

Smarter Engineering

— より効率的な「モノづくり」を目指して —



本稿では、製造業における「モノづくり」が現在直面している問題の中から、「製品の複雑さ」と「開発プロセスの複雑さ」に焦点を当て、それを解決するためのアプローチとしての Smarter Engineering という IBM リサーチにおける研究イニシアチブについて述べます。ここで述べる Smarter Engineering は、「モデル共有」と「シミュレーション」をベースに、「クラウド・コンピューティング」を情報インフラとして、「モデル駆動開発」を推進するというコンセプトです。開発プロセスにおいて、上流から下流まで（詳細度は異なるにしても）共通のモデルを基に作業を進めるための「モデル共有」、そのモデルを用いてソフトウェアを含むシステムの検証および最適化を行うための強力な「シミュレーション」、それらを効率的に行うためのセキュリティーが正しく管理された「エンジニアリング・クラウド」などについて述べます。

Article 2

Smarter Engineering

- Toward more efficient products development -

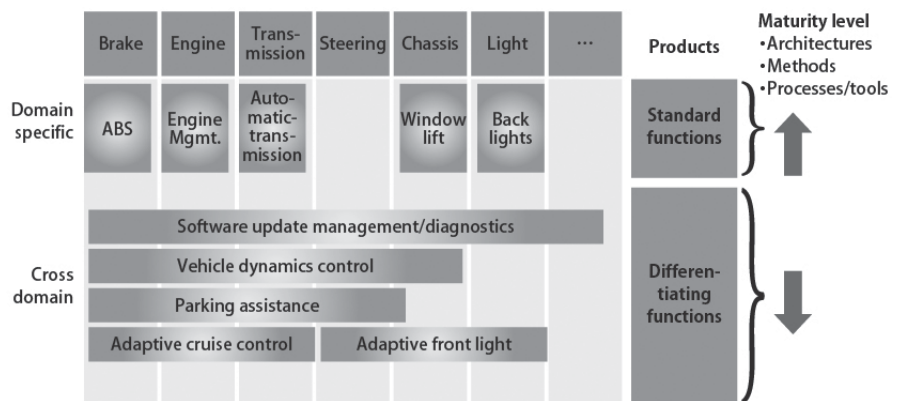
IBM Research has launched its Smarter Engineering initiative, with the objective of providing an approach to solving two major current issues in the manufacturing industry: the complexity of products, and the complexity of the development process. This article describes some important aspects of Smarter Engineering in the context of Model Driven Development based on (1) model sharing across each phase in the development process, (2) simulations which can be used for the validation and optimization of systems, including both software and hardware, and (3) the “engineering cloud,” an information infrastructure offering high-level security.

① はじめに

製造業が現在直面している課題には、「増大する開発コスト」「開発期間の短縮」「製品の複雑化」「開発プロセスの非効率性」「開発要員の不足」「企業の枠を越えた協業」「環境問題への対応」などがあります。当然ながら、これらは相互に深く関係しています。経済産業省の統計 [1] によれば、製造業における開発コストは年率 10% 増加しており、その中でもソフトウェアの開発コストに関しては年率 14% もの伸びを示しています [1]。この原因の 1 つは製品そのものおよび開発プロセスの複雑化であり、また環境対応も無視できない原因の 1 つです。本稿では「製品の複雑化」「開発プロセスの複雑化」に焦点を当て、それらを解決するための技術の方向性について

て論じます。

「製品の複雑さ」をより深刻にしている理由の 1 つは、ソフトウェア（コントローラー）による機能が増えてきていることです。例えば自動車の場合、高級車で使われているソフトウェアの総量は、1,000 万行に達しようとしています。「ソフトウェアによる機能」によって機械系だけでは不可能であったような高度な制御が可能になり、その



Source: IBM Institute for Business Value.

