



家庭から宇宙まで、エコチェンジ



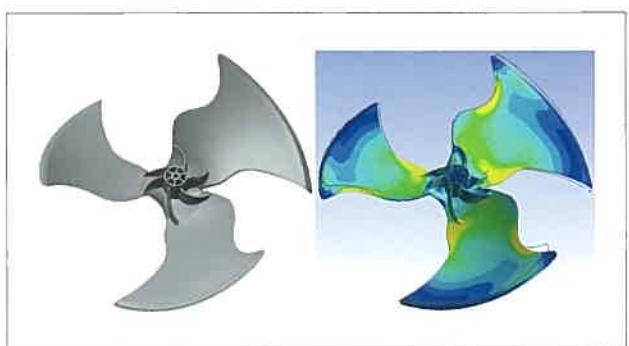
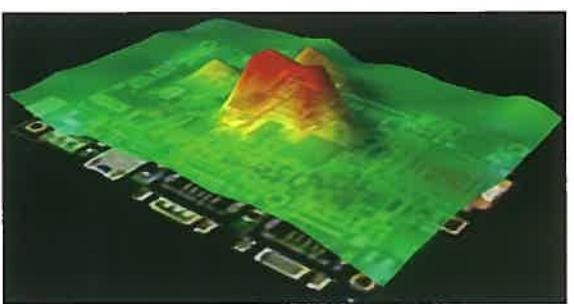
# 三菱電機技報

12

2016

Vol.90 No.12

## 成長戦略を支える設計・検証技術



## 目 次

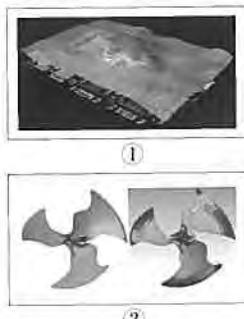
特集「成長戦略を支える設計・検証技術」	
情報システムと人の協調で進歩する設計技術.....	卷頭言 1 瀧 寛和
成長戦略を支える設計・検証技術.....	卷頭論文 2 竹野祥瑞・山中康弘
ドメイン固有言語を利用した設計手法.....	7 佐藤美和・岡 藍子
プロセス改善技術者育成コースの設計.....	11 久野倫義・中島 錠・芝田 見・近藤聖久・小笠原公一
静的解析技術によるソースコードの効率的な品質確保.....	15 藤本卓也・中村勝彦・浅川忠隆
コードレビューハン法の改善によるコード品質確保の効率化.....	19 井元 煙・細谷泰夫
高位LSI設計検証技術とその応用 .....	23 峯岸孝行・山本 亮・宮野昇晃士・元濱 努・坂手寛治
ADC回路におけるアナログ-デジタル回路協調設計技術.....	27 大東睦夫
仮想実行環境の構築容易化技術と応用.....	31 奥田勝己・竹山治彦・三輪昌伸・伊藤貢二・町田宏隆
幅広い分野の製品を支えるEMC技術 .....	35 佐々木雄一・大橋英征・岡 尚人・宮崎千春
空調用冷凍サイクル設計技術の高度化.....	39 羽下誠司・山下浩司・鶴村 優・玉木章吾・竹中直史
製造ばらつきを考慮した機構解析技術.....	43 幸本宏治・舛田真一・中田光昭・渡辺和昌・三木伸介
3D測量データを活用した現地据付工事改善 .....	47 安達裕輔・安木俊之・桑名善太郎・石川 秀
グローバル事業拡大に対応した梱包設計技術.....	51 潮 敬之・森 健晴・瀧村文裕・山中泰弘・山田泰寛

Design and Verification Technologies to Support Growth Strategy	
Design Technology Improvement by Human Computer Collaboration	
Hirokazu Taki	
Design and Verification Technologies to Support Growth Strategy	
Shouzui Takeno, Yasuhiro Yamanaka	
Model-based Approach to the Design of Domain Specific Language	
Miwa Satou, Aiko Oka	
Design of Training Course for Software Process Improvement Engineers	
Noriyoshi Kuno, Tsuyoshi Nakajima, Akira Shibata, Kiyohisa Kondo, Koichi Ogasawara	
Efficient Quality Assurance of Source Code by Static Analysis Technologies	
Takanari Fujimoto, Katsuhiko Nakamura, Tadataka Asakawa	
Efficiency of Code Quality Assurance by Improving a Method of Code Review	
Kaoru Imoto, Yasuo Hosotani	
High Level LSI Design and Verification Technology and its Application Examples	
Noriyuki Minegishi, Ryo Yamamoto, Koji Miyahama, Tsutomu Motohama, Hiroharu Sakate	
Collaborative Design Technology for Analog and Digital Circuits in ADC	
Mutsuo Daito	
Automated Generation of Virtual Execution Environments and its Applications	
Katsuji Okuda, Haruhiko Takeyama, Masanobu Miwa, Shinji Ito, Hirotaka Machida	
EMC Technologies for Supporting Wide Range of Products	
Yuichi Sasaki, Hideyuki Ohashi, Naoto Oka, Chiharu Miyazaki	
Advancement of Refrigeration Cycle Design for Air Conditioning	
Seiji Haga, Koji Yamashita, Takeshi Hatomura, Shogo Tamaki, Naofumi Takenaka	
Method for Analyzing Multibody Motion Considering Production Tolerance	
Koji Sachimoto, Shinichi Masuda, Mitsuaki Nakata, Kazumasa Watanabe, Shinsuke Miki	
Improvement of Installation Work by Use of 3D Laser Scanner	
Yusuke Adachi, Toshiyuki Yasuki, Zentaro Kuwana, Shu Ishikawa	
Packaging Design for Global Business Expansion	
Takayuki Ushio, Takeharu Mori, Fumihiro Takimura, Yasuhiro Tanaka, Yasuhiro Yamada	

## 特許と新案

「プリント配線板のはんだ接合検査用配線構造」	
「ハーネス検査装置」.....	55
「空調装置および空調制御プログラム」.....	56

三菱電機技報90巻総目次 .....	57
--------------------	----



表紙：成長戦略を支える設計・検証技術

三菱電機では、高機能・大規模システム化する製品開発プロセスやグローバル化する開発環境の変化に対応し、もう一段高いレベルの成長を実現するため、設計・検証技術の更なる高度化に取り組んでいる。

表紙は、設計・検証技術の中から、(1)電子回路基板におけるEMC(Electromagnetic Compatibility)の可視化測定図、(2)プロペラファンの応力解析モデルを示す。

## 巻/頭/言

## 情報システムと人の協調で進歩する設計技術

Design Technology Improvement by Human Computer Collaboration

瀧 寛和  
Hirokazu Taki

情報システムの設計生産への活用は、新たな段階に来ている。三菱電機が提案する“e-F@ctory”に掲げる“人・機械・ITの協調”的進展は世界の潮流であり、ドイツのインダストリー4.0、アメリカのインダストリアルインターネット、中国の中国製造2025など、ものづくりの効率と品質向上を更に発展させる技術革新が進んでいる。日本も内閣府の政策としてスタートした第5次科学技術基本計画で、産業だけでなく、一般社会の情報化の進展によるサイバー空間(情報社会、仮想空間、インターネットでの結合)とフィジカル空間(物理空間、人間や生物)の融合によるスマート社会(Society5.0)を構築することを進めている。

これらに共通する技術として、人工知能、ビッグデータ処理やIoT(Internet of Things)などを駆使することで、社会システムが人へのサービス能力を向上させるスマート社会の実現を進めている。また、製造現場では、製造装置の知能化やインターネットによる装置間の連携によって、情報や知識の伝達を通じて生産システムの効率を飛躍的に高めることを狙っている。一方、製造物の機能・コスト・品質は、製造物“作るもの”が大きな要素であり、生産設備だけの改革では、製品競争力には限界がある。コモディティ製品の製造では、どの製造業も同じものをいかに安く作るかの競争に陥ることになり、同じ装置で製造する製造業が利益を出しにくい状況になっている。製造物の製品競争力は、設計技術にかかっていると言っても過言ではない。

人工知能やIoTなどの新しい情報技術を設計・検証技術によってうまく活用することで、“作るもの”そのものの革命が進んでいる。IoTを活用し、生産現場での製造物の部品や製造装置の状態、出荷後の利用環境での様々な項目のデータを収集し、これを製品設計にフィードバックすることで製品の利用環境への適合性がより増している。

人々、現在の製品は、コンピュータが埋め込まれた製造物が多く、設計時に考慮しなければならない設計諸元や検証項目も膨大であり、ハードウェアとソフトウェアがともに複雑化している。物理的な原理・構造とソフトウェアが融合して、機能を実現しているものがほとんどである。検証でも、物理的な部分とソフトウェアの両方を、個別に、また、統合して調べる必要があり、設計・検証のプロセスも発展していく必要がある。このような中、以前から設計・検証支援には、CADとCAEが活用されてきたが、こ

の複雑化する設計対象には、より高度な情報処理技術が必要とされている。

設計対象の機能や物理現象のシミュレーションと利用環境の仮想現実化による検証も可能となってきた。複雑な電子回路や大規模なソフトウェアの挙動は、見える化の技術もさることながら、見えやすい設計も必要とされている。製品挙動のシミュレーション技術は、IoT時代では設計時だけでなく、出荷後の遠隔診断に応用される重要な技術と言える。また、IoTのメリットとは裏腹に、インターネット経由で装置がハッキングされる脅威もあり、あらゆる製品のセキュリティ対策も必要となってきた。情報家電などがハッキングされて迷惑メールの発信源となっている事例も増加している。

良い製品設計には、その製品の機能や物理現象をシミュレーションすることで、適切な構造や寸法のパラメータを求める支援ができる。特定の機能を実現する設計解は複数あり、適切な設計解の探索が重要な作業となっている。多くの設計制約を満たす設計解を探すのは時間のかかる作業であり、今後は、迅速な設計には人工知能をいかにうまく利用するかが、性能や品質の良い設計の鍵となる。多くの設計候補から性能の良い解を選び、良い解を組み合わせて、新しい解を提案していくものに遺伝的アルゴリズムがあり、良い解だけ生き残る方式が設計システムに広く応用されている。

人工知能で注目されているアルゴリズムにディープラーニング(Deep Learning)がある。これは、画像認識であれば、画像の要素を自動的に分類し、その要素の組立てで、画像を認識することを学習する。製品の利用環境適合性の認識であれば、特定の製品の諸元と利用環境の状態の不適合事例から、製品と利用環境の適合性を自動的に学習し、ある環境にその製品が利用できるかを事前に判断することなどに利用できる。同様に、設計仕様からの適した構造の選択、相性の良い部品の組合せ選択など設計作業で利用できる場面は沢山ある。人工知能は、設計分野での利用も更に進み、設計の効率化と品質向上のキーテクノロジーとして、使ったもの勝ちの世界が到来すると予想されている。

IoTや人工知能を活用した情報システムと人の協調は、確実に産業に大きな影響力を与えることになり、新しい情報処理技術をうまく設計・検証に融合させた企業がその設計力・製品力を大きく伸ばすことは間違いない。

# 卷頭論文



竹野祥瑞\*

山中康弘\*\*

## 成長戦略を支える設計・検証技術

*Design and Verification Technologies to Support Growth Strategy*

Shouzui Takeno, Yasuhiro Yamanaka

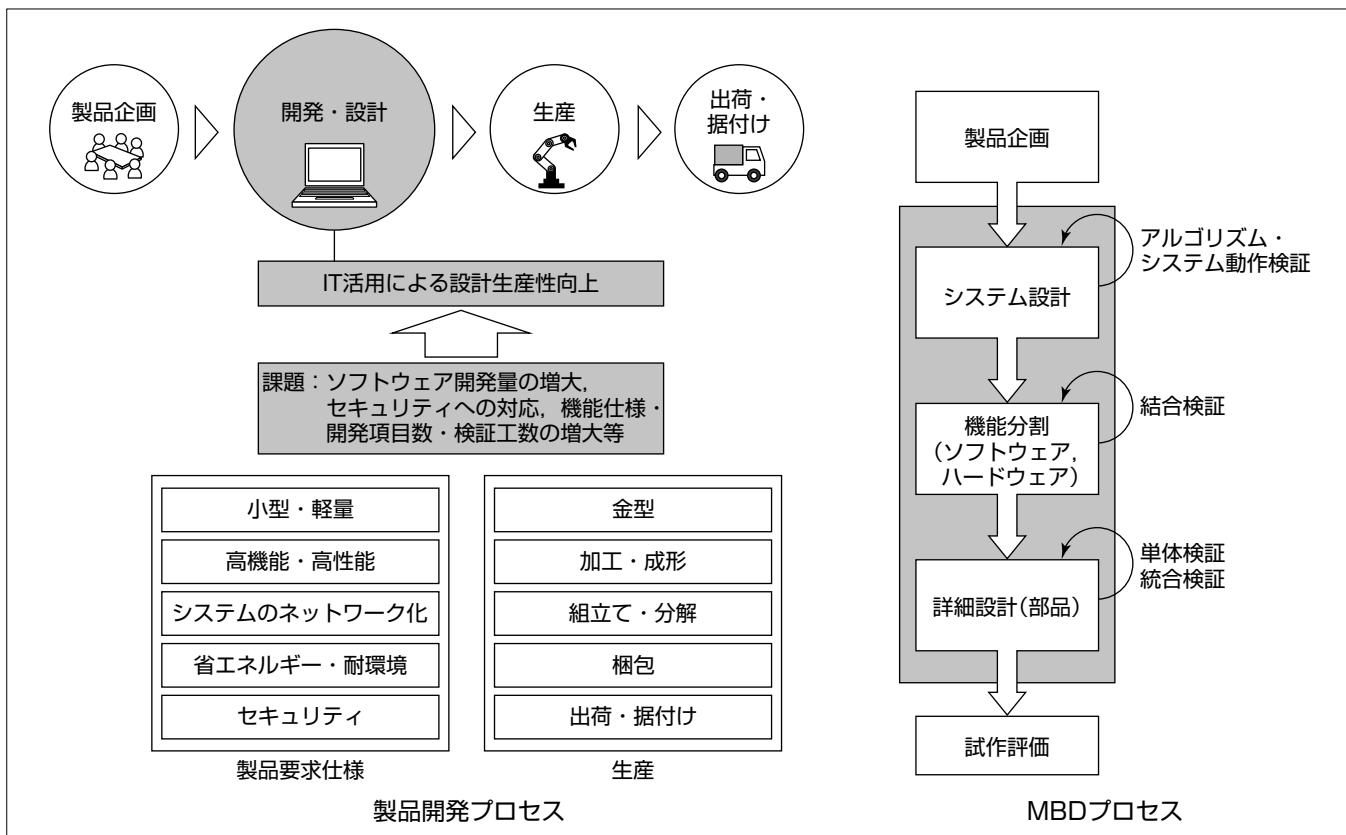
### 要旨

IoT(Internet of Things)時代の製品開発では、機器の高機能・高性能化に加え、システムのネットワーク化に伴うソフトウェア開発量の増大、セキュリティへの対応など開発課題が増大している。また、製品のグローバル展開に伴う使用環境、品質設計の多様化、国ごとの省エネルギー規制やリサイクル法に対応した改良型の派生機種数の増大、現地据付け・保守性、国際物流への対応など、設計者が考慮すべき製品の機能仕様や開発項目数、検証工数は増大の一途をたどっている。三菱電機では、高機能・大規模システム化する製品開発プロセスやグローバル化する開発環境の変化に対応し、もう一段高いレベルの成長を実現するため、ITの浸透による設計生産性向上を中心としたプロダ

クトとプロセスの改善活動に取り組んでいる。

特に、製品開発プロセスの中で上流に位置する“設計”では、製品モデルを用いて、システムレベルや部品レベル等の各階層でのモデルシミュレーションを重ね、素性の良い基本設計仕様を段階的に詳細化して精確に作り込んでいく“モデルベース開発(Model Based Development : MBD)プロセス”を導入して曖昧な部分や開発ボトルネックを早期に洗い出し、重要課題に開発リソースを集中させて、合理的に性能・品質の確保や作業効率化を実現している。

この特集号では、“ソフトウェア設計”“電子回路設計”“機械・構造設計”分野で、ITによる設計・検証シミュレーションを活用したモデルベース開発の適用事例を中心に述べる。



### 設計生産性向上の取組み

製品開発プロセスで、近年の開発・設計、生産への製品要求仕様からくる、ソフトウェア開発量の増大、セキュリティへの対応、機能仕様・開発項目・検証工数の増大等の課題に対し、MBDプロセスの適用等、ITを活用した設計・検証技術の高度化に取り組んでいる。

## 1. まえがき

IoT時代の製品開発では、機器の高機能・高性能化への要求に加え、システムのネットワーク化に伴うソフトウェア開発量の増大、セキュリティへの対応など開発課題が増大している。また、製品のグローバル展開に伴う使用環境、品質設計の多様化、国ごとの省エネルギー規制やリサイクル法へ対応した改良型の派生機種数の増大、現地据付け・保守性、国際物流への対応など、設計者が考慮すべき製品の機能仕様や開発項目数、検証工数は増大の一途をたどっている。さらに、近年急増する海外拠点への現地化プロセスでは、“日本人同士の“あうんの呼吸”による従来型のものづくりチームワーク”や“職人的な暗黙知技術”的展開は期待できない。国内マザー工場に蓄積された設計情報・技術資産を、より明示的な共通言語であるデジタル製品定義(DPD: Digital Product Definition)や機能モデルを媒介にして正確に現地へ伝達し、海外拠点と国内マザー工場間で協調開発の効率化を図る必要がある。

当社では、高機能・大規模システム化する製品開発プロセス、グローバル化する開発環境の変化に対応し、もう一段高いレベルの成長を実現するため、ITの浸透による設計生産性向上(Digital Transformation)を中心とした、プロダクトとプロセスの改善活動に総合的に取り組んでいる。

## 2. 設計・検証技術の高度化

製品開発プロセス(図1)の中で最上流に位置する“設計”は、価値の高い製品設計、適正な原価企画を源流で創出するための製造業活動でのコアコンピテンシーであり生命線と言える。特に近年は、電子・機械・ソフトウェア制御が融合した高性能・大規模システムが急増し、限られた期間内に膨大な数の性能仕様・品質基準を低コストで具現化するため、量産開発のウエイトを上流化させるフロントローディング手法<sup>(1)(2)(3)</sup>の更なる進化・深化が求められている。

そこで、自然言語の仕様書や図面をベースとした従来法に対し、図式表現や動作レベルが記述された製品モデルを用いた仮想検証環境上で、実機試作を極力行わず設計上流段階からシステムレベルや部品レベル等の各階層でモデルシミュレーションを重ね、素性の良い基本設計仕様を段階的に詳細化して精確に作り込んでいくMBDが有効となる。MBD手法ではモデル記述言語や物理モデルで表現された製品モデルを用い、設計上流段階から電子・機械・ソフトウェア制御系の仮想検証を繰り返して曖昧な部分や開発ボトルネックを早期に洗い出し、重要課題に開発リソースを集中させて、合理的に性能・品質の確保や作業効率化を狙う(図2)。ただし、ITベースのMBD手法を活用して大きな経営効果に結びつけるには、開発現場における人間系を中心とした設計エンジニアリング活動の本質を理解した上

で、効果的にデジタル手法を埋め込み、地道にプロセスを改善していく現場活動との高度な融合が不可欠である。

この特集号では、グローバル競争の中で多様化・大規模システム化する製品開発で、“ソフトウェア設計”“電子回路設計”“機械・構造設計”分野で、ITを活用したMBDの適用事例を中心に述べる。

### 2.1 ソフトウェア分野の設計・検証技術

ソフトウェア設計・検証技術の高度化に向けた主な取組みとして、仕様の記載漏れを抑制する図式表現による仕様の明確化とソースコードの自動生成、派生や複雑化を抑止するソフトウェアの統合・再生・改造技術の構築、海外顧客に対応した国際標準プロセスに準拠したプロセス運用について述べる。

#### (1) 図式表現による仕様の明確化とソースコードの自動生成

自然言語で記述した複雑な仕様が誤認識される問題、大規模化によってリソースが不足する問題に対して、図式表現を活用することで、ルールに基づいた記述による仕様の明確化、ソースコード自動生成による仕様展開の効率化に取り組んでいる。

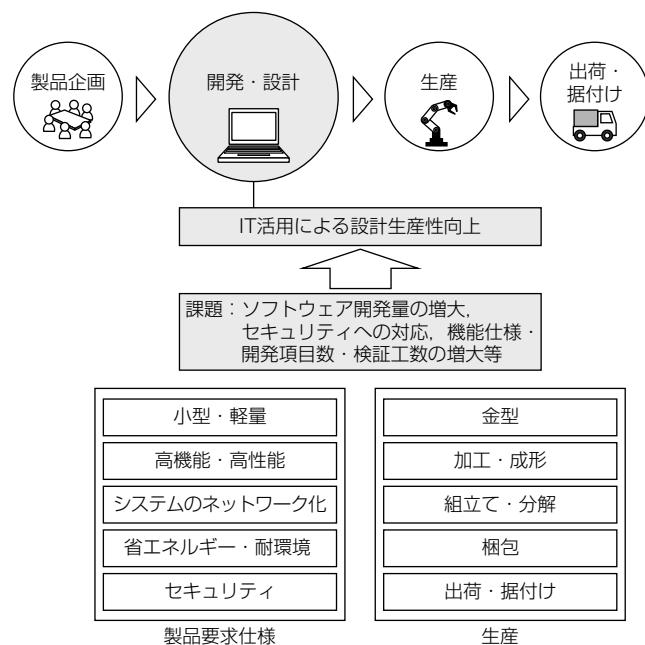


図1. 製品開発プロセス

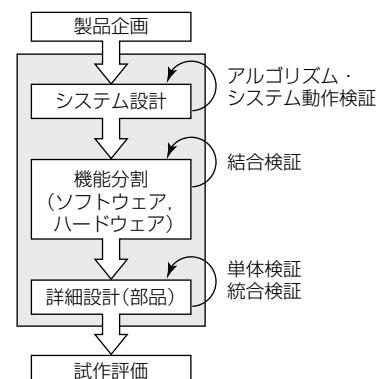


図2. MBDプロセス

# 卷頭論文

自然言語で記述した仕様は曖昧さが多く、仕様の複雑化によって、開発の際に仕様の記述漏れや仕様の誤認識が発生する。そこで、ソフトウェアの設計品質向上のため、仕様の記述ルールを定義した図式表現を用いて仕様の明確化を図っている。離散系ソフトウェアでは図式表現(Unified Modeling Language: UML<sup>(注1)</sup>)の記述を、制御系ソフトウェアではMATLAB/Simulink<sup>(注2)</sup>の記述を推進している。さらに、図式表現は、ソフトウェアの大規模化に伴う設計リソース不足解消にも効果があり、従来は仕様書を見ながら人手で作成していたソースコードを自動生成することで実装工数を大幅に削減できる。ただし、UMLの記述が複雑になると仕様の認識に時間がかかってしまうため、その対策としてUMLを補完する位置付けで、ドメイン固有言語(Domain Specific Language: DSL)で記述した仕様書からのソースコードの自動生成手法を構築した(図3)。この特集号の論文“ドメイン固有言語を利用した設計手法”(p.7)で、大規模通信システムの開発にDSLを適用した事例について述べる。

