実現します。例えば、ある結果によってそれ以降のプロセスが分岐するときの判断や自動処理に活用できます。

- ・ プロダクト・エンジニアリング・オプティマイザ2

前述したように、設計とは「ある制約の中で最適な解を見つけること」です。設計における制約は、性能・寸法・重量・法規・加工/製造方法などの多岐にわたります。その解が一意に求められる場合は単純なパラメトリックとして定義できますが、求められない場合は、プロダクト・エンジニアリング・オプティマイザ2を活用して、パラメーターをある範囲内で変更しながら計算を繰り返すことで、最適解を見つけることが可能です。

5. ビルドタイムのメソロジー

2章において、設計プロセスを機能単位に分解することで次のような流れに整理でき、プロセス全体は機能単位のインプット/アウトプットによるリンク関係で記述可能であると論述しました。



つまり、設計プロセスはあたかもプログラムのように記述でき、前章で述べたCAD機能を活用すれば自動設計をシステム化できると考えられます。

2章における分析結果のプロセスを、実際の製品設計活動にそのまま適用することはもちろん可能ですが、類似製品の設計が多い場合には、対象製品の設計を分析して設計テンプレートを作っておき、さまざまなバリエーションの設計に効率良く利用することができます。

製品設計のノウハウを凝縮しテンプレート化することを「ビルドタイム」、そのテンプレートを利用して具体的な実際の製品の設計を行うことを「ランタイム」と呼ぶことにします。

ここでは2章における設計プロセス分析結果を基に、設計テンプレート作成に適用する場合のワークフローについて論述します。

5.1. 準備

対象製品の設計をテンプレート化するための準備として、以下の項目があります。

- 織り込むべき要件の収集・整理

どれだけの要件を織り込めるかがテンプレートの有用性を左右します。特に製造要件など下流工程の要件をどれだけフロント・ローディングできるかが、コスト/手戻り削減のキー・ポイントとなります。

- テンプレートの最終的なアウトプットの適用レベルの決定
「どの程度、詳細にするのか」プロセスのどの段階で利用

可能か」などを決定します。例えば、お互いに影響を及ぼす多数の部品の配置検討のテンプレートであれば、アウトプットは部品配置の決定のみであり、それを基に別テンプレートまたは手作業で本体側を設計することになります。