図 1. 複雑化する制御システム

結果、高性能かつ安全で環境に優しい製品を作ることが可能になりました。また複数のコントローラーを用いた統合制御も一般化してきています（図1）。ところが、ソフトウェアは、それが正常に機能しているかどうかの検証が機械系に比べて困難です。さらに機械系に比べると、小変更がより大きな影響を及ぼす可能性があります。また、複雑な制御を行うものに関してはテスト・ケースの作成自体が困難な作業となります。

現在、製品開発コストにおけるソフトウェア開発コストの割合は42%に達しており[1]、市販後に起こる問題の46%がソフトウェア起因のものであります。また、昨今多くの製品がネットワーク接続されつつあることも無視できません。この場合、いわゆる「製品」の開発のほかにも、ネットワーク側で提供される「サービス」の開発も行わなければなりません。

開発プロセスに関しては、いわゆるV字チャート（図2）の各フェーズでの作業の効率化ということも、もちろん重要ですが、フェーズ間の情報あるいは知識の受け渡しがより重要になると考えられます。要求分析、基本設計、詳細設計、実装、検証とプロセスが進むときに上位のフェーズで作られた情報が下位のフェーズに正しく伝わらないと、例えば検証フェーズでは本来何を検証しなければならないかがあいまいになりますし、設計変更を余儀なくされたときに本来の要求に沿っているかどうかを確かめるだけでも大変な作業となります。このようなことは製品が複雑でない場合は大きな問題とはなりません、複雑で開発に多くの人々がかかっているような場合は問題が顕在化します。上で述べたような製品と開発プロセスの複雑化への取り組みとして、IBMはSmarter Engineeringという研究イニシアチブを始めました。2章では、その構想について論じます。

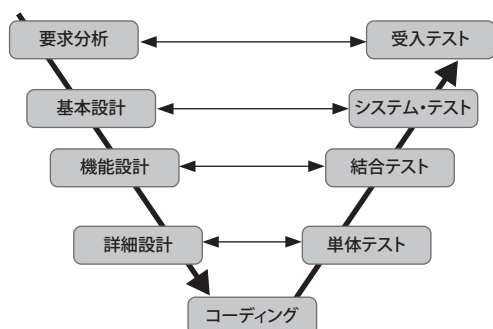


図2 V字チャート

2 Smarter Engineering 構想

1章でSmarter Engineeringが相対する課題を述べましたが、より具体的には、「製品と開発プロセスが複雑化している現況においてITは何を解決できるか」ということです。本稿では主に製品に組み込まれるソフトウェア開発を対象として話を進めますが、ハードウェア開発にも大いに関係します。

われわれは、「モデル共有」と「シミュレーション」を中心に置いた「モデル駆動開発」をベースに、その課題に取り組みます。図3はその概念図です。開発プロセスの上流から下流のどのフェーズからでもアクセスできるモデル・データベースとそのモデルを使用して検証や最適化を行うための強力なシミュレーターを備え、最適化、検証を繰り返しながら各開発フェーズを行うというコンセプトです。情報インフラとしてはクラウド・コンピューティングを想定しています。

2.1 モデル駆動開発

本稿での「モデル駆動開発」は、プログラム開発の前にモデルを作成し、そのモデル上で機能の検証を行い、プログラムそのものはモデルから自動生成する、というものです。実際に「もの」を作る前での機能検証が可能なので「手戻り」が低コストでできること、モデルが「動く仕様書」として機能するので、開発者同士のコミュニケーションが円滑に行われ得ること、モデルの情報を元に、よりクリティカルな部分（厳密な検証が必要な部分）にフォーカスしたテスト・ケースを生成できることなどが利点として挙げられます。また、同じ機能を実現するのに複数