## (2) ソフトウェアの統合・再生・改造技術の構築

空調機器等では、システム化による既存機種間の接続、海外の向け先ごとの機種展開による流用開発が増加しており、繰り返し開発による派生や機能の複雑化による開発効率の低下が問題となっている。そこで、多機種に派生してしまったソフトウェアに対して再統合する技術の開発を取り組んでいる。派生したソフトウェアを構成する内部モ

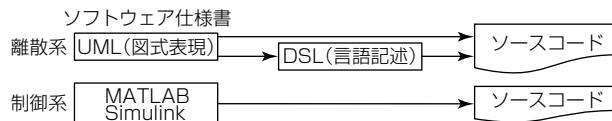


図3. 仕様書からソースコードの自動生成

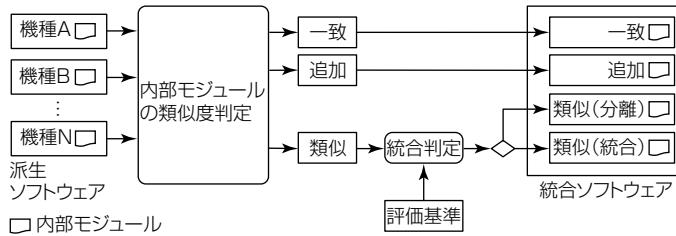


図4. ソフトウェアの統合技術

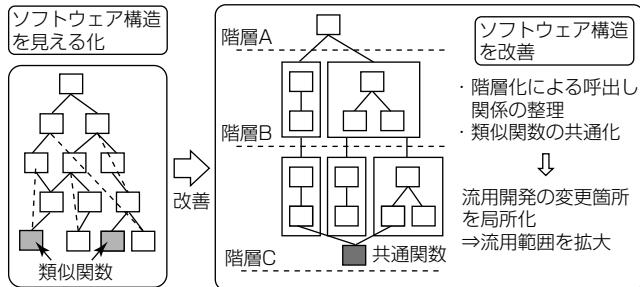


図5. ソフトウェアの再生技術

ジューの類似度を数値化して評価基準を基に定量的に統合判定し、再統合することで、派生したソフトウェアを1つに集約する(図4)。これによって、仕様の修正漏れの防止、修正箇所の集約による開発効率化を図った。

統合後は、複雑化したソフトウェア構造をシンプルな構造に見直すソフトウェア再生技術(図5)、シンプルなソフトウェア構造を維持するソフトウェア改造技術を適用する。

## (3) 国際標準プロセスに準拠したプロセス運用

海外展開に伴い、開発プロセスが一定水準以上で運用されていることを示すため、国際標準プロセスに準拠する必要性が増えている。

車載機器等では、海外顧客が要求するプロセス能力水準の保有を証明するため、国際標準プロセス(Automotive SPICE<sup>(注3)</sup>)に準拠したプロセス運用に取り組んでいる。ソフトウェア開発のPDCA(Plan Do Check Action)サイクルを回すための支援として、現状のプロセスを診断して問題点を抽出し、開発プロセス改善計画の立案、改善の実行支援、規則化等のプロセス開発の定着支援を実施している(図6)。また、プロセス改善の定着化のため、プロセス改善技術者育成にも取り組んでいる。この特集号の論文“プロセス改善技術者育成コースの設計”(p.11)で、現場で直面する問題を解決するプロセス改善技術者を育成する事例について述べる。

(注1) UMLは、Object Management Group Inc. の登録商標である。

(注2) MATLAB及びSimulinkは、The Math Works, Inc. の登録商標である。

(注3) Automotive SPICEは、Verband der Automobilindustrie e.V. の登録商標である。

## 2.2 電子回路分野の設計・検証技術

電子回路の設計・検証技術の高度化に向けた主な取組みとして、LSI開発での高位設計動作モデル活用による設計効率化、アナログ機能検証の高度化による設計品質向上、仮想システム検証環境の構築によるソフトウェアの上流検証について(図7)、電子機器開発で複雑化するEMC(Electromagnetic Compatibility)課題に対し、開発初期段階から設計品質を作り込むEMC設計フロー構築について述べる。

### (1) システムレベルの高位設計動作モデル活用

ソフトウェア、アナログ回路、デジタル回路の区別のないシステムレベルモデルから機能分割したデジタル回路のレジスタ転送レベルのHDL(Hardware Description Language)記述を自動生成するフローを構築し、さらに、回路削減手法等をルール化することでHDL記述の自動生成適用範囲を拡大した。この特集号の論文“高位LSI設計検証技術とその応用”(p.23)で、組み込みシステム設計への応用事例について述べる。

### (2) アナログ機能検証の高度化

車載向けを中心としたセンサ応用機器がキー製品となっており、タイムリーな差別化機能の製品搭載のためにアナ

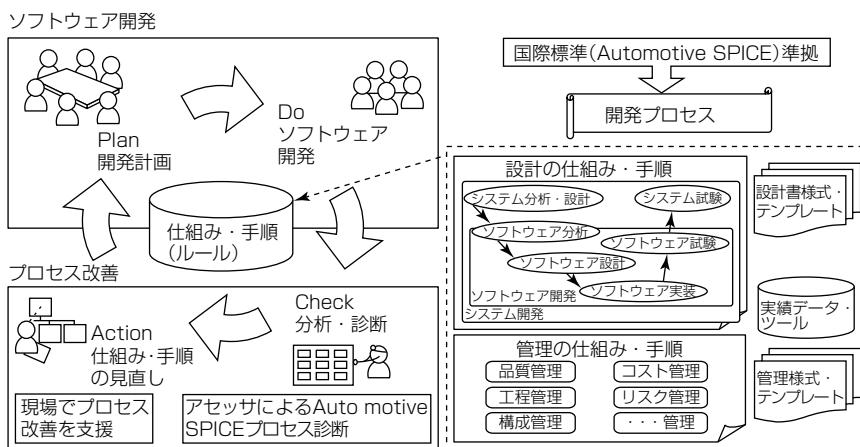


図6. ソフトウェア開発プロセスの改善手法

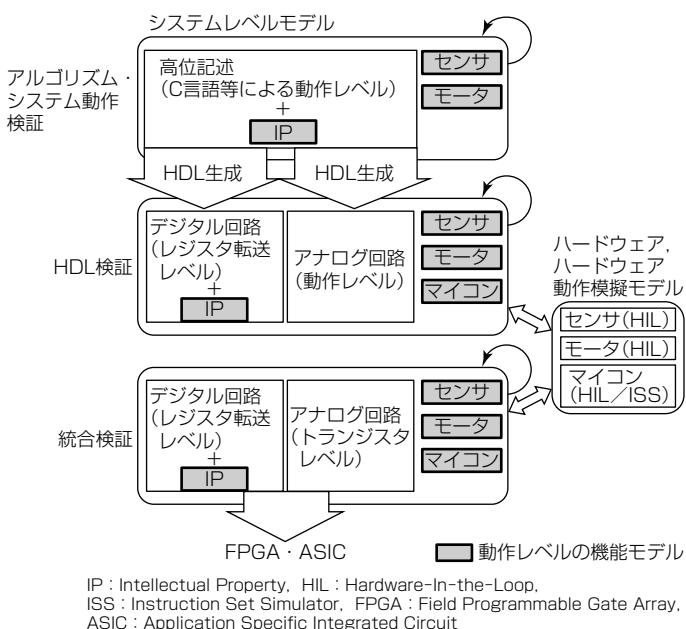


図7. MBDプロセスのLSI仮想システム検証環境

ログ設計上流での機能検証を実施することで設計手戻りの抑制を図っている。

アナログ機能検証で、検証対象をデジタル機能との協調設計に範囲を広げ、アナログとデジタルの機能分割から検証までを可能とした。さらに、検証モデルの改良、検証速度の向上等の改善に取り組んでいる。この特集号の論文“ADC回路におけるアナログ-デジタル回路協調設計技術”(p.27)で、アナログ回路とデジタル回路の統合検証の前段階のHDL検証でデジタル回路とアナログ回路の協調設計を行った事例について述べる。

### (3) 仮想システム検証環境の構築

従来は、ハードウェア検証後にソフトウェアとの組合せ検証を行っていたが、組合せ検証からの手戻りロスや機能分割の適正化が課題であった。これらの対策にISSやHILを利用した仮想システム検証環境の構築によってハードウェア設計と並行した設計上流でのソフトウェア検証や、ソフトウェアの機能単位の性能評価でボトルネック抽出

を可能とし、設計期間の短縮、設計手戻りの抑制を実現した。この特集号の論文“仮想実行環境の構築容易化技術と応用”(p.31)で、ISSを利用した仮想システム検証環境の適用事例について述べる。

### (4) EMC設計フロー構築

IoTの普及によるセンサや機器のネットワーク接続、電子機器の数十Gbps以上の伝送速度、信号の低電圧化に伴うEMCの課題が複雑化している。また、世界各国がEMC規制強化の方向にあり、グローバル展開のために各国のEMC規制への対応が必要な製品が増加している。

これらの要因に対し、設計段階でEMCリスクを抽出し、要素技術検討等を行うEMC設計フローを構築して製品設計品質の向上を図っている。この特集号の論文“幅広い分野の製品を支えるEMC技術”(p.35)で、開発初期段階から設計品質を作り込むEMC設計フロー、EMC要素技術等について述べる。

### 2.3 機械と構造の設計・検証技術

機械と構造の設計・検証技術の高度化に向けた主な取組みとして、空調用冷凍サイクル制御設計技術の構築による環境負荷低減、製造ばらつきを考慮した機構解析技術の構築による複雑化する製品構造の設計品質向上、設計の海外展開に向けた3Dデータの開発プロセスへの活用について述べる。

#### (1) 空調用冷凍サイクル制御設計技術の構築

冷熱・空調機器は、開発サイクルが短い中で、高い省エネルギー性能を達成する必要がある。従来、圧縮機や弁の動作制御を行う冷凍サイクル制御ソフトウェアの設計・デバッグを実機による運転検証試験で実施していたため、設計手戻りが発生していた。

今回、冷凍サイクル計算と制御ソフトウェアを統合して、パソコン上に仮想の運転検証試験の環境を構築して、設計上流段階で設計者による冷凍サイクル制御設計を可能とした。これによって、様々な外気温度・負荷条件の性能予測・冷媒制御設計を効率化し、設計・デバッグ期間を短縮した(図8)。この特集号の論文“空調用冷凍サイクル設計技術の高度化”(p.39)で、仮想の運転検証試験の環境を開発した事例について述べる。

#### (2) 製造ばらつきを考慮した機構解析技術の構築

従来、設計上流段階から設計者自身がCAEを活用して設計品質を検証することで設計手戻りを抑制し、対策時間と対策コストの低減を図ってきた。更なる設計品質の改善、設計効率化に向け、従来は解析の専門家が実施していた、製造ばらつきを考慮した機構動作部の設計品質

# 卷頭論文

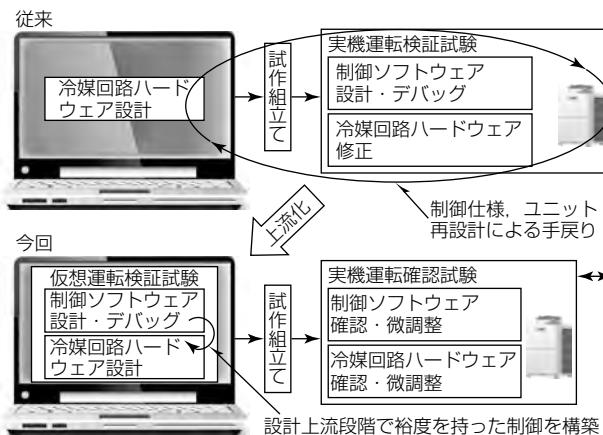


図8. 冷凍サイクル制御設計の流れ

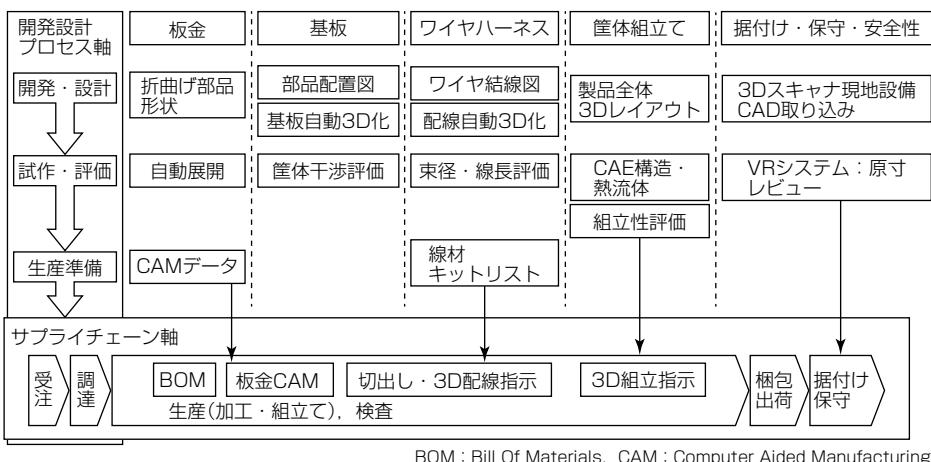


図9. 開発プロセス全体に3Dデータを活用する仕組み

確認を設計者が簡便に検証できる環境を構築した。

多数の部品を接続した機構部を持つ製品は、従来、簡略化した機構解析モデルを用い、影響の大きさと考えた代表パラメータを用いて動作特性を計算していた。そのため、製造ばらつきが製品機能に与える影響の定量化が困難であった。

今回、部品間のガタつき、摩擦等の製造ばらつきを考慮した機構解析モデルを構築した。さらに、設計上流段階で品質工学を適用して、製品ばらつきが製品機能に与える影響を効率的に検証する手法を構築し、製造ばらつきによる設計手戻りを抑制した。この特集号の論文“製造ばらつきを考慮した機構解析技術”(p.43)で、遮断器を対象にした製造ばらつきを抑制する設計技術の事例について述べる。

### (3) 3Dデータの開発プロセスへの活用

これまで当社はベテラン設計者や熟練工のスキルを前提にした、ツールに頼らない開発プロセスを構築してきた。一方、海外メーカーは、設計・製造スキルに頼らない3Dをベースにした開発プロセスを構築しており現地設計・製造化の際、当社の開発プロセスを適用できない。グローバル化による機種展開の増加で生じる設計リソース不足に対応するには、海外拠点で現地仕様に合わせて設計する開発現地化を進める必要があり、現地で運用可能な開発プロセ

スの構築が必須である。

グローバル化に向け、海外拠点への展開を見据えて、まずは国内拠点を対象に、サプライチェーンマネジメントプロセスと開発設計プロセスの連携を明確にして、開発プロセス全体で3DデータによるIT活用の仕組み構築に取り組んでいる(図9)。開発設計プロセス改善の取組みとして、電気設計と機械設計の連携、設計と生産の連携を図り、設計から下流工程で3Dデータを活用できる仕組みを構築してきた。例えば、板金部品の3DデータからCAMデータを自動作成する仕組み、基板図面から3Dモデルを自動作成して筐体(きょうたい)との干渉チェックをする仕組み、3Dで配線指示する仕組み等を構築した。先進事業分野で構築したこれらの仕組みを他の事業分野に展開して底上げと定着化を図るとともに、据付け・保守・安全性評価のため3Dスキャナ、VR(Virtual Reality)システムの活用に取り組んでいく。

さらに、最新技術を活用した、CAT(Computer Aided Testing)による検査業務の効率化、3D単独図の実用化、3Dデータを活用した仮想生産ラインによる生産準備業務の効率化に取り組んでいく。この特集号の論文“3D測量データ

を活用した現地据付工事改善”(p.47)で、3Dスキャナを活用した事例について述べる。

### 3. むすび

当社での設計・検証技術の高度化に向けて“ソフトウェア設計”“電子回路設計”“機械・構造設計”的各分野でITを活用して設計生産性を向上させた事例の概要、取組みビジョンについて述べた。今後、ますます高機能・大規模システム化が進む製品開発プロセスや、グローバル開発環境の変化に対応した設計プロセス革新を継続し、もう一段高いレベルの成長を実現していく。

### 参考文献

- (1) 竹垣盛一, ほか: フロントローディング型開発設計への取り組み, 三菱電機技報, 80, No.10, 636~638 (2006)
- (2) 山下昭裕, ほか: 設計プロセス革新による開発効率化, 三菱電機技報, 84, No.12, 660~663 (2010)
- (3) 中岡邦夫: 製品の設計初期段階で品質を作りこむ設計検証技術, 三菱電機技報, 87, No.4, 204~209 (2013)

# ドメイン固有言語を利用した設計手法

佐藤美和\*  
岡 藍子\*

Model-based Approach to the Design of Domain Specific Language

Miwa Satou, Aiko Oka

## 要旨

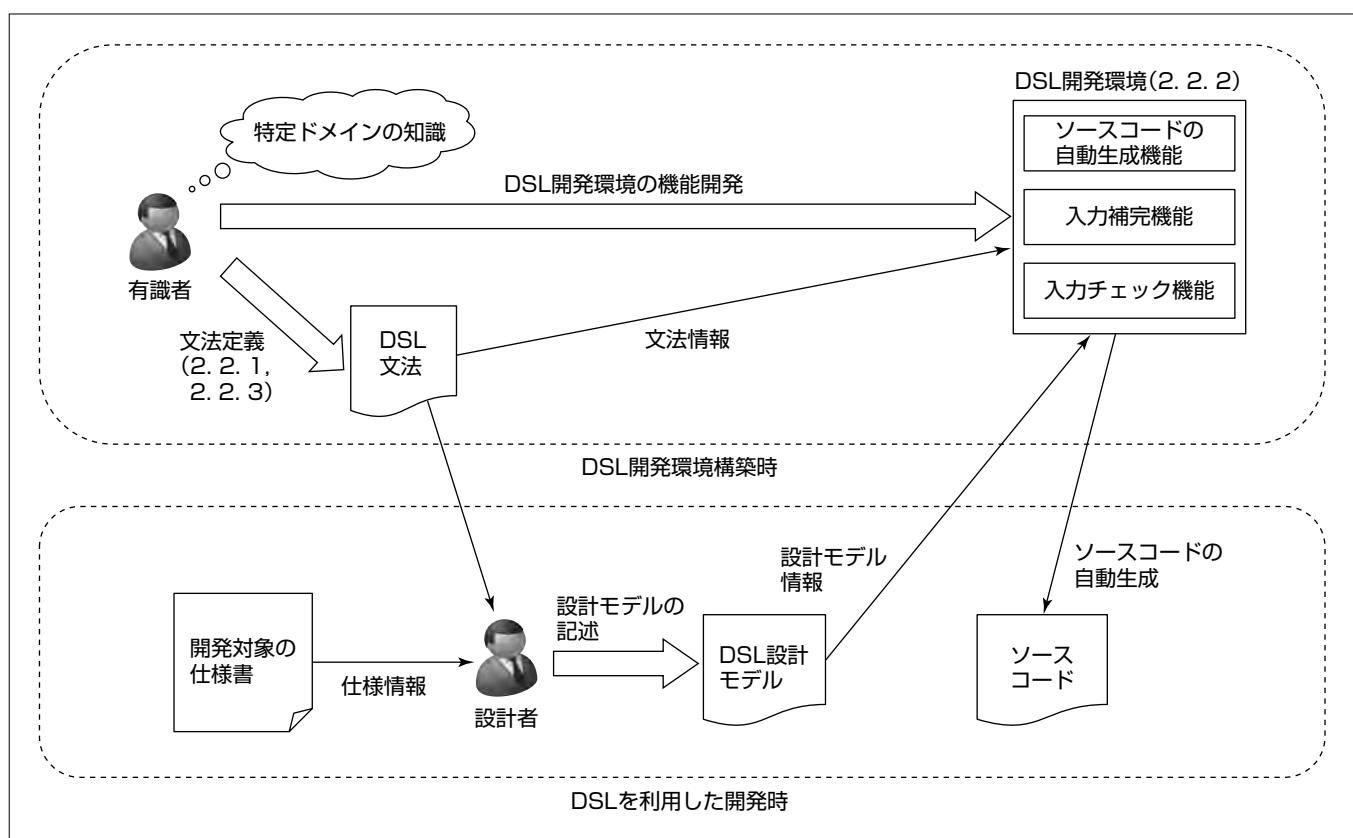
大規模通信システムの開発では、システムを構成する装置が多く、各装置を並行して開発する。このため、各装置の組合せ試験は開発終盤にしか行えず、試験で不具合が検出されると、原因究明、ソフトウェア改修、再試験と手戻りが発生する。対策として各装置の通信応答を模擬した通信シミュレータによって、各装置間の試験を事前に実施しているが、装置数が多い場合、通信シミュレータ開発の負荷が課題となる。

今回、大規模通信システムでの装置間の通信応答を模擬した検証用通信シミュレータの通信パケット解釈処理部分の設計にドメイン固有言語(Domain Specific Language : DSL)を適用した。DSLを用いて、特定領域(ドメイン)のふるまいに特化した設計モデルを記述し、設計モデルから

ソースコードを自動生成する。

適用事例では、必要最低限の要素に当たるデータ名称、データ長、入力値範囲の制約など、通信インターフェースの仕様の約80%をDSL文法として定義した。また、DSLを用いて記述した設計モデルからソースコードの自動生成を実現することで誤りが混入する可能性を低減した。さらに、あらかじめ決められた値や通信インターフェースの仕様を表現する要素だけを記述できるように入力補完機能などを加えることで、誤った値や内容が記述された場合には警告メッセージを表示するので、意図しない設計の防止が可能である。

今回構築したDSL開発環境では、通信パケット解釈処理部分の開発工数を70%削減、仕様解釈の誤りに起因するシステムテスト工程での不具合件数0件を実現した。



## DSLによる開発スタイル

DSLによる開発スタイルは、DSL開発環境構築時とDSLを利用した開発時の2段階で構成している。DSL開発環境構築時には、特定ドメインの知識を基にDSL文法を定義し、DSL開発環境の機能(ソースコードの自動生成など)を開発する。DSL開発環境構築後は、設計者がDSLで設計記述すれば、ソースコードが自動生成される。

## 1. まえがき

携帯電話やデジタル家電、自動車など多機能なシステムでは、複数のハードウェアと複数のソフトウェアが連携して動作する。このようなシステムは多くの開発人員と開発期間を要することから大規模組み込みシステムと呼ばれ、複数企業で分担開発される場合が多い。

大規模組み込みシステムのソフトウェア開発では、設計、実装、試験の開発工程を時系列に分割して管理を行うウォーターフォール型開発手法が採用されており、各企業で開発した装置を組み合わせた試験は開発終盤のシステムテスト工程で行われる。この開発手法は前工程が問題なく完了していることを前提とするため、開発終盤で不具合が検出されると、原因を究明して不具合が混入された工程まで戻って修正する。この修正に伴い、不具合が混入した後の工程の開発もやり直す必要があるため、手戻り工数の削減が課題である(図1)。