- 新たに必要となる付帯作業の有無のチェック

使用する標準部品のモデリングなど、新たに必要となる付帯作業の有無をチェックします。必要に応じてそのスケジューリングも行います。

5.2. 要件定義

要件定義フェーズにおいては以下の作業を行います。

- 設計単位の決定
- 適用シナリオの決定
- 機能単位分解
- 設計プロセス・シートの作成

5.2.1. 設計単位の決定

設計対象が大きい場合に、一人の設計者で扱える単位に分割します。通常は、部品・サブアセンブリ単位・部分単位(部単位)に分割します。

対象が単純な場合は、設計対象がそのまま一つの設計単位となります。

(1) 適用シナリオの決定

ランタイムにおける設計活動を想定し、以下の項目をまとめます。

- どのようなレベルまでの自動化を実現したいのか
- どのような使い方をするのか

(2) 機能単位分解

- 設計単位を機能単位に分解します(機能単位の定義は2章を参照)。
- 分解した機能単位に固有の名称を付けます。

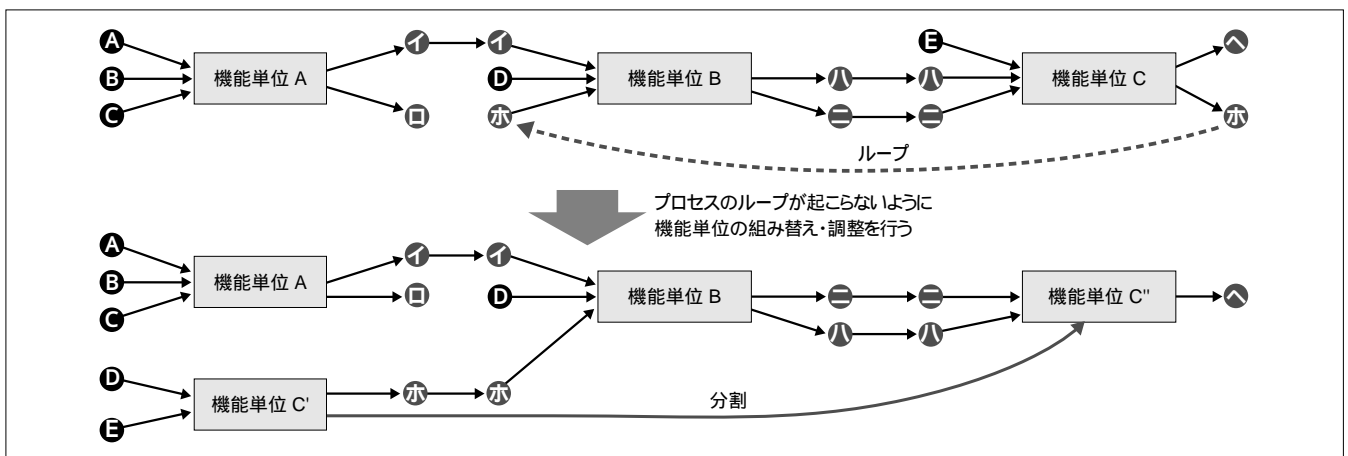


図2. 設計プロセス検討例

(3) 設計プロセス・シートの作成

- 機能単位の決定レベルで、現状の設計手順を整理します。
- ループや途切れのないように再構築します(図2参照)。
- プロセス上の不整合(ループや途切れ)がなくなるまで以下の作業を繰り返します。
 - ① 要求の優先順位順に設計単位を並べます。
 - ② 各設計単位に必要なインプットとなるジオメトリ、パラメータを書き出します。
 - ③ 書き出したインプットがアウトプットされる機能単位を特定します。
 - ④ アウトプットする機能単位 インプットが必要な機能単位の順に並べます。
 - ⑤ プロセス中にループがないか、必要なインプットで得られないものはないか、確認します。
 - ⑥ ループがある場合、機能単位の分割・統合・部分移動などで回避します。
 - ⑦ 必要なインプットがない場合、それをアウトプットできる処理を考えて該当機能単位に含めるか、もしくは新しく機能単位を定義します。あるいは既存の機能単位に含めます。
 - ⑧ 互いに依存関係にある複数の機能単位はオブティマイザーを利用した一つの機能単位としてまとめます。

5.3. 設計仕様書の作成

実際にアウトプットを生み出すのは、機能単位内の設計ロジックです。ここでは機能単位下で最終目的出力となるジオメトリ・フィーチャー(形状・位置・数値などの決定事項)が、インプットからどのようなプロセスで出来上がるかを整理します。

設計仕様書は以下の要素から成り、これはITシステム開発における外部仕様書に相当します。

- 機能単位関係シート
- 設計手順書
- IPO(Input Process Output)シート
- パラメーター・シート

(1) 機能単位関係シート

機能単位関係シートによって機能単位間の関係を整理します。

- 機能単位間の関係をマトリックス(機能単位関係シート)に書き出します。
- 機能単位関係シートでは、それぞれのインプットがどの機能単位で決定されるのか、矢印でドライバー、ドリブンを表現して各機能単位の依存関係を明確化します。

(2) 設計手順書

設計手順書の内容としては、各機能単位がどのように決め

られるのかを書き出します。アウトプットは決定ロジックそのものとなります。決定ロジックはしかるべき形状になるような論理型であり、後からのチェックは不要とするべきです。つまり、取りあえず作った形状や、あるいは過去の製品の形状を基に断面などを切つて間隙・関係をチェックする試行錯誤型であってはなりません。

設計手順からIPOシート(後述)につなげるために、各機能単位の成立の流れが分かるよう設計手順をまとめます。

以下の項目が要点となります。

- 設計プロセス・シートと同様に、条件を優先順位順に積み上げます。
- 条件の積み上げで成立範囲を絞り込みます。
- 最終的に一意の解が求められない場合は、最適化計算などのナレッジ・ツール活用を検討します。
 - ジオメトリ・フィーチャー間のバランスで成立するような事柄(非手続的プロセス)については、目的・可動条件・拘束条件を明示します。
 - 判断・分岐・繰り返しなどが必要な場合は、その条件を明示します。