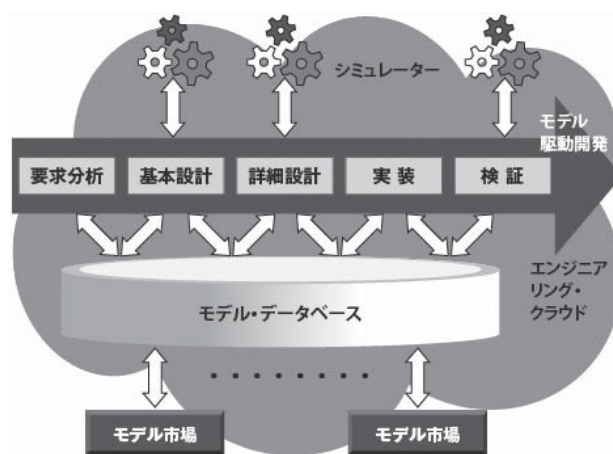


図3 Smarter Engineering の概念図

の選択肢がある場合に、まずモデルで優劣を検討し候補を絞り込むことが可能です。

モデル駆動開発において、プログラムの自動生成は重要です。もしモデルからプログラムの自動生成ができなければ、従来のプログラム開発とは別にモデル開発を行うということになり、コストの増大を招きます。

いわゆる IT におけるプログラム開発の場合は、それが稼働するコンピューターの細かい仕様を気にすることはありません。ハードウェアの仕様はデファクトまたは標準でほぼ決まっており、また、多くの機能が仮想化されているからです。その場合モデル化とは、例えば UML によるプログラムの仕様の記述をするということになります。一方、製品に組み込まれるソフトウェア（組み込みソフトウェア）の場合は、それが制御する対象、例えば機械系のモデル化も並行して行う必要があります。また、組み込みソフトウェアが稼働するハードウェア（車の ECU など）も必ずしも標準が決まっているというわけではないので、ECU（Electronic Control Unit）のモデルが別に必要となります。また、組み込みソフトウェアは、タイマーなどの周辺機器を有効利用しているケースが多いので、ECU のモデル化による周辺機器のモデル化は重要です。

IBM リサーチでは、

- 要求分析に自然言語処理／テキスト・マイニングの技術を適用するための研究
- スプレッド・シートを用いたモデル作成ツールの研究
- モデリングを正しくかつ効率よく行うための研究などを行っています。

2.2 モデルの共有

例えば半導体の設計においては VHDL などの言語が用いられますが、そうした言語を使って書かれた仕様（モデル）は製品開発、検証にも用いられますし、実際の製品を量産するときにも用いられます。このように、可能ならばモデルは V 字プロセスの設計以降の各フェーズにおいて上流フェーズで作成したモデルを再利用できることが望ましいのですが、組み込み開発の現状はそうになっていません。また、モデルと要求分析が連携していないケースも多いと思われます。

以上をかんがみ、われわれの考える Smarter Engineering では、図 3 のように、開発の上流フェーズから下流フェーズで統一的なモデルが使用できることを目標とし、そのために全フェーズからアクセスできる共通のモ

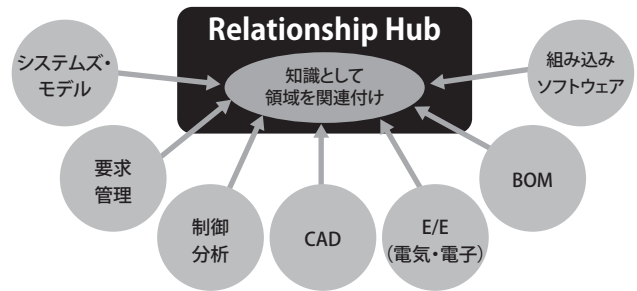


図 4. モデル・データベースの概念

デル・データベースを備えます。モデルはエンジニアが使用しているツールに大きく依存し、すでにそのようなツールを用いてデータベース化されている場合が多いので、このモデル・データベースは、データ・ウェアハウスの形式ではなくて、図 4 のようにデータ・フェデレーションの形をとります。