本稿で取り上げる大規模通信システムの開発では、この課題を解決するために、開発装置と通信を行う対向装置との通信を模擬する検証用の通信シミュレータが活用されている。シミュレータの活用によって、他装置との通信機能試験を上流工程で実施し、後工程の試験不具合の発生率の低減を図ってきた。しかし、通信システムの規模が大きくなることで、システムを構成する装置数が増加し、装置間ごとのインターフェースの仕様が複雑・多様化してきている。これに伴い、対向装置ごとに開発を要する通信シミュレータの開発規模の拡大、仕様解釈の誤りによる不具合混入が問題となっている。

これらの問題を解決するために、特定領域のふるまいに特化した設計モデルを記述するDSLと呼ばれるコンピュータ言語を導入した。ソフトウェア開発でのDSLの活用は、Martin Fowler等によって提唱されている<sup>(1)</sup>。特定領域のふるまいを記述するためのDSLの文法(以下“DSL文法”という。)を定義し、DSLによる記述結果である設計モデルからソースコードを自動生成する機能を作成しておくことで、開発工数削減と品質向上を実現した。

## 2. 通信シミュレータ導入での課題と対策

### 2.1 課題

大規模通信システムは、複数のハードウェアやソフトウェアで構成し、複数の企業で開発している。他社システ

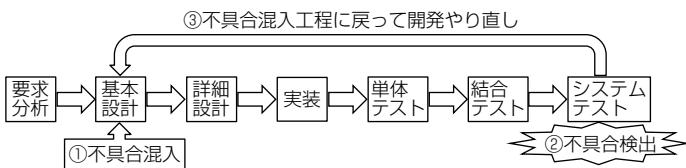


図1. 不具合検出による手戻り

ムや並行して開発している装置との組合せによる通信機能試験は、開発終盤や現地でしか行えない。試験で発生した不具合の多くは、通信インターフェースの仕様を記述するときの曖昧性に起因しており、原因究明、ソフトウェア改修、再試験によって大きな手戻りとなっている。

そこで、開発装置と通信を行う対向装置との通信を模擬する検証用の通信シミュレータを導入し、通信機能試験の上流工程での実施を図ってきた。しかし、通信シミュレータはデータ通信を行うための信号構成や規約を定義した通信インターフェースの仕様が装置ごとに異なるため、大規模通信システムを構成する全装置を模擬する通信シミュレータを開発する必要がある。通信機能試験を上流工程で実施するためには、通信シミュレータ開発の省力化と期間の短縮が必要である。

### 2.2 対策

通信シミュレータは、パケットの送受信処理、解釈処理、表示処理で構成している。このうち、送受信処理、表示処理は、通信インターフェースの仕様に依存しないためソースコードの流用が可能だが、解釈処理は、通信インターフェースの仕様が異なる装置ごとに設計・実装が必要である。そこで、通信インターフェースを対象ドメインとしたDSLの文法を定義し、DSLで記述した設計モデルから解釈処理のソースコードを自動生成するDSL開発環境を構築した(図2)。設計者は、設計を補助する入力補完機能、入力チェック機能、DSLの図式記述を活用して設計モデルを記述することで、ソースコードの自動生成が可能となる。

#### 2.2.1 DSL文法の定義

自然言語による仕様の記述には曖昧性が含まれ、仕様解釈の齟齬が発生するおそれがある。そこで、記述の曖昧性を排除するため、通信インターフェースの設計モデルの記述に特化したDSLの文法を定義し、設計者は、DSLで設計モデルを記述することにした。DSL文法は、実装を意識した明確な設計モデルの記述ができるように、製品やシステムの暗黙知も含め、通信インターフェースの仕様を明確に理解している有識者が定義する。DSLで記述された設計モデルからソースコードが自動生成される。このDSL文法を通信インターフェース文法という。図3に、設計モデルを記述するための通信インターフェース文法と、通信インターフェース文法を用いて作成した設計モデルの例を示す。通信インターフェース文法は、信号種別、送信先アドレス等のデータ名称、数値型や文字列型などデータの内容を解釈するのに必要なデータ型、データ長や値の範囲等の制約事項など、通信インターフェースの設計モデルを表現する要素で構成する。

#### 2.2.2 DSL開発手法の定義と機能

##### (1) ソースコードの自動生成機能

DSLで記述した設計モデルからソースコードを自動生成する機能は、通信インターフェースの仕様に依存しない処

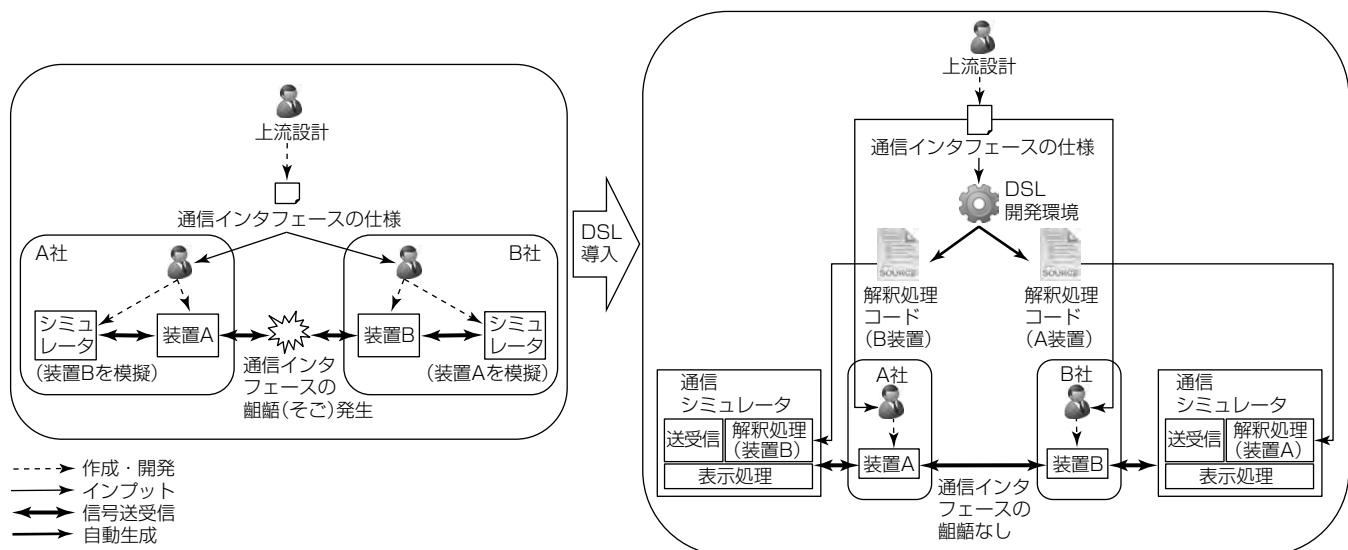


図2. 通信シミュレータへのDSL導入イメージ

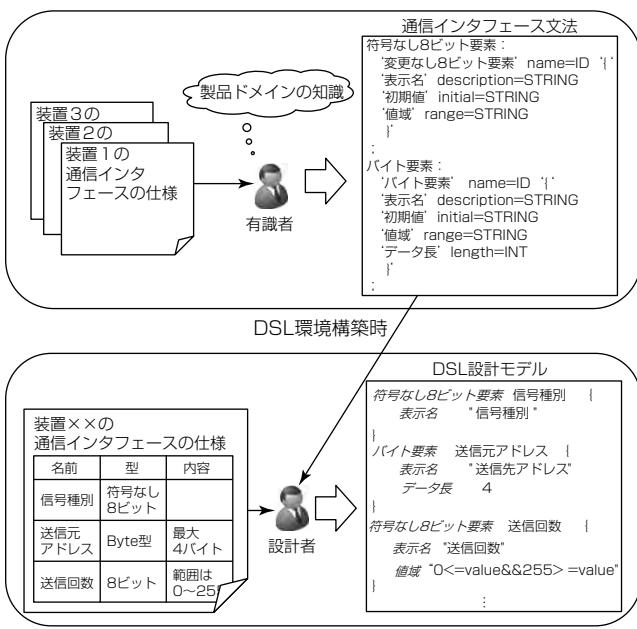


図3. DSLによる仕様記述例

理をパターンコードとしてあらかじめ保持しておく、仕様ごとに異なるパラメータ値等の情報をDSL設計モデルから直接パターンコードに埋め込むことで実現する(図4)。自動生成によって、開発工数が削減されるだけでなく、仕様解釈の誤りや実装時の人的ミスの可能性が大幅に減少する。自動生成対象のソースコードは、設計モデルによる変動要素が多い箇所とする。変動が少ない箇所は、通常の汎用言語による実装を行い、流用部分としてテンプレート化し、自動生成コードとテンプレートコードを組み合わせて製品コードとする(図5)。通信シミュレータでは、通信インターフェースの仕様の設計情報が多く含まれる解釈処理を自動生成対象とし、通信インターフェースの仕様に依存しない送受信処理、表示処理をテンプレートコードとした。

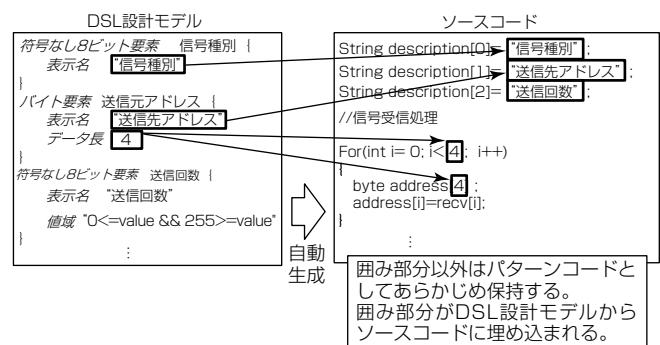


図4. ソースコード自動生成の仕組み例

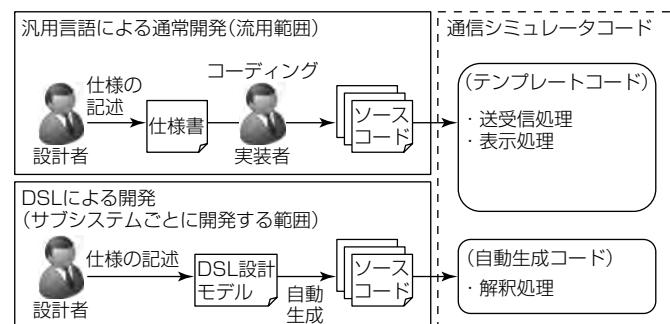


図5. DSLを利用した通信シミュレータの開発

## (2) 設計を補助する入力補完機能

入力補完機能は、任意の設計モデルの記述に対して入力候補となり得る内容を事前に設定しておき、DSLで設計モデルを記述する際に、入力候補を表示することで設計誤りを未然に防ぎ、設計を効率化する。例えば、通信インターフェース文法の構成要素であるバイトオーダー(メモリ配列への格納順序)には、最上位データから格納するビッグエンディアン、最下位データから格納するリトルエンディアンがあり、必ず一方を指定する必要がある。設計者がバイトオーダーを記述する際に、候補としてこれら2種類を表示し、ど

ちらか一方を選択すると選択した文字列が自動で入力される。

### (3) 設計を補助する入力チェック機能

入力チェック機能は、任意の設計記述に対してエラーとなり得る内容をあらかじめ設定しておき、DSLで設計モデルを記述する際に、誤った値や内容が記述された場合には警告メッセージを表示することで、意図しない設計を防ぐことが可能となる。例えば、符号なし整数8ビット型の場合には0～255しか入力できないため、0～255以外の場合に、“符号なし8ビット型は0～255である”という警告メッセージを表示し、早期に誤り発見を促す(図6)。

### 2.2.3 DSL手法の利用容易化に対する開発環境の構築

DSLで記述した設計モデルからソースコードが自動生成されると、実装時の人の介在が低減され、開発の効率化が見込まれる。しかし、通信インターフェースの仕様は仕様書に図表で記載されており、DSLを用いて設計モデルを記述するには人手で行う必要がある。大規模通信システムを構成する装置が数十個であれば、DSLを習得した技術者1人で通信シミュレータの作成は可能であるが、装置が数百個の場合1人では開発期間がかかりすぎるという問題がある。この場合、通信シミュレータを複数人で作成することになるが、DSLを初めて使う技術者はDSL文法を習得する必要があり、特にプログラミングを不得手とする技術者にはDSL導入の敷居が高くなる。そこで、設計モデルの記述を、DSLの代りにブロック図を用いてできる機能を実現することで設計モデル記述の容易化を図った。通信インターフェースの仕様のデータ型ごとにブロックを用意し、ブロックにデータの名称・説明、データ型に合わせた値の範囲を定義する“プロパティ”，ブロック間をつなぐ“シンボル”，ブロック間の“接続可否の定義”によるブロック図式を定義した。

図7に通信インターフェースの設計モデルをブロック図で記述した例を示す。信号のデータ部分を示すデータ部のブロックの下に左から順にデータ型のブロックをおき、シンボルでつなぐことによって信号のデータ部分を表現する。

### 2.3 効 果

今回、大規模通信システムでの装置間の通信応答を模擬する通信シミュレータ開発用に、通信インターフェースの仕様の約80%をカバーするDSL文法を定義し、DSL開発環境を構築した。これによって、通信パケット解釈処理部分の開発工数が約70%削減された。また、通信インターフェースの仕様をモデル化したことで、設計と実装が標準化され、仕様解釈の誤りに起因するシステムテスト工程の不具合件数0件を実現し、開発終盤での不具合検出による手戻り発生を抑止した。

### 2.4 今後の課題

装置間の連動する部分の自動試験で、送受信データのテストデータを人手で作成しており、自動試験環境構築に時

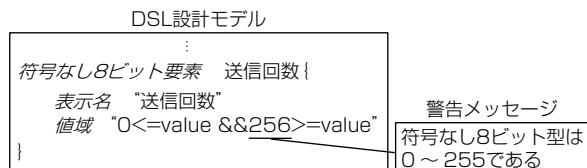


図6. 入力チェック機能による警告メッセージ例

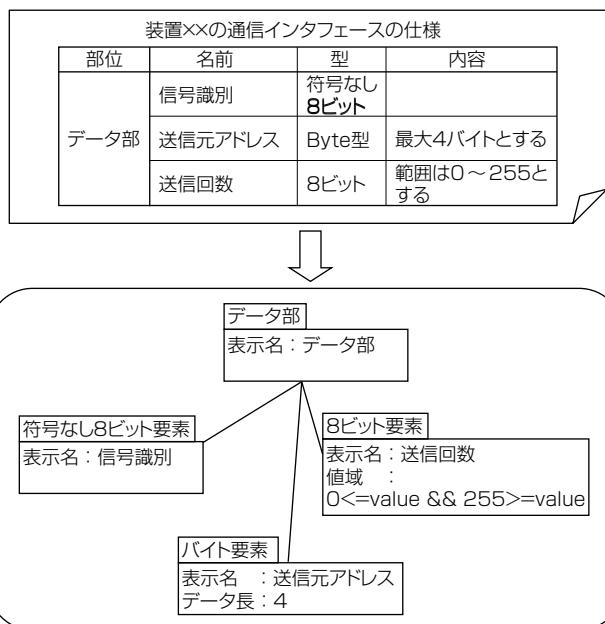


図7. ブロック図による記述例

間と人手がかかっている。今後はテストケース定義と通信インターフェース仕様からのテストデータ自動生成による試験工程の工数削減が課題である。

## 3. む す び

大規模通信システムの開発では、システムを構成する装置を並行開発しており、各装置の組合せ試験は開発終盤に行うため、不具合発生時の手戻りが大きくなる。対策として、各装置間の通信を模擬した通信シミュレータを導入しているが、装置数が多い場合、通信シミュレータの開発が負荷となる。そこで、DSLを利用したソフトウェア設計手法と通信シミュレータにDSLを導入した事例について述べた。

今後も、システムの大規模化及び要求される品質の高度化に伴い、特定ドメインを対象とした設計・実装のモデル化はますます重要になってくる。DSLを設計だけではなく試験にも活用できるよう進化させ、開発工数削減と品質向上に役立たせていく。

## 参 考 文 献

- (1) Fowler M., et al.: Domain-Specific Languages, Addison-Wesley (2010)

# プロセス改善技術者育成コースの設計

久野倫義\* 近藤聖久†  
中島毅\*\* 小笠原公一††  
芝田晃\*\*\*

*Design of Training Course for Software Process Improvement Engineers*

Noriyoshi Kuno, Tsuyoshi Nakajima, Akira Shibata, Kiyohisa Kondo, Koichi Ogasawara

## 要旨

近年、多くのソフトウェア開発部門のプロセス改善に取り組んでおり<sup>(1)</sup>、その改善の核となる改善推進グループ(SEPG: Software Engineering Process Group)の重要性が指摘されている<sup>(2)</sup>。しかし、SEPGを構成するプロセス改善技術者(以下“SEPG要員”という。)は、高い問題解決力と、ある程度の開発経験とパーソナリティを必要とするため、その確保が難しい。そのため、質の高いSEPG要員を効率良く育成することはプロセス改善を進める上で重要な課題である。しかし、SEPG要員の育成には、多岐にわたるスキルが要求され達成目標を定めにくい。三菱電機

グループはこうした課題を解決するプロセス改善技術者育成コースを開発した。同コースは、下の表に示す単元／講義項目及び達成目標／内容に示すように、現場のSEPG要員が会う困難を解決するためのスキル習得に目標を絞り、短い講座期間での実用的なコースを実現した。また、三菱電機グループ全体を母体として育成対象者と指導者を選出することで、集合形式コースで効率良く育成できた。

受講後のアンケート調査などによって、受講が成長に貢献したという回答が89.8%であったことから、このコースが有効であることが分かった。

単元／講義項目	達成目標／内容
1 SEPGとは	SEPGの役割・意義を知り改善活動の流れを理解する
1 SEPGがなぜ必要か	①事業課題から展開された組織QCD(品質、コスト、工程)の向上が目標 ②標準プロセス・インフラの確立・維持・向上→組織能力の向上
2 プロセス改善のPDCA(Plan Do Check Action)	①SEPGはどんな流れで活動するか：年度計画・プロジェクトのPDCAサイクル、SQA(Software Quality Assurance)との関係
3 プロセス改善のプロセス	①SEPG活動内容と具体的な活動項目
4 改善推進体制と改善の進め方	①SEPG体制のタイプ別特徴と留意点、②プロセス改善虎の巻Q&A紹介、(G演習1)自職場SEPG機能の過不足
2 プロセスに焦点を当てる	プロセスの重要性とプロセスの構成要素を理解する
1 プロセスに注目する理由	①仕事の段取り(プロジェクト、個人)と制御(進捗、品質) ②標準化→コミュニケーションの向上、ノウハウの蓄積・再利用、教育の容易化、見える化、改善・評価が進む
2 プロセスを制御する	①プロセスアプローチ：高品質の製品は良いプロセスから ②プロセス品質を制御することでプロダクト品質を達成する
3 プロセスモデルとその意義	①モデルを活用して組織の(あるべき)プロセスを定義する ②世界の企業間で共通の仕事のやり取りができる ③国際標準の動向(ISO/IEC 15288, 12207, 15504)
4 V字モデルの意義	①設計と検証の対応関係をとる、②品質要求を含む要求の定義とその追跡性の確保が重要
5 ライフサイクルモデル	①ウォーターフォール、インクリメンタル、エボリューションモデルの意味、強み弱み ②ライフサイクルの組立て
6 プロセスの要素	①プロセスチャートを使ったプロセスの構成要素の説明、(G演習2)“契約見積り”のプロセスチャートを作成
3 モデルベースの改善	モデルベースの改善の流れを理解する
1 モデルをものさしにする	①プロセスモデルとの差異で、改善機会を抽出：改善機会の優先度は、事業課題の達成への貢献で判断
2 モデル体系	①ISO/IEC 15504のモデル体系
3 QMS(Quality Management System)整備との関係	①モデルベースの改善はQMS整備と矛盾する活動ではない
4 改善のアプローチ	①改善活動でプロセスモデルを活用する
4 問題解決手順	ワークショップ実施に必要な技術・技法として、問題解決手順の流れと考え方を理解する
1 事業目標、改善スコープ・目標の設定	①事業目標から改善目標へブレーカダウン、②目標(To)と現状(From)のギャップ認識、③データからの傾向分析
2 プロセス点検	①開発プロセスガイドに沿った主要プロセスのプロセス成果と基本プラクティス、レベル2、3の共通プラクティスの理解、②プロセス点検の実施方法
3 対策の立案	①取組課題の策定化、②投資対効果に基づく優先順位付け、実施施策の選択、③計画(担当、スケジュール、監視・制御)、現場の巻き込み方
4 改善の量定化	①改善を進めるための指標の重要性、②改善の活動、効果のメトリクスの考え方、③指標の設定・評価方法
5 組織展開	改善の組織展開に向けた具体的なノウハウを共有する
1 改善の定着	①組織展開の重要性の意識付け、②具体的な展開のノウハウなど(教育、体制、評価方法等)、③(G演習3)プロセス改善推進上の課題と取組みの共有

## 三菱電機グループのプロセス改善技術者育成コースにおける単元／講義項目及び達成目標／内容

高い問題解決力と、ある程度の開発経験とパーソナリティを必要とし、その確保が難しい質の高いSEPG要員を効率良く育成するために開発した育成コースであり、効果的、効率的にSEPG要員を育成することが可能である。

\*三菱電機(株) 設計システム技術センター(工博) \*\*芝浦工業大学(工博) \*\*\*三菱電機(株) インフォメーションシステム業務部  
†同社 設計システム技術センター ††同社 生産技術部

## 1. まえがき

質の高いSEPG要員を効率良く育成することはプロセス改善を進める上で重要な課題である。一方、SEPGの業務は重要かつ多岐にわたり、Humphrey<sup>(3)</sup>は、SEPGの役割を①プロセス標準の確立、②プロセスデータベースの維持、③技術導入支援、④プロセス教育提供、⑤コンサルテーション、⑥定期的なアセスメントと状況報告と定義した。

これら①～⑥を実施できるSEPG要員には、プロセスに関わる技術だけでなく、開発技術や管理技術、パーソナリティなど広範囲なスキルが必要である。そのため、SEPG要員の育成コースを作る場合、求められる幅広いスキルの範囲に対して、どのように達成目標を定めるかが重要となる。