(3) IPOシート

IPOシートで機能単位を構成するフィーチャーの関係(設計ロジック)を整理します。フィーチャーを生成するために必要なインプットや決定ロジックの流れは、フィーチャー間を結ぶ矢印で模式的に記述します(図3)。

(4) パラメーター・シート

インプットとなる要素と、設計でキーとなる要素、およびパラメーターに固有名称を付け、その説明を行います。各パラメーターに対してはデフォルト値、上/下限値を設定します。

5.4. モデリング仕様書

モデリング仕様書は以下の要素から成り、これはITシステム開発における内部仕様書に相当します。

- モデル構造図
- モデル・フロー図
- モデル手順書
- ナレッジ仕様書(必要な場合のみ)

(1) 設計単位]モデル構造図

モデルは検討単位(機能単位)のモデルのアセンブリーとなることを3.3節で論述しました。通常の設計対象では複数の機能単位が存在するため、モデルは必ず構造を持つことになります。そこで、まずモデル構造を決定する必要があります。

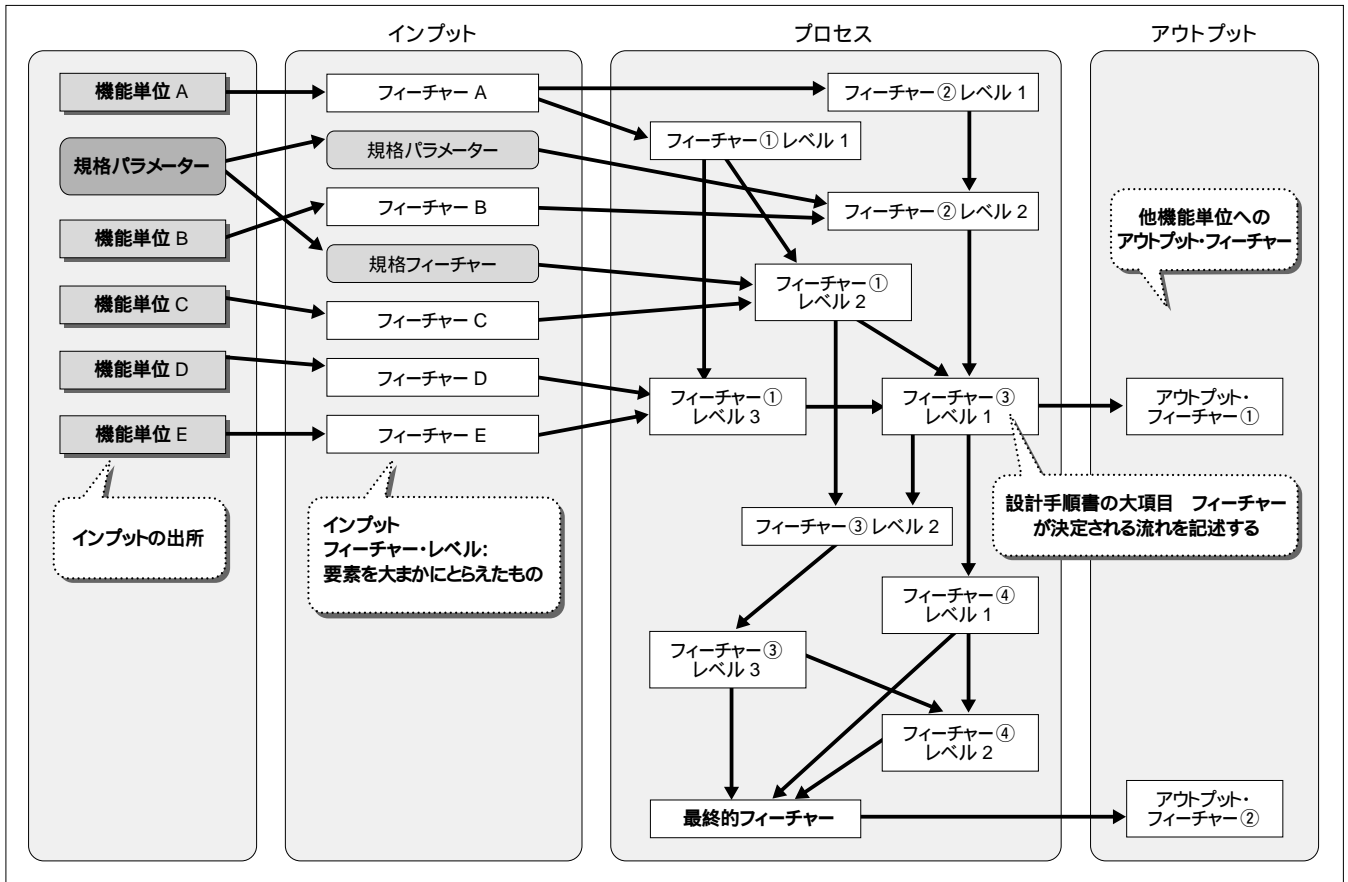


図3. IPOシートの例

• 設計単位と機能単位の関係の記述

具体的には、トップのCATProductとサブCATProduct、またはCATPartの関係、つまり階層化されたモデル構造およびモデル間のリンク構造を記述します。