モデルの供給元としてモデルそのものが流通するモデル市場の構築を目指します。モデル市場を作るためには、ある程度の標準化が必要で、逆に言うとその標準化ができれば、大いに開発コストの削減に貢献します。ただし、モデルを使用する側の各企業は差別化のための余地を残したいはずなので、標準化をどのレベルまで行うかは検討すべき大きな課題の 1 つです。

IBM リサーチでは、

- 散在するモデル間の関係を記述することで統一したモデル・データベースとしての機能を提供するための研究（Relationship Hub）などを行っています。

2.3 シミュレーション

モデルを使つての機能の検証は、モデル駆動開発の最も重要な点の 1 つですが、それには静的なもの動的なものがあります。図 3 でも示しているように動的な検証を行うツールとしては、シミュレーションがあります。

例えば、自動車などの制御システムの開発で使われるものとしては、制御アルゴリズムの検証に使われる MILS（Model-In-the-Loop Simulation）、特定の ECU を前提としたときの制御機能の検証に使われる SILS（Software-In-the-Loop Simulation）、実際の組み込みソフトウェアをエミュレーター上で実行し、そのソフトウェアの実行時の正しさ（割り込みタイミングなどの検証を含む）の検証を仮想プロセッサ上で行う VPILS（Virtual Processor-In-the-Loop Simulation）、実際の ECU を用いて検証を行う HILS（Hardware-In-the-

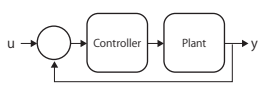
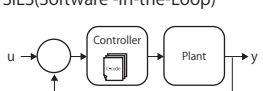
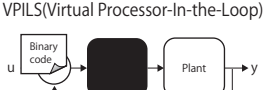

 <p>MILS(Model-In-the-Loop)</p>	<p>プラント・モデルも コントローラーも SimulinkやStateFlow で記述</p>	<p>制御アルゴリズム の検証</p>
 <p>SILS(Software-In-the-Loop)</p>	<p>プラント・モデルは Simulinkで記述 コントローラーは C言語で記述</p>	<p>C言語で記述された コントローラーの 機能の検証</p>
 <p>VPILS(Virtual Processor-In-the-Loop)</p>	<p>プラント・モデルは Simulinkで記述 コントローラーは実際 のバイナリー・コード をエミュレーション</p>	<p>ECUの中の本物の コードを検証</p>
 <p>HILSHILS(Hardware-In-the-Loop)</p>	<p>プラント・モデルは Simulinkで記述 コントローラーは 本物のECUを使用</p>	<p>複数のECUハード ウェアの協制御の 検証</p>

図5. 制御システム検証用のシミュレーション

Loop Simulation) があります (図5)。それぞれ制御をつかさどるフィードバック制御系とそれが制御する対象である機械系のモデル (プラント・モデルと呼ばれます) を記述します。ソフトウェア検証という目的では、プラント・モデルは通常、常微分方程式で表される範囲で、Mathworks 社の Simulink [2] などを用いてモデル化

されますが、話をソフトウェア開発に限らなければ、設計のときに物理系のシミュレーション、例えば、機構シミュレーション (Multi-body Dynamics)、流体シミュレーション (Computational Fluid Dynamics)、構造シミュレーション、ノイズや振動のシミュレーションなどを用います。物理系のシミュレーション、特に機構シミュレーションは、制御系と深く関連するので可能ならば制御系との統合シミュレーションが望ましいところです。

IBM リサーチでは、

- ソフトウェア検証のための高速 VPILS
 - MILS/SILS の高速化
 - モデルからのテスト・ケース/プログラムの自動生成
 - テスト・ケースのカバーする範囲を推定する研究
 - さまざまな物理系シミュレーションのための HPC (ハイ・パフォーマンス・コンピューティング) の研究
- などを行っています。