こうしたSEPG要員育成の課題を解決するために、三菱電機グループはプロセス改善技術者育成コース(以下“SEPG要員育成コース”という。)を企画・設計した。

本稿では、ワークショップを中心としたこのコースの設計・実装、コース評価及び今後の課題について述べる。

## 2. SEPG要員育成コースの検討

### 2.1 SEPG要員育成が必要となった背景

三菱電機グループでは、全社的な枠組みを作り、ソフトウェア開発改善活動を推進している。三菱電機の各事業所とソフトウェア開発を請け負う関係会社(以下“事業所”という。)が、年度ベースで事業所のソフトウェア開発の実態把握と改善施策を立案する。各事業所の自律的改善を進めるため、2007年度にプロセス改善推進体制を強化することが課題とされ、事業所の改善推進者とコーポレートのSEPG要員が集まり、プロセス改善に関する課題解決を行う場としてEP-WG(Engineering Process Working Group)を設立した。このWGの中でISO/IEC 15504 Part 5<sup>(4)</sup>のプロセス記述の解釈や、改善情報の交換を進め、SEPG要員は事業所のコア人材として育っていった。

次に、自律的なプロセス改善を促進するために、事業所のSEPG要員をソフトウェア技術者に対して一定比率にする目標を立て、2年でほぼ達成した。しかし、SEPG要員は、一定の開発経験を持つことを条件に集められたものの、本人の業務に対する動機付けや改善を推進するスキルが不足しており、確保した要員の活用が十分できていないことが問題となった。そのため、現有及び将来のSEPG要員の質向上を次なる課題として設定した。

### 2.2 SEPG要員の現状と育成ポイント

SEPG要員に対する教育は、ISO/IEC 15504 Part 5等で示されるプロセスの理解など、抽象的な知識を詰め込む形式に陥りやすい。そのため、講座を受講して獲得した知識をすぐに開発現場の問題解決に利用することが困難という課題があった。そこで、SEPG要員育成コースの目標

を、SEPG要員が現場で直面する問題を解決できることとし、EP-WGでSEPG要員が現場で直面する問題を抽出し、次の4つの課題に集約した。

#### (1) 課題1：役割の不明確さ

SEPGの役割と責務を明確に定義していないことが多い。その場合、真の改善を推進していく役割と能力が見過ごされがちとなる。また、SEPG要員も限られたリソースの中で活動範囲と優先順位を個別に設定して行動し、組織全体として整合性を欠いた改善活動となってしまう。

#### (2) 課題2：経営者の理解と現場の賛同の欠如

ハードウェアの生産ラインでは、ライン効率や歩留まり率など改善指標と経営数値の関係が比較的明確なため、組織的な合意の下に改善が進みやすい。しかし、ソフトウェア開発では、価値・規模に相当する生産量の定義に決定打がなく、改善効果の定量化を軽視する傾向がある。改善を導入する際の初期コストと運用コストを超える改善効果を示せなければ、改善活動に賛同を得ることは難しい。

#### (3) 課題3：プロセス理解のない問題分析と解決

プロセス改善には、ISO/IEC 15504等のプロセスモデルという実績のある道具があるが、知っていても使いこなせていない。そして、職場の問題に対し、現状のプロセス分析をしないか、分析しても答えありきで、概して問題に対して的外れであるか、考慮が欠けたものとなり、失敗の確率が高い。

#### (4) 課題4：PDCAのない改善活動

改善活動は、対象プロジェクト中心に準備と適用を計画し、適切な時期にその進捗と効果を確認し、必要なら再計画しながら進むべきである。そうでない場合、改善活動が無理や無駄を生み、中途半端になりがちである。

### 2.3 SEPG要員育成コースに対する要求事項

EP-WGが設定したSEPG要員育成コースの育成目標は、“ソフトウェア開発現場の現状を正しく認識でき、組織の事業目標を実現する効果的で実行可能な改善計画書を書き、説明できる”こととした。これを行うためには、2.2節で述べた課題を解決する能力がSEPG要員に求められる。4つの課題に対応して、受講生の達成レベルに対する要求事項を次のように設定した。

要求(1) SEPGの役割・意義を知り動機付けを持つ(課題1)

要求(2) プロセスモデルを用い改善する重要性理解(課題3)

要求(3) 開発プロセスガイドでプロセス点検ができる(課題3)

要求(4) 問題解決手順を実践し改善計画を作成できる(課題4)

要求(5) プロセス改善を経営の問題と考え、経済的合理性を訴えることができる(課題2)

ここで、開発プロセスガイドとは、ISO/IEC 15504の各プロセスを三菱電機グループの組織がどう実装すべきかをガイド化したものであり、改善のための問題解決手順は、プロセスモデルベースの改善を実施するための手順である。

### 3. コースの設計

#### 3.1 コース構成

図1に設計・実施したコーススケジュールを示す。受講日数は要求事項を満足するように、全体で4日間、連続は3日以内となっている。

#### 3.2 座学の構成

座学は、SEPG業務に対する正しい認識と基礎的な知識の習得及びワークショップの導入の位置付けを持っている。グループ演習を組み合わせ、知識の定着を図るとともに、事業所間の状況や課題を情報共有する機会としている。各受講者に対し、事業所全体の活動と自らの活動内容を検討する事前課題を与え、座学による解説の後、その題材を基に所属する部門のSEPGの活動内容の過不足を議論させる演習を実施する。

#### 3.3 ワークショップの構成

##### (1) モデルベースのプロセス改善と問題解決手順

モデルベースの改善アプローチの有効性は一般によく理解されており、SEPG育成の取組み事例でもCMMI(Capability Maturity Model Integration)等のプロセスモデル習得やアセッサ育成の内容をプログラムに組み込んでいることが多い。しかし、ISO/IEC 15504<sup>(4)</sup>等を理解しても、そのままでは記述の抽象度が高く応用が難しいので、抽出した弱みや改善機会を、有効で実施可能な施策に落とし込むことが困難である。そこで、問題解決の手順を定義し、手順の中でプロセスモデルベースの改善を、②(i)(ii)及び③(i)として位置付け、適用を容易にする工夫を入れた。

##### ①問題分析：問題を設定する。

- (i)改善テーマ、ビジネスドライバを設定する。
- (ii)問題と考える現状QCD状況を把握する。
- (iii)狙いとすべき目標QCDを設定する。
- (iv)目標と現状ギャップを問題として認識する。

##### ②問題分析：課題を抽出する。

- (i)問題となるプロセスとその悪い状態を洗い出す。
- (ii)原因を深掘りし、課題として抽出する。
- (iii)重要な課題に狙いを定める。

##### ③施策立案

- (i)課題をプロセスの枠組み(組織・プロジェクト

PDCA、標準化、インフラ、教育、評価・検証)に沿って施策化する。

- (ii)プロジェクトへの適用評価、施策スケジューリングを行う。
- (iii)改善の活動目標と効果目標を設定する。
- (iv)改善活動のコスト対効果を計算する。
- ④改善をステークホルダに述べる。

##### (2) 3つのワークショップとその実装

受講者が上記の問題解決手順を確実に習得できるようするため、3つのワークショップを設定した(図2)。

- ①事前課題：職場の課題定義と定量的データを収集する。
- ②WS 1：課題を述べ、1つの課題(課題1)を選択する。課題を受講者3人と指導者によって分析、施策立案、発表準備を行う。これによって、問題分析のやり方や施策を立てる手順やノウハウを共有する。
- ③発表会1：課題1に対する改善施策について受講生1は、組織関係者へのプレゼンテーションを想定して発表する。
- ④WS2, 3プレ：発表会の後、他の受講生の課題2, 3について、問題分析し、問題解決の方針を策定する。
- ⑤職場での自習：約1か月間職場で課題に取り組み、発表形式にまとめる。受講生1は、改善施策の洗練・詳細化し、受講生2, 3はWS 2, 3の指導結果に基づき、課題に対する改善施策を立案する。
- ⑥WS2及びWS3：課題2, 3に関する改善施策を、指導者と他の受講生でレビューし洗練化する。
- ⑦発表会2：課題1, 2, 3を発表する。

##### (3) 狙いとする効果

ワークショップを図2のように構成した狙いは、次の3点である。()内に2.3節の要求との対応を示す。

- ①職場の課題を解くことによって、各事業所が取り組むべき改善活動を計画する機会とする(要求(3)(4))。
- ②他事業所の課題を共同して解くことで、問題解決手順を繰り返し、手順習得と定着を促進する(要求(4))。
- ③組織関係者に改善施策を述べる想定で、受講者が発表し、改善施策の必要性と経済性を論理立てて述べる方法を習得する(要求(5))。

1日目	OT	単元1: SEPGとは	単元2	昼食	単元2: プロセスに焦点を当てる		単元3: モデルベースの改善
2日目	単元4: 問題解決手順			昼食	単元5: 組織展開	OT	WS1: 課題1の問題分析
3日目	WS1: 課題1の施策立案			昼食	WS1: 課題1の発表準備	発表会(課題1)	OT
(1か月)宿題実施期間(課題1内容強化、課題2と課題3の施策立案)							
4日目	OT	WS2: 課題2のレビュー		昼食	WS3: 課題3のレビュー		発表会(課題2,3)
	9	10	11	12	13	14	15
	16	17	18 (時)				

図1. SEPG育成コースのスケジュール

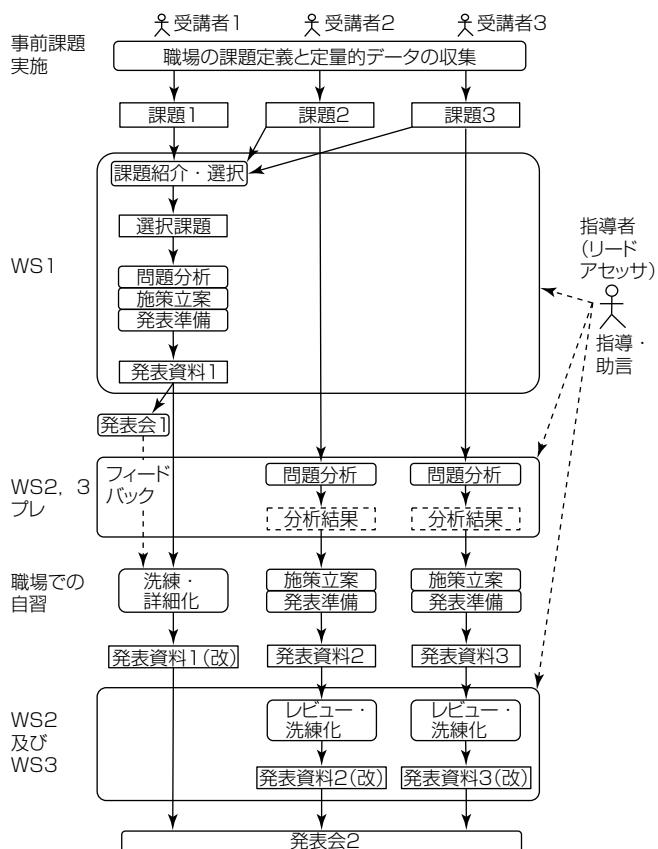


図2. ワークショップの構成と流れ

#### 4. コースの評価

##### 4.1 受講者アンケート調査結果とその分析

SEPG要員育成コース開始後から受講者に対して、アンケートで追跡調査した分析結果を示す。

###### (1) 理解度と役立ち度、成長の実感とコースの貢献度

全受講者の67.2%が改善推進者としての成長を実感していると回答し、実感していると回答した人の中で、このコースの受講が“大変貢献した”又は“まあ貢献した”と回答した人の割合は89.8%であった。受講生の実感としてのコースの有効性が分かる。

###### (2) 経験差による層別分析

経験で層別したコース効果を評価する。図3に、SEPG要員経験で層別した受講者数と、経験別の成長の自覚(“自覚あり”と回答した割合)、このコースで自ら提案した改善施策の実施状況(“概ね実施できた”と回答した割合)を示す。

SEPG要員としての経験年数が、1年以上の人は17名、1年未満の人は20名受講している。1年以上の人では全員が成長を実感しているのに対して、1年未満の人は60%になっており、効果が小さかったことが分かる。改善施策の実施についても、経験の長い人ほど実施できていることが分かる。1年未満のSEPG要員経験者にこうした傾向が現れるのは、改善施策を自ら主導できない立場にいて、このコースで得た知識と技法を利用する機会が少ないことが影響しているのではないかと考える。

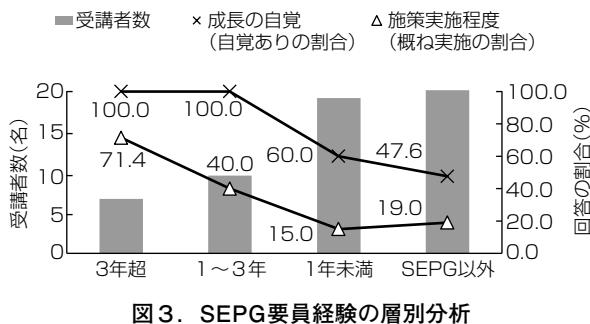


図3. SEPG要員経験の層別分析

#### 4.2 コースの課題

##### (1) 経験の浅い要員の職場における実践機会の確保

4.1節の(2)に示した経験差による層別分析から、経験の少ない要員は、コースで得た知識と技法の利用する機会が少ないと結論づけた。この問題の解決は重要であり、今後、受講者の上司と連携し、コース実施後の改善施策提案の実施に関するフォローを極めて細かく行うことなどを検討している。

##### (2) 指導者の計画的育成

コースを実施するためには5名以上のリードアセッサクラスの指導者が必要となる。そのため、指導者クラスの育成も計画的に行っていくことが必要である。

#### 5. むすび

SEPG要員の育成を企業グループ全体で効率的に行うために、三菱電機グループはSEPG要員育成コースを企画・設計して4年間にわたり実施し、事業所のSEPG要員育成に貢献してきた。プロセス改善における問題解決手順を職場の問題に繰り返して適用することで、リアルで実践的な能力の習得を行うことができる構成になっている。また、アンケート結果から、講座が高い質と育成効果を持っていることが分かった。

今後、育成効果の向上を図るために、受講者が受講後に改善施策を確実に実践できるようにフォローを行うことや、ワークショップを拡充することを検討していく必要がある。

#### 参考文献

- 小笠原秀人, ほか: 全社的なソフトウェアプロセス改善活動の実践結果とその振り返り, SQIPシンポジウム2011
- IPA : SEC BOOKS プロセス改善ナビゲーションガイド～虎の巻編～  
<https://www.ipa.go.jp/files/000005138.pdf>
- Humphrey, W.S. : Managing the Software Process, Addison-Wesley (1989)
- ISO/IEC 15504-5 : 2012, Information technology - Process assessment - Part 5 : An exemplar software life cycle process assessment model

# 静的解析技術によるソースコードの効率的な品質確保

藤本卓也\*  
中村勝彦\*  
浅川忠隆\*

*Efficient Quality Assurance of Source Code by Static Analysis Technologies*

Takanari Fujimoto, Katsuhiko Nakamura, Tadataka Asakawa

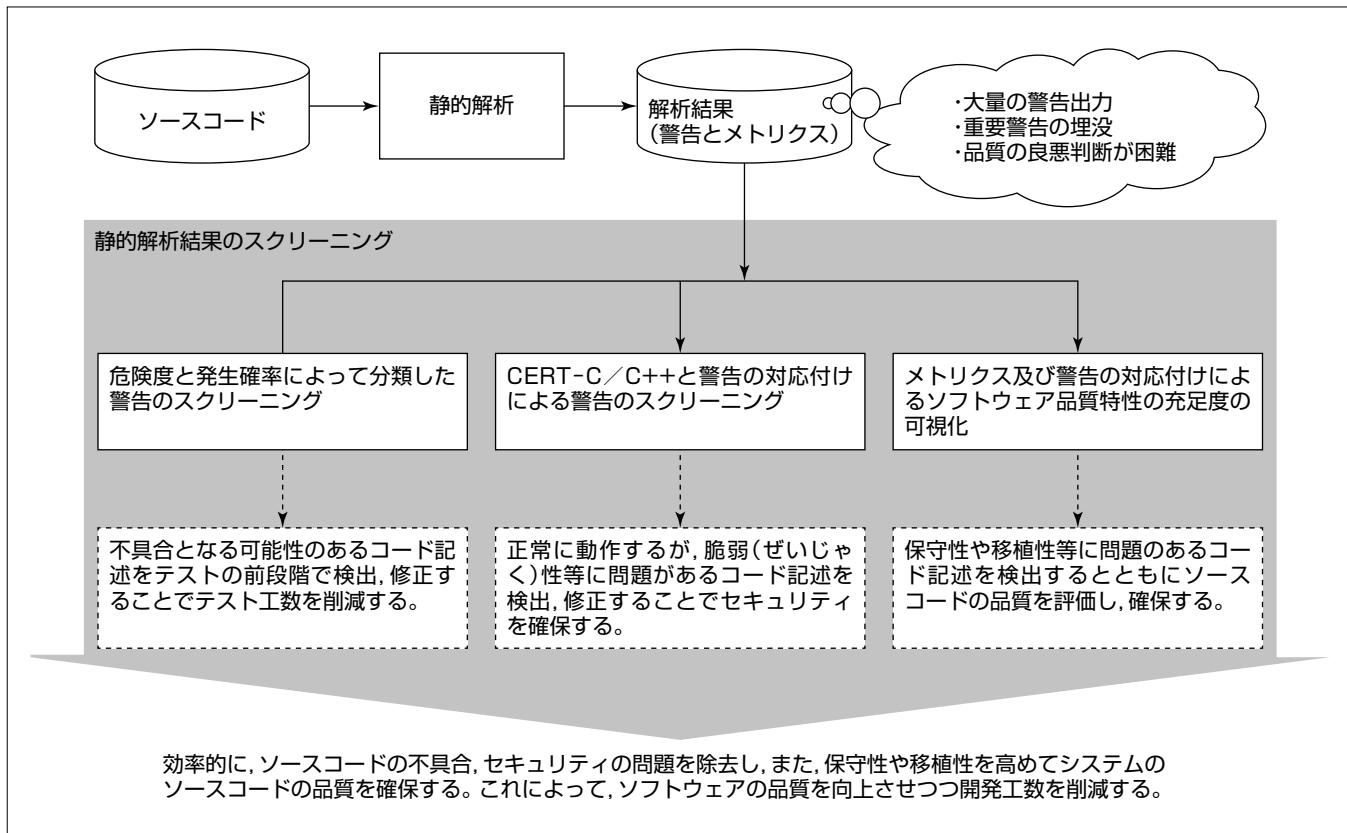
## 要旨

IoT(Internet of Things)機器の高機能化やネットワークの多様化によって搭載ソフトウェアの大規模化が進んでいる。これに対し、ソフトウェアのコーディング段階で、プログラムを実行せずにソースコードを機械的に解析する静的解析技術を活用し、内在する不具合となる可能性がある箇所(以下“問題箇所”という。)の検出作業を効率化し、フロントローディング率向上を図ってきた。

しかし、静的解析は警告の誤検出が多く、修正が必要な問題箇所の見極め作業に時間を要する傾向があり、大規模なソースコードに対しては誤検出による問題箇所の検出効率の悪化への対策が重要となっている。また、保守性やセ

キュリティ等のソフトウェア品質特性に関連付けたソースコード品質評価等、新たな観点による検証が求められており、静的解析結果から問題箇所を見極める作業の更なる効率化が必要である。

これら新たな課題への対策として、膨大な静的解析結果から問題箇所を選別するスクリーニング環境を構築して適用を図り、プログラム実行停止等につながる重大な問題箇所の絞り込み、静的解析結果の警告とメトリクスを組み合わせたソフトウェア品質特性への充足度の評価を可能とし、ソースコードの効率的な品質確保を実現した。



## 効率的にソースコードの品質を確保するための静的解析技術の適用

静的解析技術はソースコードの品質を確保するために有効であるが、警告が大量に出力されるため、不具合となる可能性のあるコード記述、脆弱性等に問題があるコード記述をいかに的確かつ効率的に検出するかが重要な力技となる。また、静的解析技術によって測定されるメトリクスをコードの保守性や移植性を維持するための尺度として使用できるが、的確な判断を行うにはどのメトリクスを使い、その値をどう評価するかが重要な力技となる。

## 1. まえがき

IoT機器の高機能化やネットワークの多様化によって、搭載されるソフトウェアの大規模化が進展している<sup>(1)</sup>。これに対して従来、ソフトウェアのコーディング段階で、プログラムを実行せずにソースコード(以下“コード”という。)を機械的に解析する静的解析技術を活用し、コードの問題箇所の検出作業を効率化し、フロントローディング率の向上を図ってきた<sup>(2)</sup>。

しかし、静的解析ツールは短時間に解析結果を得られる反面、プログラム実行を伴わない分、誤検出が多発し、真に修正が必要な問題箇所の見極め作業に時間を要する傾向がある。

このため、オープンソースソフトウェア(Open Source Software : OSS)活用の進展や多人数による開発によって大規模化が進むコードに対して、誤検出による問題箇所の検出効率の悪化への対策が重要となっている。加えて、保守性やセキュリティ等のソフトウェア品質特性に関連付けたコードの品質評価等、新たな観点による検証が求められており、静的解析結果から問題箇所を見極める作業の更なる効率化が必要である。

これら新たな課題への対策として、膨大な静的解析結果から問題箇所を選別するスクリーニング環境を構築して適用を図り、次に示すコード品質の効率的な確保を実現した。

- (1) コードの大規模化に対応した、プログラム実行停止等につながる重大な問題箇所の絞り込み作業の効率化
- (2) 静的解析結果からの警告とメトリクスを組み合わせたソフトウェア品質特性への充足度評価

## 2. 静的解析結果のスクリーニング環境

開発ライフサイクルの早期から効率的に実装品質を確保するため、静的解析結果である問題箇所を指摘する警告や複雑度等のコード品質を示す指標値であるメトリクスから、問題箇所や、ソフトウェア品質特性を悪化させるコード記

述を示す警告とメトリクスを選別するスクリーニング環境(以下“スクリーニング環境”という。)を構築し、活用を図っている(図1)。

スクリーニング環境は2種類のデータを入力とする。1つは解析対象コードに対して静的解析が出力する大量の解析結果(以下“解析結果”という。)、もう1つはスクリーニング用設定データ(以下“設定データ”という。)である。

解析結果は、警告とメトリクスの2種のデータから成る。設定データは、解析結果から抽出すべきプログラム実行停止等につながる重大な問題箇所となる度合い(以下“危険度”という。)に基づき分類した警告リスト、ソフトウェア品質特性評価用の警告リスト及びメトリクスと推奨値リストから成る。

スクリーニング環境は解析結果と設定データとを照合し、設定データの情報に基づき、解析結果から不具合の可能性が高いコード記述や、ソフトウェア品質特性を悪化させるコード記述やメトリクスを抽出して出力する。

ソフトウェア開発者は、スクリーニング環境が抽出した解析結果とコードを突き合わせることで問題箇所やソフトウェア品質特性への充足度を効率的に確認可能となる。

## 3. コード品質の向上を効率化するための対策

### 3.1 内在するコード問題点の効率的かつ正確な検出

静的解析技術は、保守性の悪化、プログラム実行停止等の問題を引き起こすコード記述を示す警告の出力やコードの品質を示すメトリクスの測定を行い、問題箇所の検出を支援する有用な技術である。しかし、問題箇所を広く検出する解析方針から解析結果には多くの誤検出が内在する。近年のIoT進展によってコード規模は増加し続けており、これまでの三菱電機の解析実績から、コード規模が5倍になると、誤検出を含む検出警告数は約22倍となる傾向を得ている(図2)。検出警告数が増加すると全検出警告の確認は困難となる。また、多数の誤検出の警告が含まれている場合が多いため、重要な警告が埋没してしまう危険性が高くなる。