• 設計単位に対するインプットの定義

インプットはトップのCATProductの直下(最上位レベル)で定義するのが一般的です。

• 設計単位下で複数の機能単位が使う共用パラメーター / ジオメトリーの定義

共用パラメーター(グローバル・パラメーター)は下位のCATPartにより参照されるため、インプットと同様にトップのCATProductの直下で定義するのが一般的です。

また次項で述べるサブのCATProduct / CATPartの置換を考慮し、かつ複数の同一ジオメトリー / パラメーターが存在しないように考えます。

• 機能単位の置換を考慮したモデル構造(図4)

設計単位の中で使われている機能単位が簡単に置き換え可能であれば、設計単位のシリーズ製品全体へ横展開できるようになります。

例えば、普及版の製品では単なる「ふた」が、高級版ではそこにアクセサリ機能が追加されているといった場合です。こ

こでは「ふた」の機能単位をアクセサリ機能単位(場合によっては構造を持つサブCATProduct)に置き換えることが考えられます。従って横展開の可能性がある場合には、機能単位の置き換えに対応できるように考えます。

(2) 機能単位 [モデル・フロー図]

モデル・フロー図を使用して、機能単位内のフィーチャー定義の流れをまとめます。ナレッジ機能を埋め込む場合は、フロー内に明記します。

(3) 機能単位 [モデル手順書]