2.4 エンジニアリング・クラウド

モデル共有とシミュレーションをベースに、モデル駆動開発を推進するというのが Smarter Engineering の中心と

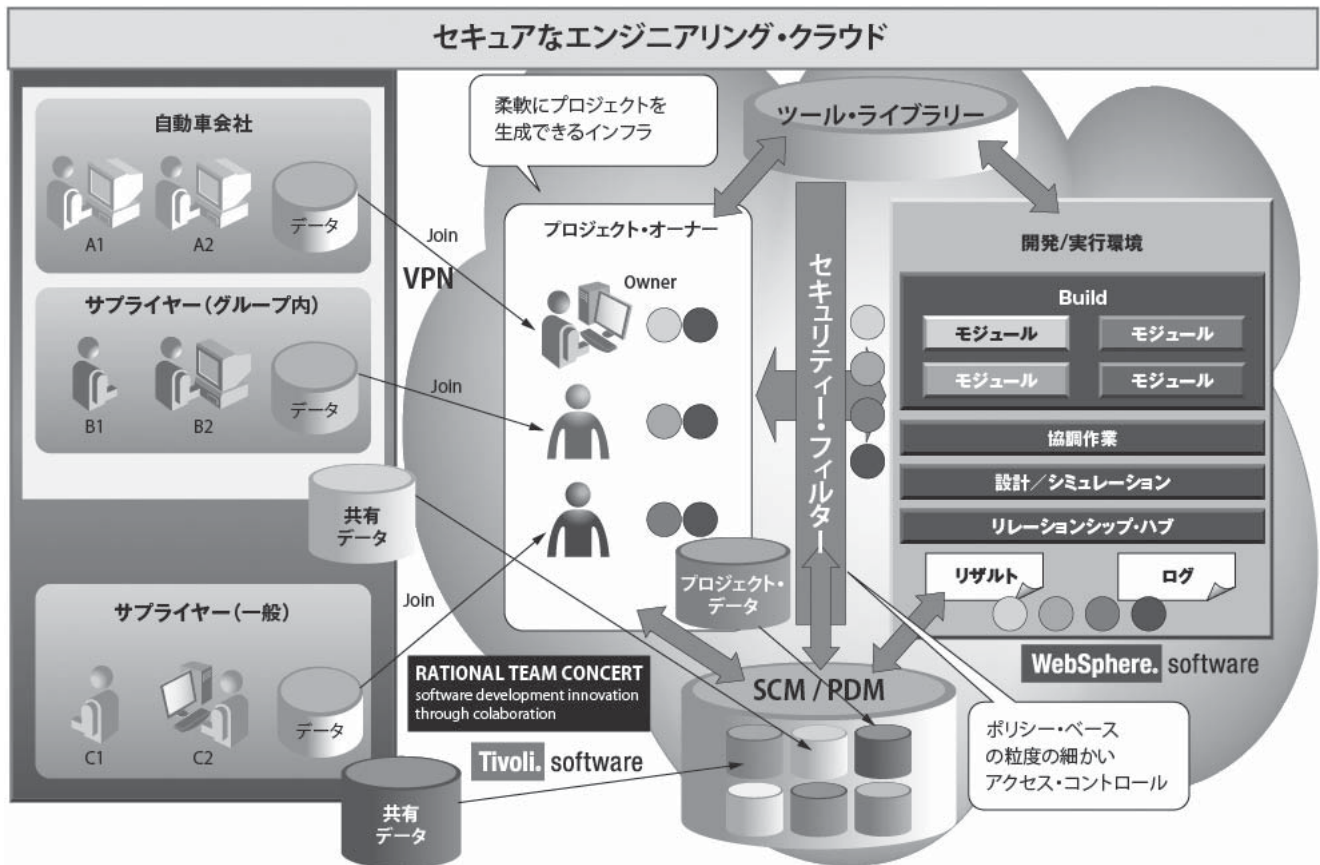


図6. エンジニアリング・クラウド

なるコンセプトですが、それを行う情報インフラとしてクラウド・コンピューティングのプラットフォームを考えています。

工業製品の多くは、単一の企業だけでは開発および製造ができないので、企業間のコラボレーションが必須になります。柔軟なプロジェクト管理機能を持つエンジニアリング・クラウドができれば、企業間のコラボレーションがより効率的になることが期待できます。