このため、問題箇所を的確に検出するには、多数の誤検

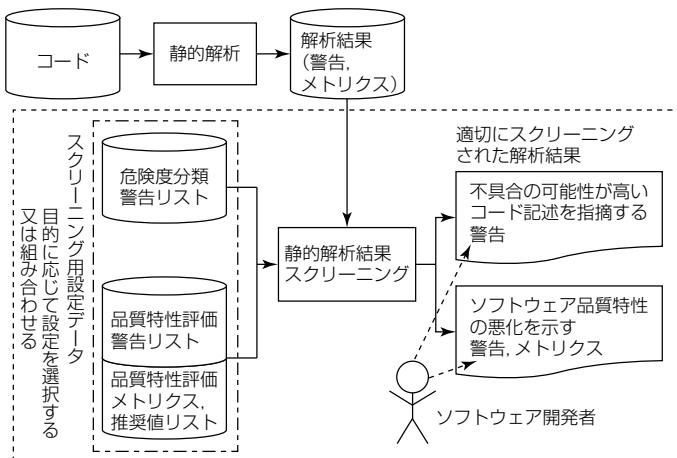


図1. 静的解析結果のスクリーニング環境の構成

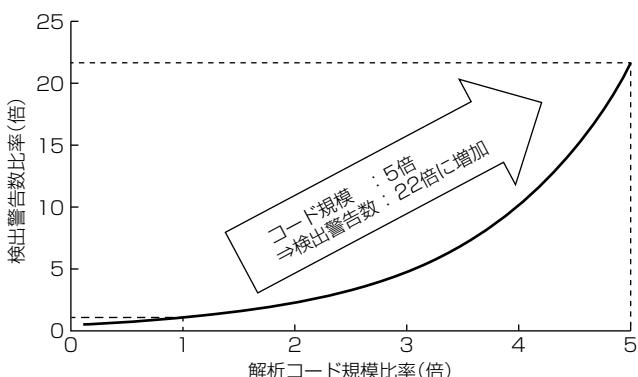


図2. コード規模と検出警告数の関係

表1. 危険度と警告の対応付け(例)

危険度	影響度	発生確率	分類基準	静的解析出力警告
A	高	高～中	不具合の可能性が高く、最優先で確認・対応すべき警告	<ul style="list-style-type: none"> <li>ゼロ除算が行われる</li> <li>式の評価順序が未定義になる</li> <li>配列の領域外アクセスが発生する</li> </ul>
B	高～中	高～中	不具合の可能性があり、確認・対応すべき警告	<ul style="list-style-type: none"> <li>異なるポインタ型間のキャストを実行している</li> <li>到達不能な文がある</li> <li>無限ループがある</li> </ul>
C	中	高～中	確認・対応を推奨する警告	<ul style="list-style-type: none"> <li>shortからcharへの暗黙変換が行われる</li> <li>戻り値が無視される関数がある</li> <li>値を返すvoid関数がある</li> </ul>
D	中～低	高～低	できれば確認することを薦める警告	<ul style="list-style-type: none"> <li>for文内でカンマ演算子が使われている</li> <li>long型からdouble型への型変換が行われている</li> <li>関数マクロが使われている</li> </ul>
E	低	高～低	無視しても問題にはならないと思われる警告	<ul style="list-style-type: none"> <li>long long型が使われている</li> <li>main関数の戻り型が誤っている</li> <li>inlineが関数以外で指定されている</li> </ul>

出を含む大量の警告の中から問題顕在化の確率が高いものを効率的に絞り込むことが要求される。これに対して、これまでの静的解析の知見に基づき、影響度と発生確率の2つの観点から、危険度という5段階の尺度を新設し、警告をグループ化した(表1)。

### (1) 影響度

検出された警告が実際に発生した場合にシステムが受けた悪影響を示す度合い(メモリ破壊、メモリリーク等の警告はプログラム実行が停止する等、致命的であり、影響度は高い)

### (2) 発生確率

警告が指摘する内容が実際に発生している確からしさを示す度合い(main関数の戻り型の誤りを指摘する警告の発生確率は100%だが、配列の領域外アクセスの発生を指摘する警告の発生確率は50%に満たない)

この分類の適用によって、不具合の可能性が高く、最優先で確認、対応すべき危険度Aに含まれる警告(以下“危険度A警告”といふ)は、検出される全警告の約25%に限定される(図3)。また、実際に解析を行った結果、危険度A警告だけを確認対象とした場合、検出された全警告を確認対象とした場合に比べ、確認すべき警告数は1/5となり、実際に解析を実施して検出された警告が不具合であった該当率(以下“ヒット率”といふ)は6%から30%へ向上し、約5倍の検出効率改善を確認した(図4)。

## 3.2 ソフトウェア品質特性に対する充足度評価

現状のソフトウェア開発で、OSSや既存製品コードを流用するケースは95%を占める<sup>(3)</sup>。このため、流用開発の効率維持では保守性等のソフトウェア品質特性の確保が重要な要素となっている。

ソフトウェア品質特性はJIS X 25010：2013<sup>(4)</sup>で8種の品質特性が定義されており、コードの品質については信頼性、保守性、移植性及びセキュリティの4種の特性が強く関連する。このうち、セキュリティについてはネットワークに接続されるソフトウェアの増加に伴い、これらのソフトウェアを狙った攻撃によって機密データの消失や漏えい

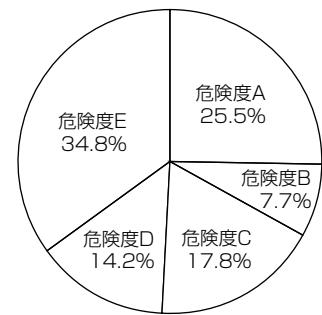


図3. 危険度によって分類した警告の比率

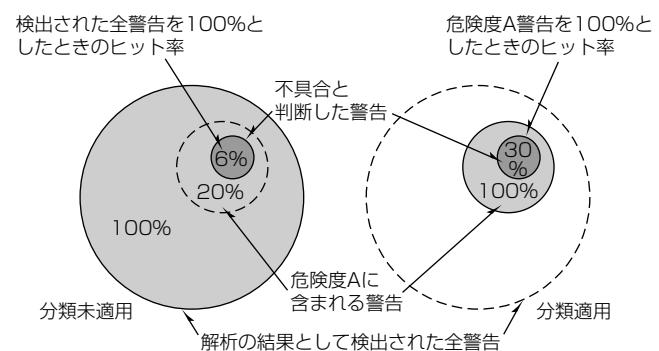


図4. 危険度分類を用いた検出可能な不具合の抽出率

といった被害が発生している。このため、脆弱性等の問題があるコード記述を早期に発見して修正することが重要になってきている。そこで、これら4種の品質特性について次に示す取組みを実施した。

### (1) 信頼性、保守性、移植性の取り組み

品質特性と、静的解析結果の警告及びメトリクスとを対応付け、二面からの充足度評価を可能とした(表2)。

### (2) セキュリティの取り組み

デファクトスタンダードとして活用が拡大しつつあるCERT-C/C++コーディングスタンダード<sup>(5)(6)(7)</sup>(以下“CERT-C/C++”といふ。)に記載されたセキュリティ上の欠陥を作りこませないためのコーディングルールと静的解析の警告とを対応付けし、コードのセキュリティ品質特性の充足度の評価に加え、セキュリティに関連する問題の検出にも利用可能とした(表2、表3)。

この対応付けの適用によって、静的解析が抽出する警告及びメトリクスから、流用開発の効率維持等で必要となるソフトウェア品質特性の評価、さらに、脆弱性等の問題があるコード記述の発見の効率性を向上させた。

## 4. むすび

大規模化するコードの誤検出警告による問題箇所の検出効率の悪化や、保守性やセキュリティ等のソフトウェア品質に関連付けたコードの品質評価の要求に対して、静的解析結果を選別して必要な情報だけを選別するスクリーニン

表2. ソフトウェア品質特性と静的解析のメトリクス及び警告の対応付け(例)

品質特性	品質副特性	具体的な特性	静的解析出力警告	静的解析出力メトリクス	
				名称	推奨値
信頼性	障害許容性 (耐故障性) 回復性	・エラー処理がきちんと実装されているか	・if-else-if文にelse節が又はswitch文にdefaultラベルがない ・例外処理が適切に行われてない		
保守性	解析性 モジュール性 再利用性 修正性 試験性	・ソースコードが読みやすいか ・全てのソースコードの書き方が統一されているか ・関数等の独立性が高いか ・試験が容易にできるか	・制御文の本体は中括弧で括る ・switch文のcaseがbreak文で終わっていない ・変数のスコープが適切でない ・1つのソースファイル内だけで使用される関数がstaticでない	複雑度	10以下
				マイヤーズインターバル	10以下
				関数行数	50以下
				推定静的パス数	200以下
				使用goto文の数	0
				凝集度	大きいほど良い
				結合度	
				ネスト段数	
				外部変数の数	小さいほど良い
				クラス継承段数	
移植性	適応性	・コンパイラやCPUが変わった場合、修正箇所が少ないか	・言語規約に記載されている未定義、未規定、処理系定義の機能を使用している ・エンディアン依存の処理がある		
セキュリティ	機密性 インテグリティ 否認防止性 責任追跡性 真正性	・想定外の条件を含め、全ての条件に対応する処理が記述されているか ・CERT-C/C++が遵守されているか	・if-else-if文にelse節が、又はswitch文にdefaultラベルがない	複雑度	10以下
				推定静的パス数	200以下

表3. CERT-C/C++コーディングルールと静的解析警告の対応付け(例)

種別	ID	概要	静的解析出力警告	CERT-C/C++コーディングルール	
				規則	規則
プリプロセッサ	PRExxx-C	字句の結合や文字列化を行う際のマクロ置換動作をよく理解する	・"##"演算子の使い方が誤っている		
宣言と初期化	DCLxxx-C	適切な記憶域期間を持つオブジェクトを宣言する	・関数内の自動変数のアドレスが外部ポインタ変数や戻り値等で関数外に持ち出されている		
	DCLyyy-CPP	領域確保／解放を行う対となる関数は同じスコープでオーバーロードする	・対となるnew演算子又はdelete演算子がない		
整数	INTzzz-C(CPP)	整数変換によってデータの消失や解釈間違いが発生しないことを保証する	・指定型サイズで表現不能な値が変数に設定されている		
配列	ARRzzz-C(CPP)	配列以外のオブジェクトを指すポインタに対して整数の加算や減算を行わない	・正しくないポインタ計算が実施される ・ポインタと整数の加減算が実施される		
オブジェクト指向プログラミング	OOPyyy-CPP	自分自身への代入を適切に扱う	・コピー演算子でコピー元と先のポインタが検査されていない		
雑則	MSCyyy-CPP	疑似乱数の生成にstd::rand()関数を使用しない	・rand()関数が使用されている		

グ環境を整備するとともに、プログラム実行停止等の重大な問題箇所を絞り込む方法、静的解析結果の警告とメトリクスを組み合わせてソフトウェア品質特性への充足度を評価する方法について述べた。

今後、IoT機器のコード規模の増加、流用開発の進展、ソフトウェア品質特性についても新たなセキュリティへの対策等が更に求められていく。

これらに対し、的確性に曖昧さが残る論理演算に起因する問題箇所の検出や、CERT-C/C++以外のセキュリティについての規格化への追従を図り、本稿で述べた環境、概念をさらに進化させ、ソフトウェア実装品質の効率的な確保、向上に努めていく。

## 参考文献

- (1) (独)情報処理推進機構：ソフトウェア開発データ白書 2007～2009/2010～2011/2012～2013/2014～2015
- (2) 中岡邦夫：製品の設計初期段階で品質を作りこむ設計

検証技術、三菱電機技報、87、No.4、2～7(2013)

- (3) (独)情報処理推進機構：組込みソフトウェア開発データ白書2015(2015)
- (4) JIS X 25010：2013(ISO/IEC 25010：2011)：システム及びソフトウェア製品の品質要求及び評価(SQuaRE)－システム及びソフトウェア品質モデル
- (5) JPCERTコーディネーションセンター：CERT Cコーディングスタンダード  
<https://www.jpcert.or.jp/sc-rules>
- (6) Carnegie Mellon University/Software Engineering Institute：SEI CERT C Coding Standard  
<https://www.securecoding.cert.org/confluence/display/c/SEI+CERT+C+Coding+Standard>
- (7) Carnegie Mellon University/Software Engineering Institute：SEI CERT C++ Coding Standard  
<https://www.securecoding.cert.org/confluence/pages/viewpage.action?pageId=637>

# コードレビュー手法の改善による コード品質確保の効率化

井元 薫\*  
細谷泰夫\*

*Efficiency of Code Quality Assurance by Improving a Method of Code Review*

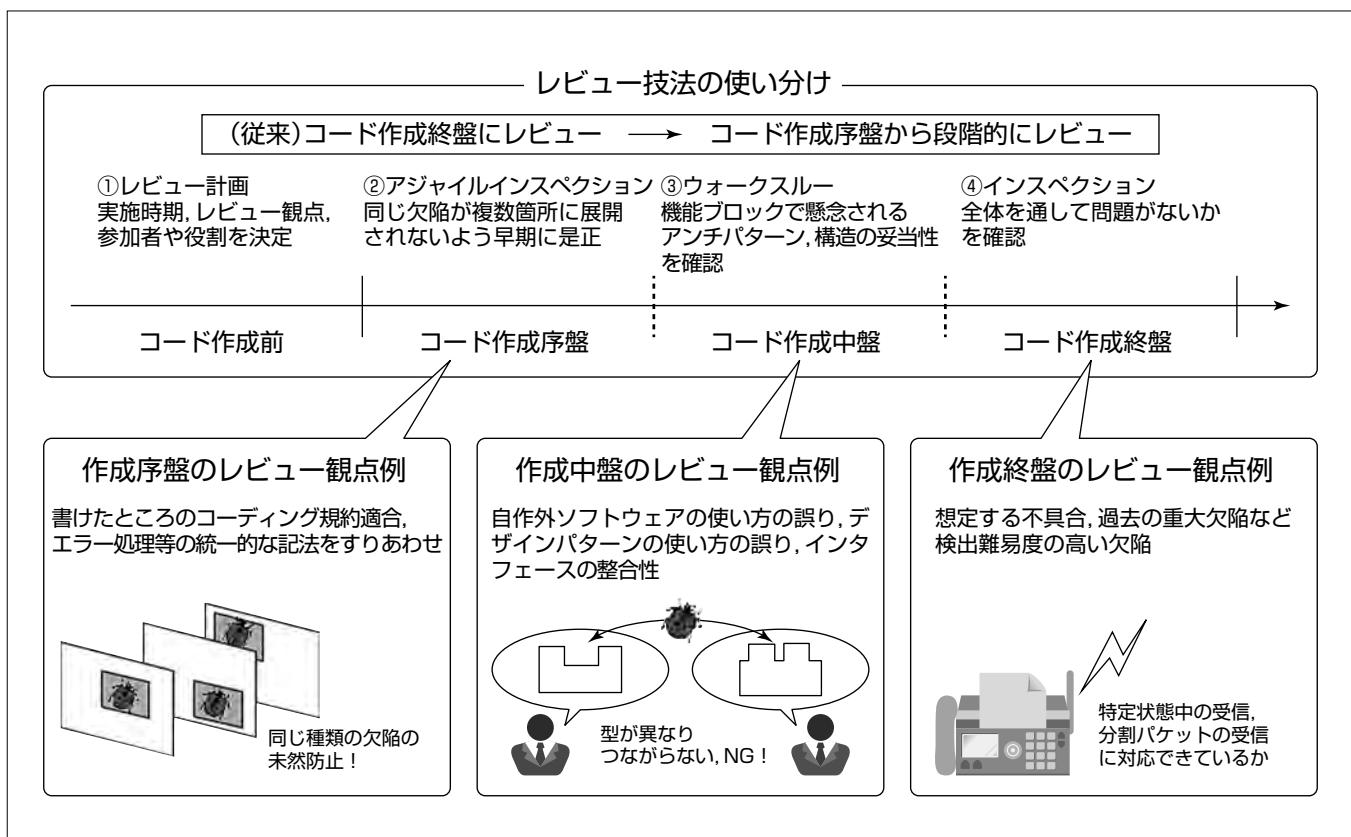
Kaoru Imoto, Yasuo Hosotani

## 要旨

近年、製品開発でソフトウェアの高機能化と短期開発が進んでいる。この状況下でソフトウェアのQCD(Quality Cost Delivery)を達成することが求められる。

ソフトウェア開発ではレビューとテストを組み合わせた品質確保が一般的である。ソースコードのレビュー(以下“コードレビュー”という。)では、ソフトウェアの誤動作の要因となる欠陥や、ソフトウェア構造崩れといった将来の保守性を阻害する要因等本質的な欠陥の除去が重要である。しかし、大規模化したコードに対し、限られたコストと期間でのレビューでは、指摘が軽微な欠陥に偏った場合に、先に述べた本質的な欠陥が除去しきれないままレビューが完了してしまうという課題があった。

指摘が軽微な欠陥に偏る原因を分析した結果、レビュー観点が曖昧であること、レビューアのスキル不足が主な原因であることが分かった。そこで、コードレビューで検出する欠陥を分類し、検出したい欠陥の種類に応じたレビュー観点を設定した。さらに、レビュー観点に応じた技法、タイミングを明確化した。レビュー計画で、必要な観点、技法、タイミングを計画し、適切なレビューアをアサインすることによって、レビューアのスキル不足やレビュー観点の重複回避が可能となった。これらの方策によって、限られた期間とコストで効果的なコードレビューの実施が可能となった。



## レビュー技法、レビュー観点、レビュータイミングの使い分けによるコードレビュー計画

コード作成フェーズによって、レビュー技法、レビュー観点、レビュータイミングを使い分けることで、軽微な欠陥の指摘は早期に、検出難易度の高い指摘は後期に集中させる。これによって、コード品質を確保する。

## 1. まえがき

近年、製品開発でソフトウェアの高機能化と短期開発が進んでいる。この状況下でソフトウェアのQCDを達成することが求められる。

ソフトウェア開発ではレビューとテストを組み合わせた品質確保が一般的である。ソースコードのレビューでは、ソフトウェアの誤動作の要因となる欠陥や、ソフトウェア構造崩れといった将来の保守性を阻害する要因等本質的な欠陥の除去が重要である。しかし大規模化したコードに対し、限られたコストと期間によるレビューでは、指摘が軽微な欠陥に偏り、先に述べた本質的な欠陥が除去しきれないという課題があった。

そこで、コードレビューで検出する欠陥を分類し、検出したい欠陥の種類に応じたレビュータイミング、レビュー技法、レビュー観点を設定することで、本質的な欠陥を効果的に検出可能なレビュー手法を開発した。

本稿では、コードレビューが抱える課題に対し、開発したレビュー手法とその効果について述べる。

## 2. コードレビューの課題

### 2.1 軽微な欠陥指摘への偏り抑制(レビューの質の問題)

ソフトウェア開発の大規模化、短納期化が進んでおり、コードレビューの時間が十分に取れないために欠陥が流出するリスクが増加している。そのため、コードレビュー

で効率的に欠陥を検出する必要がある<sup>(1)</sup>。従来、デザインレビューでは、セルフレビュー、ピアレビュー、書面レビュー、及び会議レビューを段階的に組み合わせて行っていた。しかし、次の2つの原因によって、軽微な欠陥への指摘に偏ってしまい、本質的な欠陥の除去が困難となる。

(1) レビュー観点が曖昧で、狙いが曖昧なまま目についた欠陥を指摘する。

(2) レビュアのスキル不足で保守性や移植性といった内部品質に関する指摘や本質的な欠陥が見つけられない。

### 2.2 従来のレビュー品質評価方法の問題点

品質を評価する手法の1つとして、ゾーン判定<sup>(2)</sup>がある。ゾーン判定は、図1のように表す。レビュー時間の基準値と、誤り検出率の目標範囲をともに満たしている場合に、品質が良いと評価される。図の星印のように、レビュー時間、誤り検出率がともに目標範囲内であり、かつ、図2のように、コーディング規約違反のような軽微な欠陥の指摘に偏っている場合は、ゾーン判定だけでは、コードレビューが十分に実施されていると判定されてしまう。このような場合に、重大な欠陥を見逃すリスクが大きくなる。

## 3. 対策

### 3.1 レビュー観点の体系化

#### (1) 欠陥の種類をレビュー観点として体系化

コードレビューで検出する狙いを定めやすくするために、次の観点で指摘すべき欠陥全体を図3のとおり4象限に分割した。

#### (縦軸)粒度大／小

大：複数箇所のコードを見て指摘できるもの。

小：1箇所のコードだけ見て指摘できるもの。

#### (横軸)誤り／誤りでない

誤り：不具合動作を起こす原因となるもの。

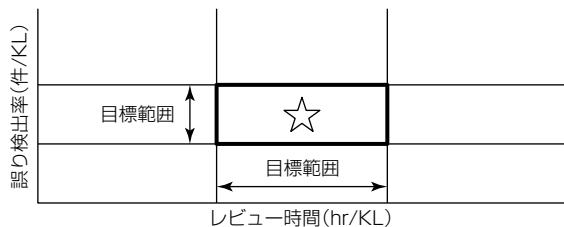


図1. コードレビューのゾーン判定

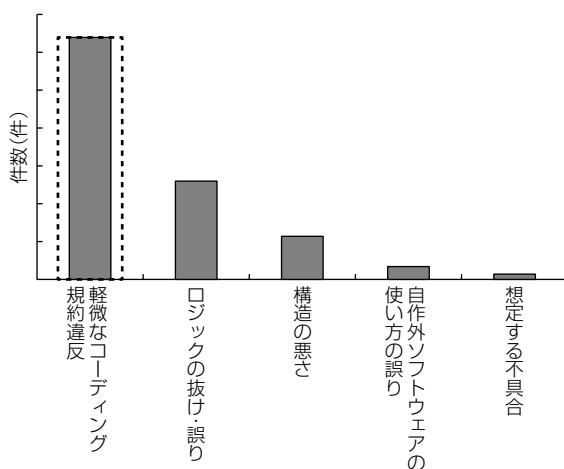


図2. コードレビューの欠陥種別

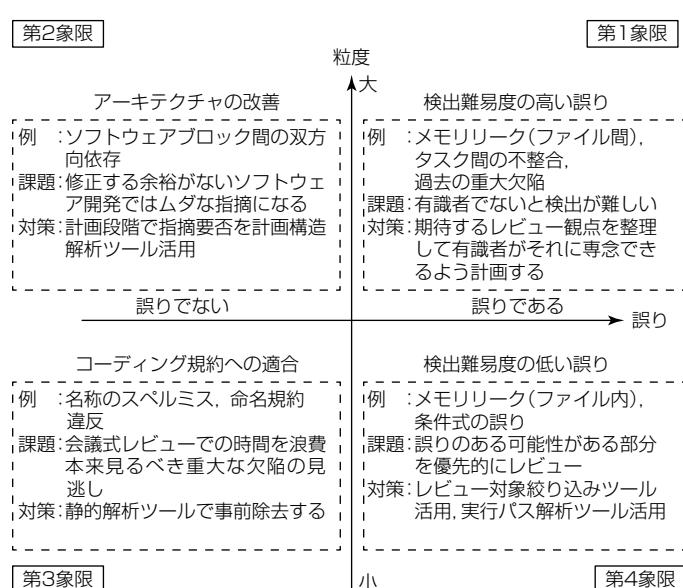


図3. 指摘すべき欠陥全体の4象限

誤りでない：改善提案、書き方に関する指摘、質問等。

正常動作するが、コーディング規約適合、保守性悪化等の影響があるもの。

例) 命名規則違反、スペルミス、複雑なソフトウェア構造

さらに、それぞれの象限に対応したレビュー観点を体系化した(表1)。例えば、構造の悪さは、複雑さ等構造に関する要件に対し、コードの適合を確認する観点でレビューを行う。レビューには、構造の良しあしを判断できるスキルが必要となるため、難易度は高くなる。また、構造の悪さは、複数のファイルにまたがる観点と、単一のファイル内に収まる観点があるため、対応する象限は第2象限、第3象限となる。