モデリングの手順を書き下します。使用するCAD(本論文での想定はCATIA/V5)の機能に準拠して作業します。

(4) ナレッジ仕様書

ナレッジ機能を使用する場合は、必要に応じてルールや、オプティマイズ、ジェネレーティブ・ナレッジなどの適用を決め、必要な要件を整理して仕様をまとめます。

5.5. モデリング

この段階になると、モデル手順書などを参照しながらCADを操作する比較的単純な作業となります。企業内でのモデリング標準に準拠して作業します。

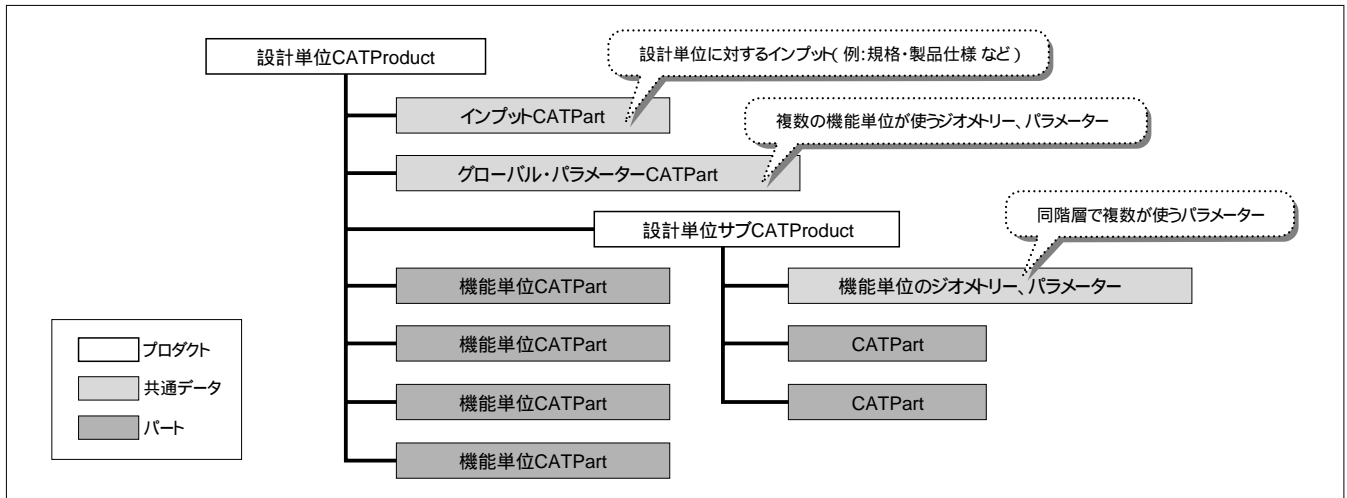


図4. 設計単位における全体モデル構造例

5.6. テスト・保管

作成されたCADモデルが仕様通りに機能し、例外ケースを排除できるかどうかを、各種パラメーターを与えてテストします。テストにパスしたモデルは正式に保管します。保管時の配慮事項としては、ネーミング・ルールの徹底、保管ディレクトリーなど保管場所のくくりといった物理的な管理と、改定履歴・機密保持・使用権限などの一般的な知的資産管理の運用が必要となります。

6. ランタイムにおける設計活動

ランタイムでは、実際の製品設計にビルドタイムで作られたテンプレートを活用します。対象がビルドタイムで想定したものとまったく同一であればそのまま利用できますが、現実には若干の相違があることでしょう。

ランタイムでは以下の作業を行ってから対象の設計作業に入ります。

- 利用するテンプレートの収集。
- テンプレートの組み合わせのチェック。必要に応じてテンプレートの調整・修正。

適用シナリオについては既に触れましたが、テンプレートのアウトプット・レベル設定によっては以下に挙げるようにさまざまなケースがあるでしょう。

- 詳細まで一気に完成するケース。
- 検討段階までを行い、詳細は人手で設計するケース。
- 詳細さが増すに従って使用するテンプレートのレベルが分けられるケース。

いずれにせよ、このステージにおけるテンプレート利用時の設計者の役割は、インプット・パラメーターを正しく選択し、後は設計の監視と、機能単位の最適化のモニタリングとなります。

設計の結元(寸法・性能・外形形状など)のすべてがパラメーター化されたテンプレートを使うことができれば、設計変更は瞬時であり、最新の市場ニーズ/トレンドに追従することも容易です。短時間のうちに多くの代替案を検討でき、非常に大きなメリットを享受できます。

なお、詳細を人手で行うケースであっても、その作業で編集履歴付きの機能を利用していけば、2回目以降のパラメーター変更では、編集履歴のアップデートにより自動的に最終形状を生成できます。詳細レベルまでテンプレート化するとあまりにも複雑になる場合は、ある程度アウトプット・レベルにとどめて全体の効率を考えるとよいでしょう。

7. 従来の設計プロセスへのインパクト

ビルドタイム/ランタイム・アプローチのメソッドロジーは、設計作業を厳密な解析することにより、設計行為そのものと「設計のナレッジ化」が直接結び付いています。設計のナレッジ化に当たっては、ともすれば「従来の設計プロセスを変えずに、どのように推進すべきか」という議論になりがちですが、本メソッドロジーの採用によって新しいパラダイムへ脱皮できることでしょう。

設計者は、繰り返し設計により最適解を求めることを常識として疑っていません。

実は、かなり複雑なある設計ケースにおいて、「ベテランの設計者のだれもがここから設計を開始する。それが常識」としていた手順が誤りであることが検出されたことがありました。設計プロセス解析による流れ図からループが起こるため、最適化を行ったところ、別の観点から設計を始めれば大きなループを防げることが発見されたのです。このことから分かるように「経験 = ナレッジ」ではありません。長らく行っていた無駄な作業