ただし、ある種のモデルは、その企業の「知財」であるので、「モデル共有」は、「正しくアクセス制御されたモデル共有」でなければなりません。例えば、モデルへのアクセス・レベルとしては、「モデルの中身（ソースコード）が見える」「モデルを実行（シミュレーション）することだけができる」などが考えられます。各製品開発プロジェクト単位で、そこに参加する企業や人が異なる可能性があるため、モデル共有を行うためのアクセス・コントロールは細かく制御された（fine-grained）ものでなければなりません。逆にこのような制御ができると、企業 A と企業 B との間で、モデルの中身を開示しないシミュレーションによって相互接続の検証を行うといったことも可能になります。

シミュレーションは一般に大きな計算資源を必要としますが、計算需要はプロジェクトの開発フェーズによって大きく変化します。エンジニアリング・クラウドはそれを吸収する役割も提供します。

③ まとめ

以上、IBM リサーチの考える Smarter Engineering 構想について述べてきましたが、この構想には技術的にも政治的／経済的にも幾つかの課題があります。

技術的な課題としては、

- モデルの精度の保証
- シミュレーションの精度と速度の両立
- 複数のシミュレーターを組み合わせた統合シミュレーションの実現

などがあります。ただし、これらは時間はかかるかもしれませんが、次第に解決していく問題だと考えます。

政治的／経済的な課題としては、なんとんでも、

- モデルの（一部の）標準化
- モデル市場の確立

です。

例えば ECU のモデル化を行うときに、現状は個別にモデルを作成しているわけですが、周辺機器（ネットワー

クやタイマーなど）だけでも標準があれば、ECU のモデル化にかかるコストと時間は大幅に軽減されます。また、モデルからソフトウェアの自動生成もより容易になります。さらに、特定のツールに依存しないモデルの標準があれば、モデルそのものが流通するモデル市場を構築することにより、毎回新しくモデルを作る必要がなくなります。

上で挙げた政治的／経済的な課題は、各企業が何を自社の競争力と見え、何をそれ以外と考えるかに関係します。例えば、ECU の場合、組み込みソフトウェアを自社の競争力と考えるなら、ECU のハードウェアはコモディティーでも構わないかもしれないので、その場合はモデルの標準化というアイデアに賛同して下さるでしょうし、エレクトロニクスを自社の競争力と考えるならば、モデルは競争力の源となるので、標準化には簡単には賛同いたさないでしょう。

ただし、1 章でも書いたように、製品の開発コストは年々上昇しており、それに対応するための手段としてモデル駆動開発を取り入れるならば、モデルの標準化は避けて通れません。現在モデル駆動開発を取り入れる場合の最大の障害がモデルの開発だからです。

われわれは、「モデル駆動開発」を支える技術の開発を行い、それを使うためのインフラを整え、さらにモデルの標準化のような活動を続けます。それが、われわれの考える Smarter Engineering です。

[参考文献]

- [1] 経済産業省2008年版組込みソフトウェア産業実態調査報告書の公表について、http://www.meti.go.jp/policy/mono_info_service/joho/2008software_research.html
- [2] Simulink製品紹介、<http://www.mathworks.co.jp/products/simulink/index.html>



日本アイ・ビー・エム株式会社
東京基礎研究所
先進組み込みシステム担当

梶谷 浩一 Kouichi Kajitani

[プロフィール]

1985 年日本 IBM 入社。東京基礎研究所でデータベース、ビデオ会議システムなどの研究に従事。1996 年同研究所企画担当、2000 年同研究所エクスプロラトリー・システム担当、2001 年公共事業ライフサイエンス事業推進部担当、2008 年東京基礎研究所システム・シミュレーション担当、2010 年同研究所先進組み込みシステム担当。