## (2) レビューごとの狙う欠陥を明確化

表2のようにレビューのスキルを考慮し、レビュー観点を割り当て、レビューの狙いを明確化する。

構造の悪さ等難易度の高いものは、ベテランをレビューとし、ロジックの抜け・誤り等難易度の低いものは、中堅や若手をレビューとする。

## 3.2 レビュー観点によるレビュー計画の適正化

### (1) 段階的なレビュー計画

図4のように、コード作成前にレビューを計画し、コード作成序盤、中盤、終盤それぞれのレビューで、アジャイ

表1. レビュー観点の体系化

No.	レビュー観点	難易度	象限
1	構造の悪さ	複雑さなど構造に関する要件に対し、コードの適合を確認する。	高 2, 3
2	想定する不具合	製品不具合事例に対し、原因となりうるコードでの対策を確認する。	高 1
3	自作外ソフトウェアの使い方の誤り	自作外ソフトウェアの不具合事例に対し、原因となりうるコードでの対策を確認する。	高 1, 4
4	設計ポリシー違反	SPL等設計ポリシーに対し、コードの適合を確認する。	中 2
5	仕様との不整合	仕様書とコードの不整合がないことを確認する。	中 1, 4
6	変更影響箇所の漏れ・誤り	ソフトウェア構造を基に変更箇所、変更方法が妥当であることを確認する。	中 1, 2, 3, 4
7	ロジックの抜け・誤り	未初期化などコーディング誤りパターンに合致するものがないことを確認する。	低 1, 4
8	コーディング規約違反	コーディング規約への適合を確認する。	低 3

SPL : Software Product Line

表2. レビュー観点とレビューの対応例

No.	レビュー観点	難易度	レビューア
1	構造の悪さ	高	ベテランA
2	想定する不具合	高	ベテランB ベテランC
3	自作外ソフトウェアの使い方の誤り	高	ベテランC
4	設計ポリシー違反	中	中堅A
5	仕様との不整合	中	中堅A 中堅B
6	変更影響箇所の漏れ・誤り	中	中堅C
7	ロジックの抜け・誤り	低	中堅A 若手A
8	コーディング規約違反	低	中堅B 若手B

ルインスペクション、ウォータースルー、インスペクションとレビュー技法を使い分ける。各段階でのレビュー観点を明確にすることによって、軽微な指摘への偏りを防止できる。

また、表3のように、レビュー観点をレビュー実施時期によって、使い分ける。第1ステップでは、軽微な欠陥を早期に指摘することで、教育効果によって同種の欠陥増加を防止する。第2ステップ以降では“構造の悪さ”等の検出難易度の高い欠陥指摘に集中する。

### (2) レビュー完了基準の設定

対象となるレビュー観点に基づいてコードレビューを実施した際、欠陥割合が基準値を超える場合に、レビュー完了とする。完了基準に未達の場合、同じ観点で再レビューを実施する。レビューごとに表3のように対象となるレビュー観点を定める。

### 3.3 ツール活用によるレビューの効率化

次の用途にツールを活用することによって、コードレビューを効率化できる。

#### (1) レビュー対象の絞り込み

レビュー観点に合致するレビュー対象を絞り込むことで、限られた工数とリソースを用いて効果的なレビューを実施できる。

#### (2) ロジックの抜け・誤りの自動検出

ロジックの抜け(if文のelse抜け等)、誤り(配列外アクセス等)の可能性がある箇所を自動検出することでレビューを効率化できる。

#### (3) レビュー対象の構造の理解

レビュー対象の静的構造に関する情報をレビューに提示することで、レビュー対象を理解しやすくし、目視によるレビューを効率的、効果的に実施可能にする。

(従来)コード作成終盤にレビュー → コード作成序盤から段階的にレビュー

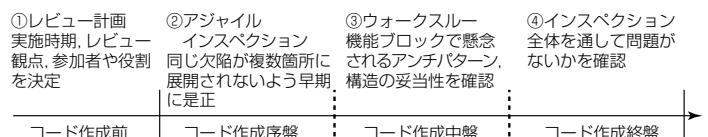


図4. レビュー技法の使い分け

表3. コードレビュー計画例

No.	レビュー観点	難易度	レビュー# 1		レビュー# 2		レビュー# 3	
			実施	レビューア	実施	レビューア	実施	レビューア
1	構造の悪さ	高					○	ベテランA
2	想定する不具合	高					○	ベテランB ベテランC
3	自作外ソフトウェアの使い方の誤り	高					○	ベテランC
4	設計ポリシー違反	中			○	中堅A		
5	仕様との不整合	中			○	中堅A 中堅B		
6	変更影響箇所の漏れ・誤り	中			○	中堅C		
7	ロジックの抜け・誤り	低	○	中堅A 若手A				
8	コーディング規約違反	低	○	中堅B 若手B				

表4. コードレビューに活用可能なツールの例

ツールの用途	ツール名	ツール概要
レビュー対象の絞り込み	CCFinder	クローン検出ツール。ソースコードをトークン単位で直接比較することでコードクローンを検出する。
レビュー対象の絞り込み	Lattix	アーキテクチャ分析ツール。ソフトウェアアーキテクチャの構造分析を行い、サブシステム間の依存関係を簡潔な表形式(マトリックス)で表現する。
レビュー対象の構造の理解	Klocwork	実行バス解析技術の活用によって危険箇所を効率的、網羅的に検出する。
ロジックの抜け・誤りの自動検出		

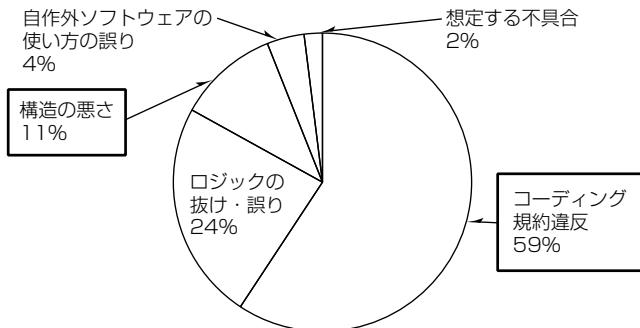


図5. 改善前のコードレビューの欠陥分類

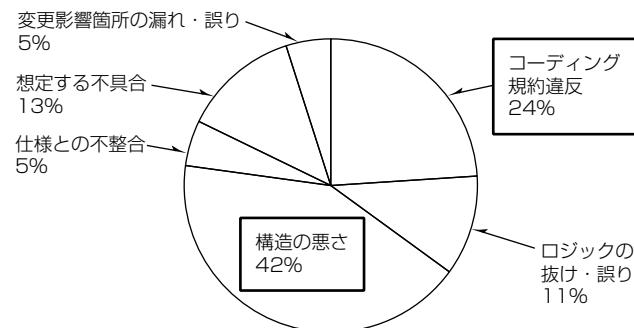


図6. 改善後のコードレビューの欠陥分類

コードレビューに活用可能なツールの例を表4に示す。

#### 4. 効 果

レビュー観点を体系化することによって、レビュー観点の偏りが減少し、本質的な欠陥除去の効果が増加した。開発期間約8か月の通信シミュレータの開発に適用し、図5、図6のように、検出難易度の高い欠陥検出(構造の悪さ)の割合が11%から42%へ増加し、軽微な欠陥検出(コーディング規約違反)の割合が59%から24%へ減少した。

また、コードレビューごとのレビュー観点設定(表5)によるレビュー計画の適正化によって、効率良く品質を確保することができた。図7のとおり、第2ステップで対象としていた構造の悪さとロジックの抜け・誤りの観点について、欠陥割合が30%となった。完了基準(50%)未達となつたため、再レビューを実施したことによって、狙っていた欠陥を検出することができた。

表5. コードレビューごとのレビュー観点例

レビュー	対象レビュー観点
# 1	コーディング規約違反 仕様との不整合
# 2	構造の悪さ ロジックの抜け・誤り
# 3	構造の悪さ ロジックの抜け・誤り
再レビュー	ロジックの抜け・誤り

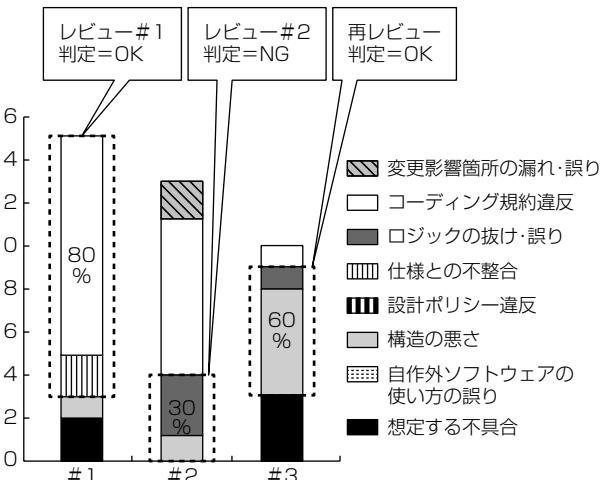


図7. コードレビュー実施結果例

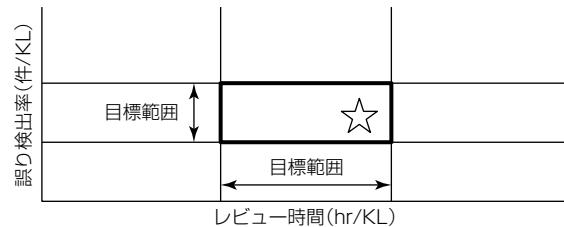


図8. 再レビュー後のゾーン判定

再レビュー後のゾーン判定は図8の星印となり、レビュー時間、誤り検出率ともに目標範囲内となった。

#### 5. む す び

製品開発で、ソフトウェアの高機能化と短期開発が進んでいる。そこで、限られたコストで効果的に欠陥を検出するため、開発したレビュー手法とその効果について述べた。

今後は、レビュー手法を更に進化させ、品質確保の効率化を推進していく。

#### 参 考 文 献

- (1) (独)情報処理推進機構 ソフトウェア・エンジニアリング・センター：組込みソフトウェア開発における品質向上の勧め(コーディング編)，翔泳社，31(2005)
- (2) (独)情報処理推進機構 ソフトウェア・エンジニアリング・センター：定量的品質予測のススメ，オーム社，58(2008)

# 高位LSI設計検証技術とその応用

峯岸孝行\* 元濱 努\*\*  
山本 亮\*\* 坂手寛治\*\*  
宮野鼻晃士\*\*

*High Level LSI Design and Verification Technology and its Application Examples*

Noriyuki Minegishi, Ryo Yamamoto, Koji Miyahana, Tsutomu Motohama, Hiroharu Sakate

## 要旨

製品に必要だが汎用デバイスでは実現できない機能・性能を実現するとき、FPGA(Field Programmable Gate Array)やASIC(Application Specific Integrated Circuit)を用いる。多くの場合、実現すべき機能は複雑なアルゴリズムを起点とし、汎用品で実現できない高い性能が要求される。このような課題に対して、C言語でFPGAやASICを設計する高位LSI設計検証手法が生まれ、適用されている。

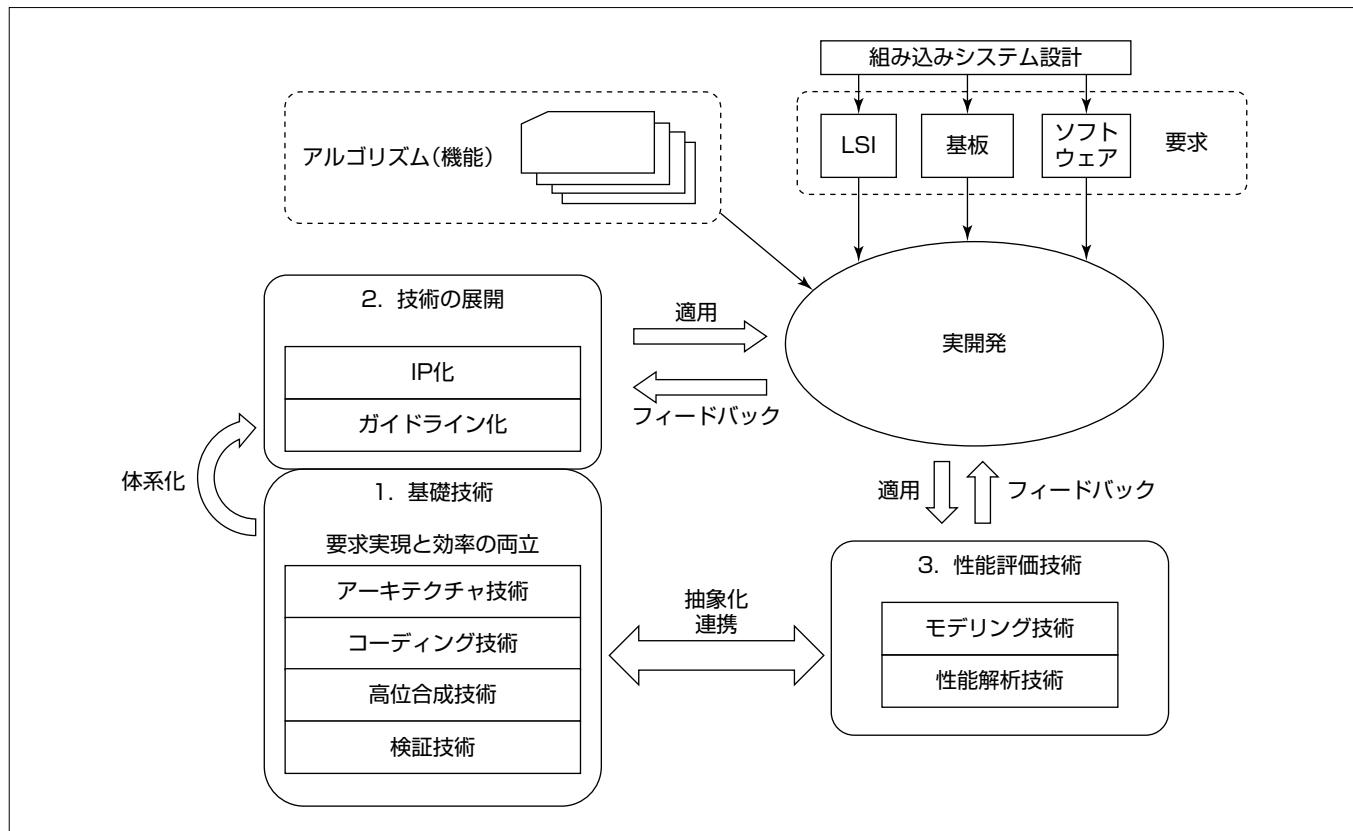
三菱電機では、事業ドメインごとに異なる機能・性能・信頼性・資源制約(コスト・サイズ・電力等)を満たす組み込みシステムをいち早く実現するため、高位LSI設計検証技術の高度化を進めてきた。

基礎技術として、要求実現と効率を両立させる設計方式と記述スタイルを確立した。そして、これをIP

(Intellectual Property)化・ガイドライン化し、FPGAの実開発に適用した。高位LSI設計検証が不得手とされている制御回路を含むFPGAでコード量を従来手法比38%削減、高位LSI設計検証で発生する回路規模増を抑えRTL(Register Transfer Level)と同等の回路規模の実現、そして、アルゴリズムからの短期開発で世界初<sup>(注1)</sup>のスーパーハイビジョンHEVC(High Efficiency Video Coding)符号化装置開発実現などの成果を上げた。さらに、組み込みシステムに拡張適用し、実機試作なしのフロントローディングで電力変換装置の性能達成を実現した。

今後は高位LSI設計検証をアーキテクチャ設計レベルまで拡張し、更に高い要求機能、開発効率化を実現していく。

(注1) 2013年5月9日現在、当社調べ



## 高位LSI設計検証技術の全体像

高位LSI設計検証技術の基礎技術(アーキテクチャ・コーディング・高位合成・検証)を確立し、IP化とガイドライン化によって体系化する。一方、基礎技術を抽象度の高い表現に応用し、設計と連携をとって性能評価技術に展開する。これらの技術を実開発に適用することで成果を得る。また、開発中に発生した課題の解決結果を技術に反映することで技術の改善を図る。

## 1. まえがき

近年、組み込み機器への要求は高くなっています。進歩性のある特長的なアルゴリズムを、性能・品質・信頼性を確保して早期に組み込みシステムとして実現することが必要です。大規模・複雑化する組み込みシステムの要求、特に性能を実現するために、キーデバイスとしてFPGAやASICを開発する場合があります。FPGAやASICの開発効率化手法の一つにC言語を用いた高位LSI設計検証手法があります。当社では、高位LSI設計検証手法を単なる開発効率化にとどまらず、事業実現のための課題の解決に用いている。例えば、従来の7倍の性能を達成する電力変換装置の設計と評価を、FPGAを核とした性能評価モデルによってソフトウェアも含めて評価することで手戻りなしに実現している。

本稿では、要求実現と効率を両立させるための高位LSI設計検証技術とその適用について、事例を交えて述べる。

## 2. 要求実現と効率を両立させる技術

一般に、高位LSI設計検証は演算回路に向いていると言われ、過去の事例も画像処理向けに高位LSI設計検証を適用した例が多い<sup>(1)(2)</sup>。しかし、FPGAやASICの構成要素には、演算を制御する論理、データの入出力回路といった制御回路がある。制御回路は定められたクロック周期に合わせて論理処理を行う必要があるため、設計効率を上げるために抽象度を高くすると性能が出ず、性能を出そうと思うと設計効率が上がらない傾向にある。

今回、制御回路の基本である状態遷移回路に着目し、性能と効率を両立させる記述方法を考案した<sup>(3)</sup>。状態遷移回路を構成する3つの要素を抽出し、3つの要素各々にクロック周期を満たしつつ設計効率を上げる“デザインテンプレート”を用意した。この3つのデザインテンプレートを用いることでどのような状態遷移回路も実現できる。図1に3つの状態遷移要素と、デザインテンプレートを示す。

この3つのデザインテンプレートを用い、FPGAのデータ制御とバスインターフェースの状態遷移回路を設計し、従来手法と比較してコード量を38%（従来77行・提案29行）に削減した。

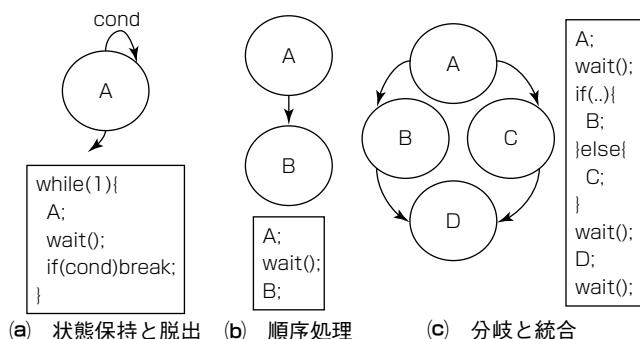


図1. 状態遷移回路のデザインテンプレート

## 3. 技術を展開させるための方策

2章では要求と効率を実現する技術の一部について述べた。これらの技術は当社の研究所のLSI設計技術部門が培ったものである。事業部門や、同じ研究所でもアルゴリズム開発部門が適用するには、技術習得と実設計に応用するためのノウハウ蓄積が必要となる。この章では、高位LSI設計検証技術の適用ハードルを下げるための取組みについて述べる。

### 3.1 IP化による技術の展開

FPGA、ASICの回路構成要素には、似て非なるものが多数存在する。デジタルフィルタはその代表例で、積和演算という基本的な回路構成は同じであるが、タップ数やビット数が異なる。このようなケースでは要求を実現する技術をIP化してしまえばよい。そこで、信号処理アプリケーションで頻出するFIR(Finite Impulse Response)フィルタの小回路規模要求を実現する技術をIP化した<sup>(4)</sup>。フィルタ設計に必要な情報(タップ数、ビット数、四捨五入位置など)をIPにパラメータ設定することで、所望のFIRフィルタを得ることができる。

FIRフィルタを小回路規模で実現するポイントは、DSP(Digital Signal Processing)ブロックと呼ばれる積和演算エレメントの活用である。図2にアルテラ社のDSPブロックの内部構造を示す。DSPブロックには乗算器、加算器、FF(Flip Flop)が組み込まれている。これらの内部リソースに、FIRに必要な演算を最大限マッピングすることで、小回路規模が実現できる。しかし、市販の高位合成ツールは、DSPブロックの構成情報を考慮せずに合成してしまう。

そこで、FIRの回路構成を演算要素に分解し、DSPブロックの演算器を最大限に活用する高位合成方式を考案した。ポイントは演算の間にFFを挿入させないことである。DSPブロックには、入力部と出力部にだけFFが配置され、乗算と加算の間にはFFはない。市販の高位合成ツールではFFを自動挿入するため、乗算と加算の間にFFを挿入してしまう場合がある。これを防ぐため、DSPブロックに