を、設計プロセスの解析によって検出できたということです。

またビルドタイムとランタイムに分けたことで、各フェーズにおける作業分担に新しい役割を考えることができます。

ビルドタイムにおいては、要件定義から設計仕様書作成までのプロセスはITシステム設計と同様の作業となりますが、設計者にとってはまったく初めての経験です。つまり、今後の設計プロセス・コンサルティングには、ITのバックグラウンドを持つ技術者に活躍の場があると考えられます。

またモデリング作業はモデル手順書など参照しながらCADを操作するという比較的単純な作業であり、モデル作成作業そのものを設計者が直接行う必然性はもはやなくなります。モデラーと呼ばれる専門家でも可能な作業であり、アウトソーシングの対象と考えることもできるでしょう。

同様にランタイムにおいては、設計のノウハウのほとんどがテンプレートに含まれることから、作業そのものを本来の設計者が行わず、設計者の監督下でアウトソーシングしてしまうことも可能です。

今後、設計者の役割は、設計ナレッジ・エンジニアとしてビルドタイムに大きくシフトしていくと考えられます。

8. 今後の展開

筆者は、本手法こそ「本来あるべき手法である」という確信を持っています。本手法は、メソドロジーとして既に確たる実績を積んでいます。今後のさらなる展開として以下の2点へのチャレンジが必要だと考えています。

- 文書作成の効率化
- テンプレートの管理(データベース化)

(1) 文書作成の効率化

本手法ではさまざまな仕様書類を文書化しています。現時点ではPowerPointやExcelなどを使用していますが、文書内容に重複があるといった課題が存在します。より効率的な表記方法が必要でしょう。

また、全体の効率化のために「設計の記述に適した新しいCADツールはどうあるべきか」という研究も欠かせません。設計のナレッジ化を推進するための機能としての側面も考慮する必要があります。

(2) テンプレートの管理(データベース化)

“部品表”は製造業における企業活動の中核データベースです。ただし、生製品の部品構成を表現しているものの、あくまでも製品・部品という“物”としての対象ビューからの管理であって、

設計単位や機能単位といったビューでの管理ではありません。従って、設計のテンプレートや、あるいは設計の初期段階の状態を効果的に管理するには、対象製品の設計の論理的構成に着目した管理データベースが必須となります。

さらにこのデータベースは、設計のプロセスを経て具体的な形状が創生された段階で、既存の部品表と連携を取る必要があります。この分野においては、設計プロセスをサポートするかなり大規模なアプリケーション開発が必要と考えられます。

9. おわりに

本手法は、IT開発メソドロジーに詳しくない設計技術者にとっては非常に新鮮に映るようです。

ところが実際には、IBM社内でよく知られているシステム開発手法ADSG(Application Development Standardization Guide: アプリケーション開発標準化ガイド)そのものです。その意味では、本手法はADSGのソフトウェア開発以外への適用ということもできるでしょう。CADがプログラミング言語に相当し、図形(ジオメトリ)定義言語と考えられるということです。

物としての実体がないソフトウェア開発では、内容の妥当性を保証・チェックするために開発/ドキュメンテーション手法が非常に重要であり、そのためにメソドロジーが開発されてきました。しかし従来からの機構設計などでは、図面というコミュニケーション手段があり、試作などで実体を作ることも可能なため、厳密な設計プロセスの解析はなおざりにされてきた面もあると思われます。

ある目的を果たす機能の定義手法に着目すれば、その対象がソフトウェア/ハードウェアのいずれであろうとも本質的な違いはまったくありません。ソフトウェア開発のためのメソドロジーが、そのまま従来の設計プロセスに適用でき、その過程そのものが設計ノウハウのナレッジ化と直接結び付くという大きなメリットも持ちます。

また設計プロセス解析のコンサルティングとして、IT技術者が活躍できることも確認できました。

本手法は、家庭用品のような単純なものから複雑かつ巨大なものまで、その対象を選びません。すべての設計対象に対して適用可能です。設計の初期段階から、必要な機能・性能・製造要件などのフロント・ローディングが行えるためQCDの要求にマッチした手法であり、今後、急速に普及すると考えられます。

(ページ数および表記上の観点から、著者の了解を得て編集部にて手を入れてあります)