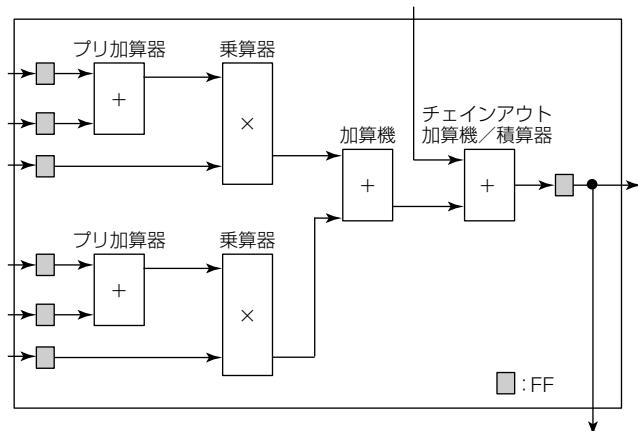


図2. DSPブロックの内部構成例

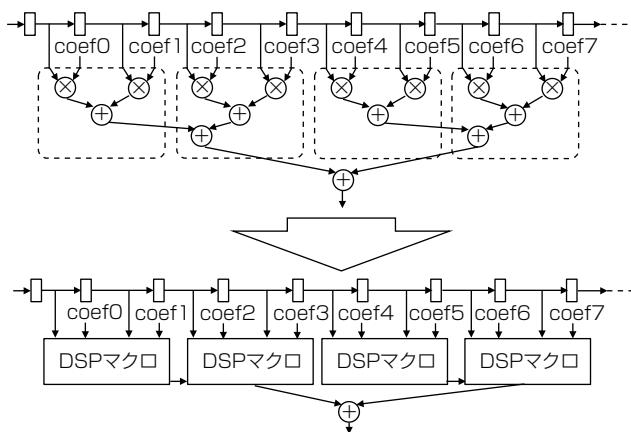


図3. 演算の関数化

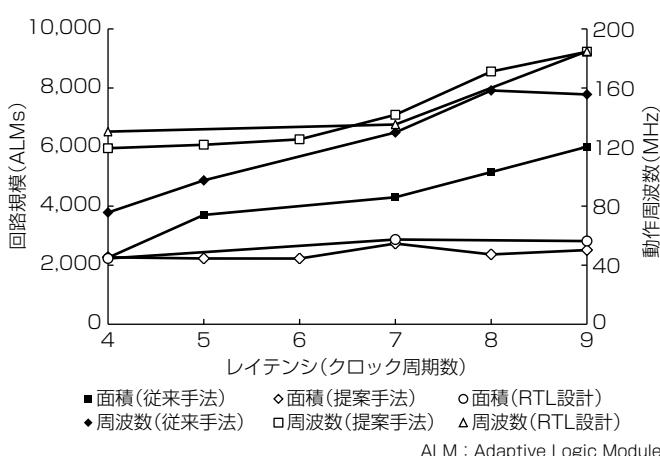


図4. 提案手法と従来手法の比較

割り当てたいFIRフィルタの演算部分を関数化する。図3に演算の関数化の様子を示す。高位合成ツールで関数単位で合成することで、演算の間にFFを挿入することを防ぐ。

この方式を用いて161タップのFIRフィルタに適用した結果を図4に示す。横軸“レイテンシ”は性能指標であり、データ入力から結果が得られるまでのクロック周期数である。提案手法は従来手法と比べ、全てのレイテンシで回路規模が小さく、動作周波数が高く、改善されていることが分かる。また、人手で行うRTL設計とほぼ同等の結果が得られていることが分かる。

### 3.2 ガイドライン化による技術の展開

高位LSI設計検証手法はC言語を源流とする手法であり、C言語でプログラムされたアルゴリズムをFPGAやASICで実現するのに相性が良い。アルゴリズム開発者が高位LSI設計検証技術を用いてFPGAやASICを開発するのが最も効率的であるが、設計品質の担保という課題が発生する。この課題に、設計を“型にはめる”ガイドラインを準備することで対応した<sup>(5)</sup>。

図5に“型にはめる”ガイドラインによる設計を示す。FPGAで実現すべき機能はアルゴリズム開発者によってC言語でプログラムされている。このアルゴリズムを埋め

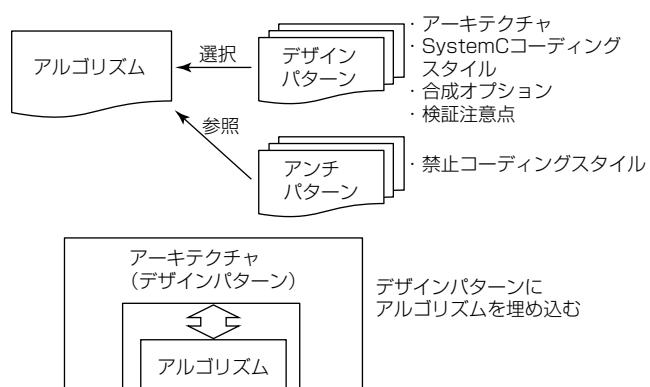


図5. “型にはめる”ガイドラインによる設計

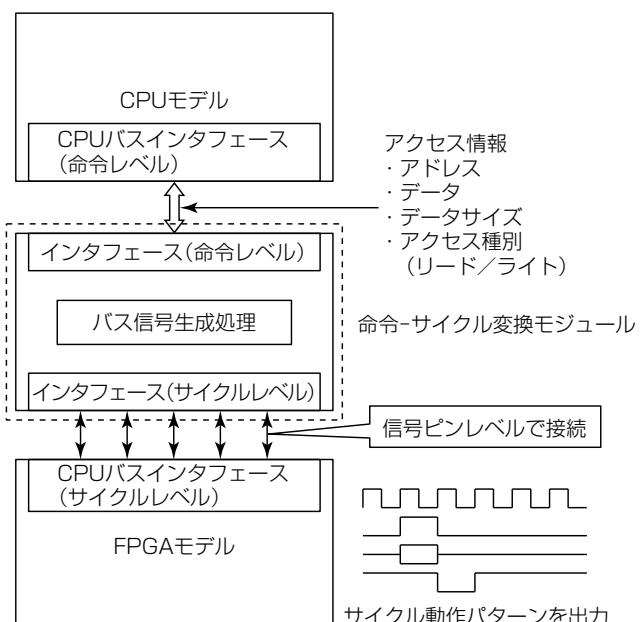
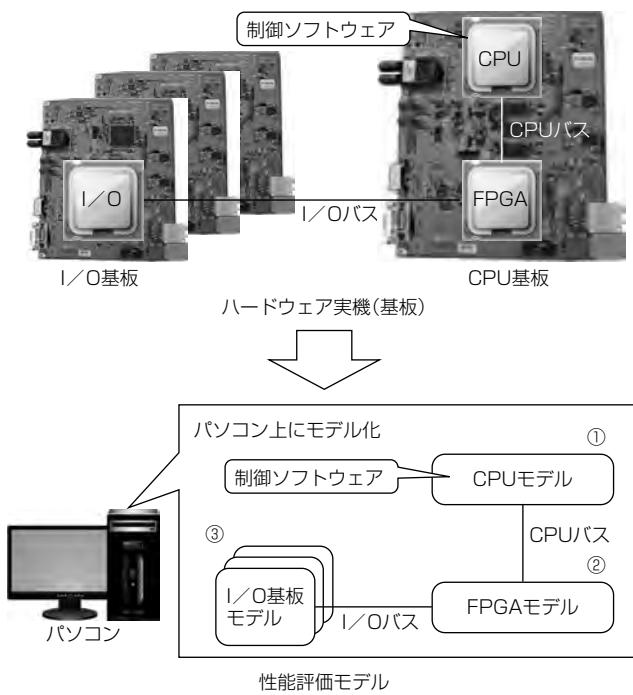
込むようなパターンを用意した。パターンには、FPGA設計で選択する“デザインパターン”と、やってはいけないコーディングスタイルを示した“アンチパターン”がある。アルゴリズム開発者は、演算系、制御系などのタイプからデザインパターンを選択し、アルゴリズムをテンプレートに組み込む形で設計を行うとともに、アンチパターンを参照してアルゴリズム本体に禁止されるコーディングがないことを確認する。

HEVC方式によるSHV(Super High Vision)リアルタイムエンコーダ向けFPGAを、“型にはめる”ガイドライン及び回路規模・動作周波数を事前に確認すること、そして、3.1節で述べたパラメータIP化の手法を適用し、アルゴリズム開発者の手によって開発した。短期間で新しい符号化方式のFPGAを開発することによって、世界初となるスーパーハイビジョンHEVC符号化装置を日本放送協会と共に開発することができた<sup>(6)</sup>。

### 4. 性能評価への取組み

高位言語SystemCは、FPGAやASIC設計だけでなく、組み込みシステム開発に拡張して適用可能である。この章では、近年大規模複雑化している組み込みシステムを高位言語でモデル化し、ソフトウェアを組み合わせた性能評価を実施することで手戻りなく製品開発に適用した事例について述べる。

開発の対象は電力変換装置である。図6に装置の概略図と高位言語を用いた性能評価モデルの対応を示す。CPU基板に搭載するFPGAは周期制御及びバスコントローラの機能を持つ。この装置はI/O基板から入力された電力値をFPGAがCPUに転送し、CPUの演算結果をFPGAがI/O基板に出力する処理を周期的に行う。この開発では従来比7倍の高速化が必要で、ソフトウェアを含めた性能評価が重要であった。システム全体の評価を行うために、電力変換装置を①CPU、②FPGA、③I/O基板(外部入出力を含む)の3つに分割してモデル化した。CPUモデルは、主にモデル化工数削減のためISS(Instruction Set Simulator)



を用い、FPGAとI/O基板はSystemCでモデル化した。

CPUモデルとFPGA・I/O基板モデルでは、シミュレーション上の処理時間単位が異なる。CPUモデルでは命令が単位だが、実際の命令は複数のクロック周期分の時間をかけて実行される。そこでCPUモデルが発行するバスアクセス命令からFPGAモデルのバスインターフェースのサイクル動作への変換を行った。図7に変換の仕組みを示す。CPUモデルが発行するバスアクセス命令から、アドレス、データ、データサイズ、アクセス種別(リード/ライト)といったアクセス情報を抽出し、CPUバスプロトコ

ルに準じたサイクル動作のパターンに変換する。

このようにしてモデル化した電力変換装置で性能評価を実施したところ、性能上の課題が明らかになった。対策として、CPUクロック周波数アップ(基板での改善)、キャッシュミスしないデータ量への削減(ソフトウェアでの改善)、FPGAの処理シーケンス変更(FPGAでの改善)を実施し、フロントローディングによる性能要求達成を実現した。

## 5. む す び

組み込みシステムへの高い要求を、特徴的なアルゴリズムを用いていち早く実現する高位LSI設計検証技術について述べた。当社では、高位LSI設計検証技術を、FPGAやASICの設計検証にとどまらず、組み込みシステムのフロントローディング開発でも適用している。要求実現と効率を両立させる技術をガイドラインとして整備し、初心者にも設計可能な形で展開することで、日本放送協会と共同でスーパーハイビジョンHEVC符号化装置を世界で初めて開発し、電力変換装置のフロントローディング性能評価を実現する成果を得ることができた。

高位LSI設計検証では回路レベルの設計検証が自動化されているものの、アーキテクチャレベルの設計は人間が考える必要があり、ここでは経験やセンスが必要となっている。当社は、高位LSI設計検証をアーキテクチャ設計レベルまで拡張し、更に高い要求実現と開発効率化を目指していく。

## 参 考 文 献

- (1) Schafer, B.C., et al. : Design of Complex Image Processing Systems in ESL, ASP-DAC, 809~814 (2010)
- (2) Wakabayashi, K. : CyberWorkBench : Integrated design environment based on C-based behavior synthesis and verification, VLSI-TSA-DAT 2005, 173~176 (2005)
- (3) Yamamoto, R, el al. : An Efficient Design Approach of Control Logic with the Use of High Level Synthesis for a Video Signal Conversion FPGA, IEEE Design Automation Conference 2012, Poster Session 2U-11 (2012)
- (4) 山本 亮, ほか:高位合成によるFIRフィルタ設計 – 任意のFIRフィルタ回路の自動生成, 電子情報通信学会技術研究報告, 114, No.476, 79~83 (2015)
- (5) 山本 亮, ほか:HEVC方式によるSHVリアルタイムエンコーダ開発への高位合成適用事例, SystemC Japan 2014 (2014)
- (6) 日本放送協会・三菱電機ニュースリリース 2013年5月9日：世界初！スーパーハイビジョン(8K)HEVC符号化装置を開発

# ADC回路におけるアナログ-デジタル回路協調設計技術

大東睦夫\*

Collaborative Design Technology for Analog and Digital Circuits in ADC

Mutsuo Daito

## 要旨

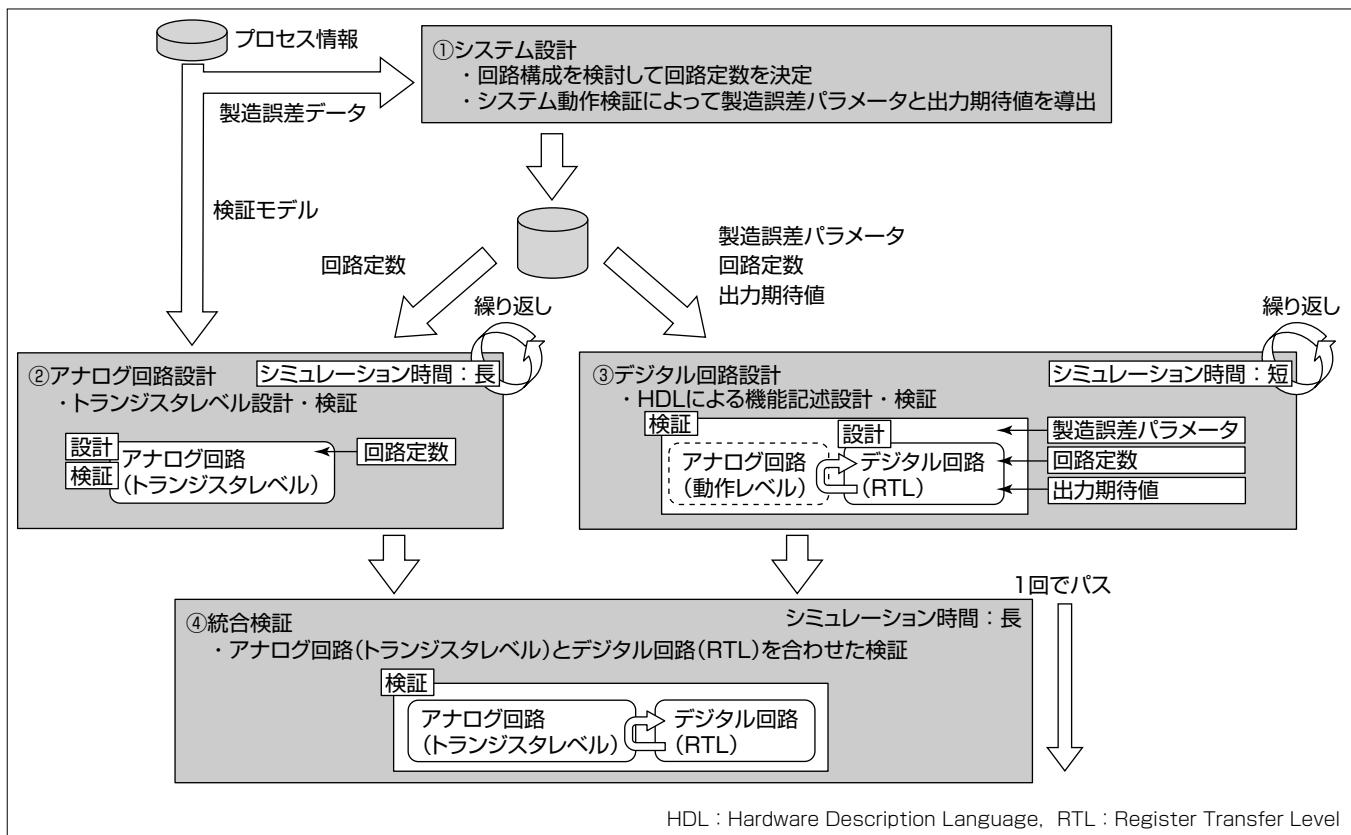
アナログ信号とデジタル信号を扱う回路ではADC(Analog-to-Digital Convertor)を使用しており、製品性能を向上させるためにはADCの高精度化が求められている。高精度化を実現するため、アナログ回路の製造誤差による精度低下をデジタル回路で補正する技術が主流となっている。

アナログ回路とデジタル回路を混載したLSIの設計フローでは、①システム設計後、②アナログ回路設計と③デジタル回路設計を並行して実施し、④統合検証で両回路を合わせて検証する。ファウンドリからは製造誤差データとトランジスタレベルの検証モデルが提供されるが、デジタル回路検証で使用するアナログ回路は機能記述モデルのため製造誤差を考慮した検証ができない。統合検証時には製

造誤差を考慮できるが、この段階で不具合を検出すると手戻りが大きい。

そこで、回路動作を関数で定義しているアナログ回路の機能記述モデルについて、従来は1つにまとめて固定値で扱っていた容量や抵抗値などの回路定数を、個別に扱えるよう実回路の構成に合わせ込む手法を確立した。また、容量や抵抗値などの製造ばらつきや単位容量・面抵抗率などファウンドリから提供される製造誤差データから関数の引数として扱える製造誤差パラメータを算出する仕組みを構築した。

デジタル回路設計時に製造誤差を考慮した検証を可能とし、デジタル補正ADCの開発に適用して、1か月の設計手戻りを抑制した。



## アナログ回路とデジタル回路混載のLSI設計フロー

システム設計段階①でプロセス情報に含まれる製造誤差データから製造誤差パラメータを導出することによってデジタル回路設計③でアナログ回路モデルが正確に動作して検証することができる。統合検証④では接続確認が主となるため、前工程への手戻りを抑制し、設計期間の短縮が可能となる。

## 1. まえがき

アナログ信号とデジタル信号を扱う回路では、デジタル処理部の前段にADCを使用しており、ADCの高精度化は製品性能を実現する上で必要不可欠となっている。ADCの中でもパイプラインADCが汎用のADCとしてよく使用されており、パイプラインADCを高精度化する技術としてデジタル補正技術が開発されている。

本稿では、デジタル補正技術を用いたパイプラインADC(以下“デジタル補正ADC”という。)の開発で、設計期間の短縮化と回路規模の最適化を行うために適用したアナログ-デジタル回路協調設計技術について述べる。

## 2. パイプラインADCのデジタル補正技術

パイプラインADCでは、図1に示すとおり低精度のA/D変換ステージを多段接続し、パイプライン動作させて高精度のA/D変換を行う。各ステージではアナログ入力信号に応じた電荷が容量に充電され、比較器を用いて低精度のA/D変換を行う。その後、容量と増幅器を用いて電荷伝送による信号処理を行い、次のステージへのアナログ信号を生成する。この時、容量に製造誤差があると次ステージへ正確なアナログ信号の伝達ができず、A/D変換結果にも誤差が生じる。パイプラインADCではこの変換誤差を補正する手法として、容量の製造誤差をデジタル的に補正する技術が開発されている。

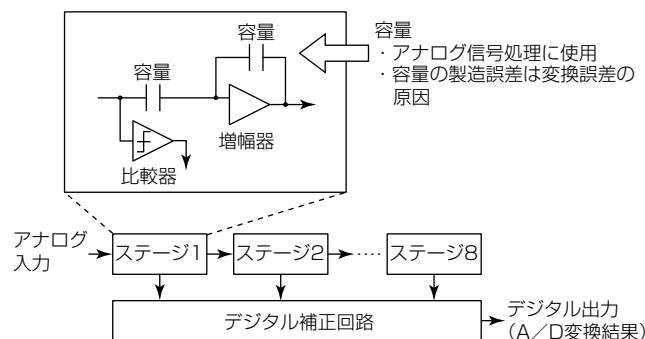


図1. パイプラインADCのブロック図

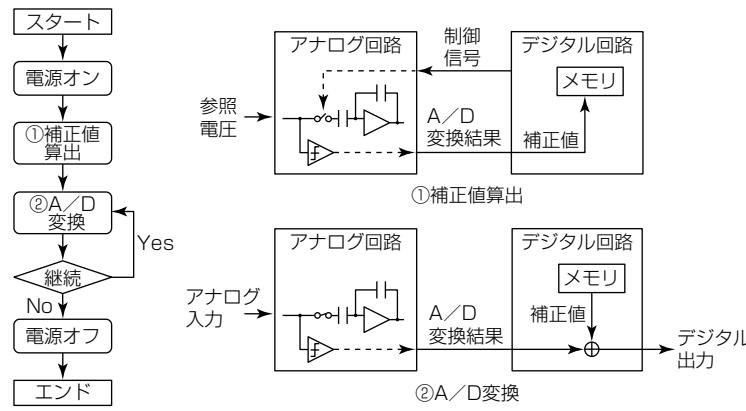


図2. デジタル補正

複数ある補正技術から、今回の開発では図2に示す方法を採用した。電源オン時に参照電圧を用いて容量の製造誤差で生じるA/D変換誤差を測定し、補正值としてメモリに保存する。通常のA/D変換時に補正值を使用し、A/D変換結果に対して補正を行う。

デジタル補正ADCではデジタル回路はアナログ回路を制御して出力された信号を処理しなければならないため、アナログ回路とデジタル回路混載(以下“アナデジ混載”という。)のLSI設計が必要となる。

## 3. アナデジ混載回路設計フロー

### (1) システム設計

システムに要求される機能や性能を実現する回路構成を数式やプログラムを用いて検討し、回路方式や目標性能を設定する。デジタル補正ADCの設計では、システムに要求される変換精度や速度に応じて、回路方式や容量の大きさ、補正を行うステージ数、補正值のビット幅などの回路定数を決定する。

### (2) アナログ回路設計

LSIを製造するファウンドリが提供するデザインキットを使用して、トランジスタや容量等の素子を回路図上に配置し、素子間を配線でつなぐことによってアナログ回路設計を行う。システム設計で決定した回路定数に基づき設計した回路は、SPICE(Simulation Program with Integrated Circuit Emphasis)シミュレータを用いて検証を行い、機能及び性能を確認する。アナログ回路はトランジスタレベルで設計・検証するため、パイプラインADCのような大規模の回路では長時間のシミュレーションが必要となる。

### (3) デジタル回路設計

ハードウェア記述言語(HDL)を用いてシステム設計で決定した回路定数に基づき機能記述を行い、機能及び性能を確認する。その後、論理合成を行い、トランジスタレベルの回路を生成する。そのため、機能検証ではトランジスタレベルの検証を必要とせず、シミュレーション時間はトランジスタレベルでのシミュレーションと比べて1/100以下である。アナデジ混載回路では、デジタル回路設計時には

アナログ回路の動作をモデル化する必要があるが、アナログ回路モデルの正確性を検証できないため、アナログ回路モデルに不具合がある場合、デジタル回路に不具合が混入しても、設計段階では検出することができない。また、デジタル回路の回路規模は回路定数から決定されるため、最適化が難しい。

### (4) 統合検証

アナログ回路とデジタル回路を統合した回路の動作検証を行う。アナログ回路はトランジスタレベルで動作するため、シミュレーション時間は長くなる。今回の開発では、15時間/回のシミュレー

ションで、動作条件を変更して4回実施し、動作確認を行った。そのため、デジタル回路設計で混入した不具合を検出した場合、デジタル回路設計への手戻りが発生し、回路修正後に再度、統合検証で長時間のシミュレーションを行う必要があり、設計期間が長期化してしまう。

#### 4. アナログ-デジタル回路協調設計技術

従来のアナデジ混載回路のLSI設計フローではデジタル回路設計時に製造誤差を考慮した検証ができないため、次のような課題があった。

##### (1) アナログ回路モデルの検証

アナログ回路モデルではファウンドリから提供された検証モデルを扱えないため、デジタル回路に不具合が混入した場合、統合検証でしか発見できず、手戻りによって設計期間が長期化する。

##### (2) 回路規模の最適化

製造誤差を過剰に見込むとデジタル回路の規模が大きくなり、コストの上昇やデジタル回路起因の雑音による性能劣化が起きやすくなる。

これらの課題を解決するために、アナデジ混載回路の設計フローで用いるアナログ-デジタル回路協調設計技術について述べる。

##### 4.1 アナログ回路モデルの検証

ファウンドリからは製造誤差データとトランジスタレベルの検証モデルが提供されるが、デジタル回路検証で使用するアナログ回路は、機能記述モデルのため製造誤差を考慮した検証を実施できない。統合検証時には製造誤差を考慮できるが、不具合を検出すると手戻りが大きい。

そこで、回路動作を関数で定義しているアナログ回路の機能記述モデルについて、従来は1つにまとめて固定値で扱っていた容量や抵抗値などの回路定数を、実回路に合わせて個別に扱えるように変更した。また、容量や抵抗値など誤差のばらつきを測定したデータや単位容量・面抵抗率等のファウンドリから提供される製造誤差データから関数の引数として扱える製造誤差パラメータを算出することで、デジタル回路設計時に製造誤差を考慮した検証を実施可能とした。

今回開発したデジタル補正ADCは差動回路構成であるため、システム設計時に作成するプログラム内の計算式も差動化(差動の両側を個別に計算)し、容量の製造誤差を反映できるよう容量は個別に計算を行う。ファウンドリから提供される容量や抵抗値など誤差のばらつきを測定したデータは統計処理した数値が提供される。図3に示すように、統計処理した数値(c\_dev)を用いて、容量の容量cs[i]をガウシアン関数でばらつかせることによって、容量に関する製造誤差パラメータを算出する。

また、デジタル設計時に作成するアナログ回路の機能記述モデルについても、計算式を差動化し、算出した製造誤

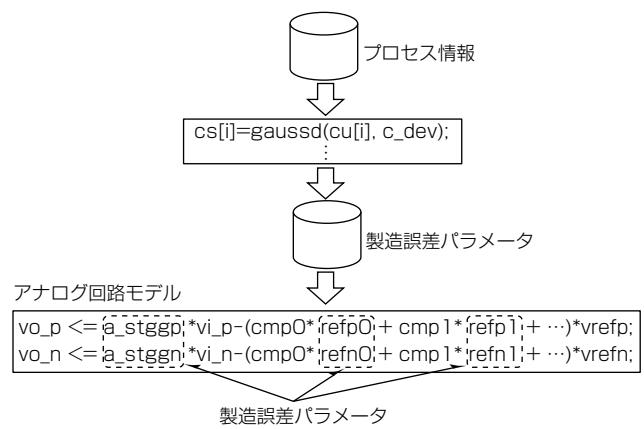


図3. 容量の製造誤差推定

差パラメータを用いて各ステージの入出力関数を記述する。それによって、アナログ回路モデルの動作をトランジスタレベルのアナログ回路の動作に近づけることが可能となる。

さらに、システム設計時に作成したプログラムを用いて出力期待値(デジタル回路のメモリに保存される補正值)を生成し、デジタル回路設計時に得られるシミュレーション結果と比較する。図2に示すように、デジタル補正ADCでは補正值を算出する際にはデジタル回路、アナログ回路、デジタル回路という順番で信号経路が形成される。アナログ回路モデルにはシステム設計時と同じ製造誤差パラメータを用いていることから、最終的にデジタル回路で算出される補正值はシステム設計時に算出された出力期待値と等しくなる。そのため、デジタル回路設計時に補正值を観測することによって、アナログ回路モデルの動作を検証でき、デジタル回路への不具合混入を防止できる。

これらのようなアナログ-デジタル回路協調設計技術を用いることによって、統合検証ではデジタル回路とトランジスタレベルのアナログ回路との接続確認が主となるため、統合検証から回路設計への手戻りを抑制でき、設計期間の短縮化が可能となる。

##### 4.2 回路規模の最適化

製造したLSIには、容量や抵抗、トランジスタに製造誤差が存在する。デジタル補正ADCでは、容量の製造誤差が精度に大きな影響を及ぼす。デジタル回路の規模は容量の製造誤差に依存する補正值で決まり、補正值が大きいとデジタル回路の規模も大きくなる。設計時に過剰に容量の製造誤差を見込むとデジタル回路の規模が大きくなり、コストの上昇やデジタル回路起因の雑音による性能劣化が起きやすくなる。

そこで、システム設計時にプロセス情報に含まれる製造誤差データを用いてプログラム内で容量値をばらつかせて補正值の算出を行い、LSI製造後の補正值の最大値を推定することによって補正值のビット幅を決定し、デジタル回路の規模を最適化する。また、デジタル補正を行うため

のアルゴリズムについても同時に検討を行うことによって、回路規模を更に最適化する。

## 5. デジタル補正ADCの設計

4章で述べたアナログ-デジタル回路協調設計技術を用いてデジタル補正ADCの設計を行った。

### 5.1 システム設計

作成した容量の製造誤差推定プログラムの一部を図4に示す。製造誤差パラメータは、システムに要求される精度

```
for(i = 1; i <= n_stg; i++){
    c_dev = cu[i]*Csig/1000/sqrt(10/Cmim*cu[i]/2);
    for(j = 0; j <= 1; j++){
        c_sum[j][i] = 0;
        for(k = 0; k <= 7; k++){
            cs[j][k][i] = gaussd(cu[i]/2, c_dev);
            c_sum[j][i] += cs[j][k][i];
        }
    }
}
```

図4. 容量の製造誤差推定プログラム

容量のばらつき値	Csig	1.5
最小容量係数	Cmim	1.0
:		

図5. プロセス情報

```
real a_csump = 7.9993274858;
real a_csumn = 8.0030400153;
real a_stgpp = 4.0013075550;
real a_stggn = 3.9996971038;
:
```

図6. 製造誤差パラメータ

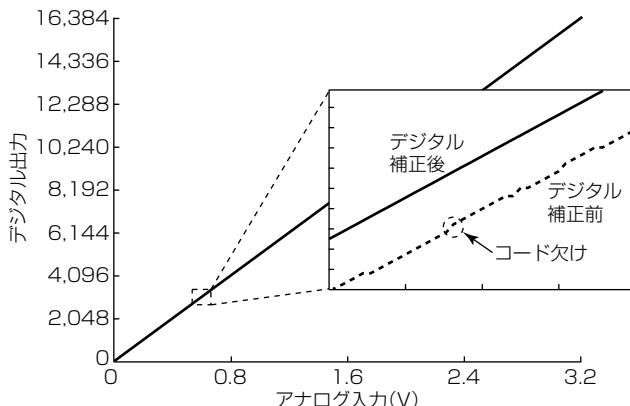


図7. デジタル補正効果の確認

から決定した各ステージの容量cuと、図5に示すようなファウンドリから提供される製造誤差データから計算される容量の製造誤差と容量値との関係を表す値Csig、単位面積当たりの容量Cmim等を用いて算出した。容量値をガウシアン関数によってばらつかせ、補正值の絶対値が最大となるような容量値csから算出される図6に示すような各ステージの入出力関数における係数(例えば、各ステージにおける容量の合計値a\_csump, a\_csumn)を製造誤差パラメータとして導出した。

また、プログラム内で容量値をガウシアン関数によってばらつかせ、1,000回補正值の算出を行い、絶対値が最も大きい値となった補正值が収まるように補正值のビット幅を決定した。また、回路定数、出力期待値の算出も行った。

### 5.2 デジタル回路設計

デジタル回路はハードウェア記述言語であるVerilog HDLで機能記述を行い、アナログ回路モデルは、使用可能な実数(real)型を用いた数値計算で実装した。デジタル回路は、補正值を算出するためにアナログ回路での各ステージのスイッチを制御する制御部と、補正值の算出・保存を行うレジスタ部、アナログ回路出力に対する補正部からなる。5.1節で作成したプログラムから得られる製造誤差パラメータを組み込み、出力期待値とデジタル回路設計時のシミュレーション結果を比較することで、アナログ回路モデルの精度を確認した。

デジタル回路設計におけるシミュレーション結果を図7に示す。デジタル補正を行う前は出力にコード欠けが存在するが、デジタル補正を行った後ではコード欠けがなく直線となっており、補正が正常に行えていることが分かる。

### 5.3 アナログ回路設計と統合検証

5.1節で算出した回路定数を基にしてアナログ回路を設計し、デジタル回路と合わせて統合検証を行った。検証結果に問題がなく、デジタル回路設計に手戻りすることがなかったため、設計期間を1か月短縮することができた。

## 6. むすび

アナログ混載回路のLSI設計フローで用いるアナログ-デジタル回路協調設計技術を開発し、設計期間の短縮を確認した。本稿では、デジタル補正ADCの設計検証技術について述べたが、この技術は様々なアナログ混載回路に展開が可能である。

# 仮想実行環境の構築容易化技術と応用

奥田勝己\* 伊藤真二\*\*  
竹山治彦\* 町田宏隆\*\*\*  
三輪昌伸\*

Automated Generation of Virtual Execution Environments and its Applications

Katsumi Okuda, Haruhiko Takeyama, Masanobu Miwa, Shinji Ito, Hirotaka Machida

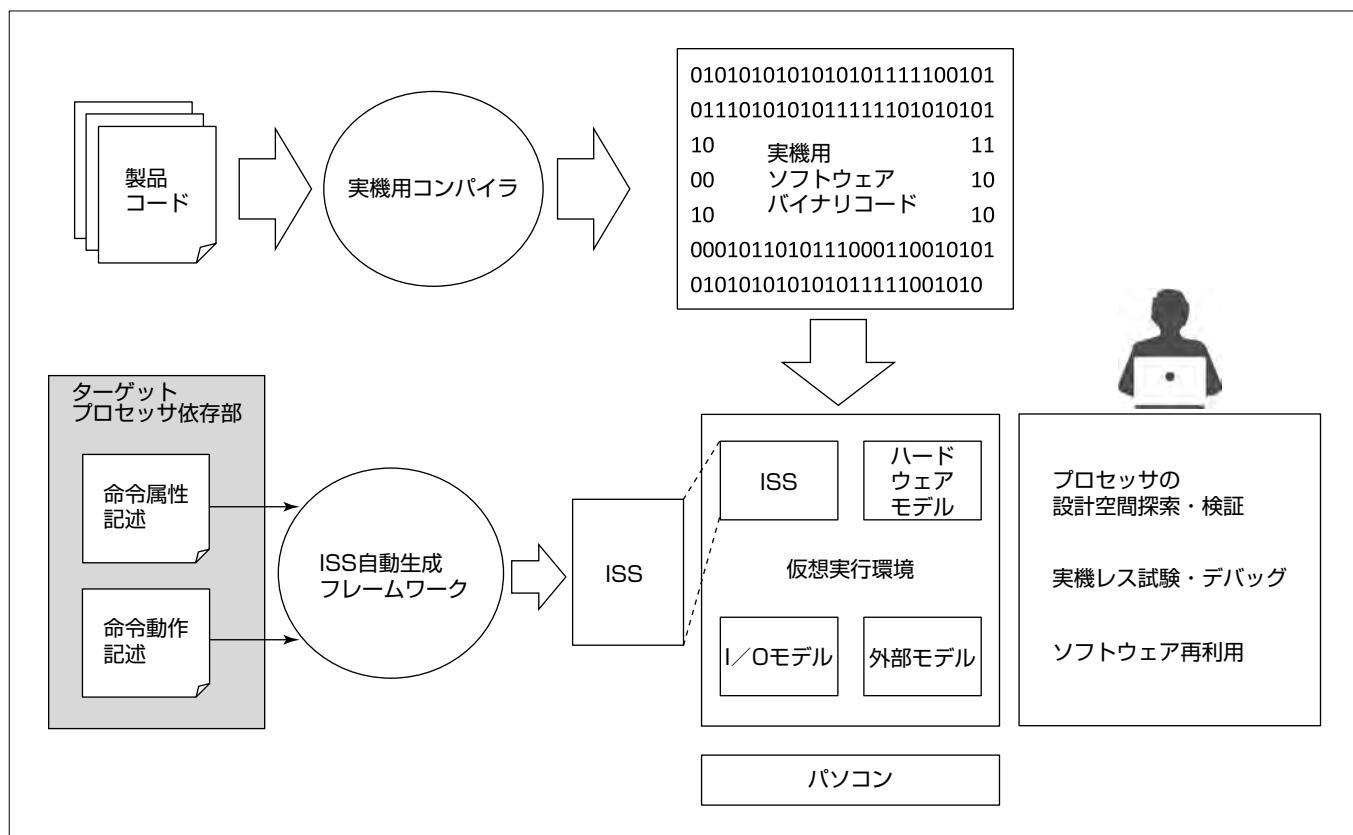
## 要旨

命令セットシミュレータ(Instruction Set Simulator : ISS)は、ターゲットプロセッサの動作を模擬するソフトウェアである。ISSは組み込みシステムの開発で多目的に利用が可能であり、重要な開発ツールの1つである。例えば、プロセッサ開発者はプロセッサの設計空間探索・検証にISSを利用する。また、ソフトウェア開発者は、ISSを組み込んだ仮想実行環境を用いることで、実機の開発と並行してソフトウェアの試験・デバッグを行う。

これらの用途ではISSが組み込みシステムの開発早期から利用可能であることが要求されるため、ISS自体を効率的に開発することが課題である。そこで、三菱電機ではISS自動生成フレームワークを開発することでISSの構築を効率化した。ISS自動生成フレームワークは、ターゲッ

トプロセッサ依存の命令属性記述と命令動作記述を主な入力とし、高速なISSを生成することができる。

ISS自動生成フレームワークで構築したISSの代表的な応用事例として、内製プロセッサの検証への適用と人工衛星運用手順検証用衛星シミュレータへの適用の2つが挙げられる。内製プロセッサの検証では、生成されたISSを期待値モデルとして利用している。また、衛星シミュレータへの適用では、生成された高速ISSをシミュレータに搭載することで実衛星のソフトウェアバイナリコードを再利用し、シミュレータ上で直接実行する。このため、衛星の高精度なシミュレーションを行うことができ、運用手順の確実な検証を実現している。



## ISSの自動生成と利用

ISSを組み込んだ仮想実行環境は、実機用ソフトウェアバイナリコードを実機レスで実行することができる。組み込みシステム開発では仮想実行環境を用いることで、プロセッサの設計空間探索・検証、ソフトウェアの実機レス試験・デバッグなどを行うことができる。開発したISS自動生成フレームワークでは、ターゲットプロセッサごとの命令属性記述と命令動作記述からISSを自動生成することができる。

## 1. まえがき

プロセッサの動作を模擬するISSは、組み込みシステムの開発で多用途に適用可能な重要なツールである。例えば、プロセッサ開発者はプロセッサの設計空間探索・検証にISSを利用する。また、組み込みソフトウェアの開発者はISSを組み込んだ仮想実行環境を用いることで、実機レスでソフトウェアの試験・デバッグを行うことができる。特に仮想実行環境を用いた試験では、実機での実現困難な故障注入試験も容易に行うことができる。さらに、ISSを製品に組み込むことで、ソフトウェアの再利用が容易となる。プロセッサAの製品にプロセッサBを模擬するISSを組み込むことで、プロセッサB用のソフトウェアを移植することなく再利用することができる。

組み込みシステム開発へのISS適用における課題は、ISSの生産性と実行速度である。組み込みシステムで使用されるプロセッサの種類は、市販品・内製品を含めて多種多様である。製品や機種ごとにプロセッサが異なる組み込みシステムで広くISSを用いるためには、ISSの効率的な構築手法が必要である。また、ISSはシミュレーション速度が高速であるほど適用範囲が広い。例えば、ISSを用いてソフトウェアを再利用するためには、ISSが十分に高速である必要がある。

ターゲットが市販プロセッサの場合、サードパーティ製のISSやQEMU<sup>(1)</sup>などのOSS(Open Source Software)の利用も考えられる。しかし、それらのISSではプロセッサの種類ごとに調達性が異なることや、性能・品質面でばらつきがあることが問題となる。

そこで当社では、ISS自動生成フレームワーク(以下“ISS生成フレームワーク”という。)を開発し、高性能なISSを効率的に構築できるようにした。ISS生成フレームワークは、プロセッサ記述を入力とし、インタプリタ方式及び動的バイナリ変換方式のISSを構成する主要なコードを自動生成することで、高速・高品質なISSを出力することができる。

本稿では、ISSの実行方式としてインタプリタ方式と動的バイナリ変換方式を述べ、両方式のISS自動生成に対応したISS生成フレームワークについて述べる。また、ISS生成フレームワークの適用事例についても述べる。

## 2. ISSの実行方式

### 2.1 インタプリタ方式

古典的なISSの実行方式はインタプリタ方式である。インタプリタ方式のISSがターゲットプログラムを実行する処理フローを図1に示す。ISSは、プログラムの終了条件が成立するまで1命令ずつフェッチ、デコード、実行を繰り返すループ処理として構成される。命令フェッチステップ

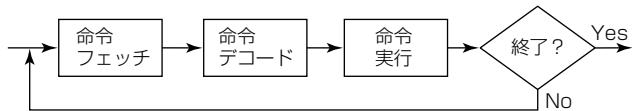


図1. インタプリタ方式の処理フロー

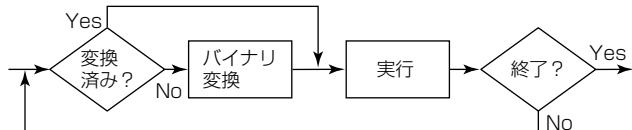


図2. バイナリ変換方式の処理フロー

ではプログラムカウンタが示すアドレスから命令語を読み出す。命令デコードステップでは、フェッチした命令の種類を特定し、命令オペランドを示す命令フィールドを抽出する。命令実行ステップでは、命令デコードステップで特定した命令種類ごとの処理を、抽出した命令フィールドの値をパラメータとして実行する。

インタプリタ方式のISSは、プロセッサの機能面での動作を素直に表したモデルであり、記述が比較的容易である。しかし、ターゲットの1命令を数十から数百のホスト命令で模擬実行するため、処理速度は低速である。

### 2.2 動的バイナリ変換方式

動的バイナリ変換方式(以下“バイナリ変換方式”という。)のISSは、ターゲットプログラムの実行時に命令列をホスト命令列に変換してから実行する。変換で得られたホスト命令列は、後で再利用が可能のように、変換元ターゲット命令のアドレスと対応する変換キッシュと呼ばれるメモリ領域に保存される。バイナリ変換方式のISSは、再度同じターゲット命令のアドレスを実行する場合、変換キッシュから変換済みのホスト命令列を取得し、変換済み命令を実行する(図2)。

バイナリ変換方式では、ターゲット命令の実行ごとに命令フェッチとデコードを行う必要がないため、インタプリタ方式と比較して高速にターゲットプログラムを実行できる。

## 3. ISS生成フレームワーク

ISS生成フレームワーク(図3)は、ターゲットプロセッサ依存部の命令属性記述と命令動作記述を主な入力とし、ISSを出力する。生成対象となるISSは、インタプリタ方式及びバイナリ変換方式のISSである。入力の命令属性記述は命令ごとのフォーマットや分岐命令か否かなどの属性を記述したファイルである。一方、命令動作記述は命令ごとのふるまいを記述したファイルである。

生成されるISSの主な構成要素は命令デコーダ、インタプリタ、動的バイナリトランスレータである。命令デコーダは、インタプリタ方式及びバイナリ変換方式のISSで共通のモジュールである。また、インタプリタとバイナリトランスレータはそれぞれインタプリタ方式ISSとバイナリ

変換方式ISSのモジュールである。

### 3.1 命令デコーダの自動生成

一般に命令のデコード処理は、デコーディングツリーで表現可能である。デコーディングツリーの各辺はビットパターンをラベルとする。また、リーフは命令の一意な識別子をラベルとする。

命令デコーダは、デコーディングツリーのルートから開始し、リーフに到達するまで子ノードを巡回することで、命令語をデコードする。次の巡回対象ノードは、命令語と一致するビットパターンをラベルとする辺でつながった子ノードである。命令デコーダの性能は使用するデコーディングツリーの深さに依存する。すなわち、ツリーが浅いほど命令デコーダは高速である。

当社では、図4の模式図のように命令属性記述に含まれる命令フォーマットを入力とし、デコーディングツリーを生成するアルゴリズムを開発した<sup>(2)</sup>。このアルゴリズムでは、変則的な命令セットにも対応可能であり、従来アルゴリズム<sup>(3)</sup>との比較で22%から40%の深いツリーを生成することが可能である。

### 3.2 インタプリタの自動生成

ISS生成フレームワークでは、命令動作記述中の処理を補完するコードを生成することで、人手によるISSの記述量を削減する。ISS生成フレームワーク適用前後の記述の比較を図5に示す。

ISS生成フレームワークを用いない場合、命令語から命令ワードを抽出する処理①や、プログラムカウンタを更新する処理②の記述が必要になる。一方、ISS生成フレームワークを用いた場合、処理①と処理②のコードは自動生成されるため、記述が不要である。ISS生成フレームワークは、参考文献(4)に記載されている手法でISSを自動生成する。オープンソースのGDB(Gnu DeBugger)との比較では、40%少ない記述量でARM<sup>(注1)</sup>用のISSを生成できることを確認した。

(注1) ARMは、ARM Ltd. の登録商標である。

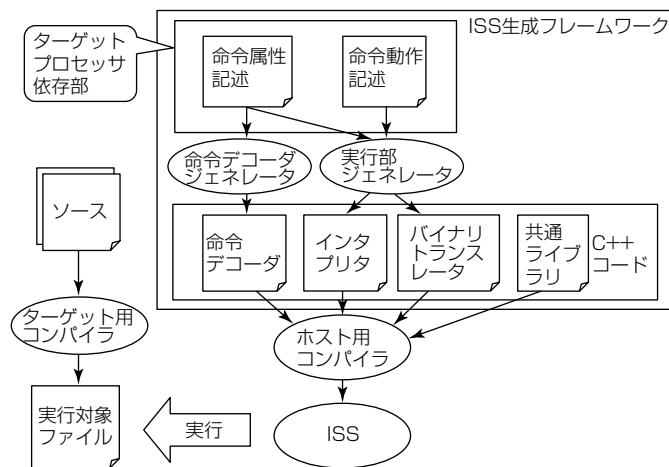


図3. ISS生成フレームワーク

### 3.3 バイナリトランスレータの自動生成

一般にバイナリ変換方式ISSは複雑で開発に労力を必要とする。バイナリ変換時における1命令変換の様子を図6に示す。ISSがターゲット命令daddをホスト命令列に変換する場合、daddに対応するホスト命令列の変換テンプレートに対して、変換時に確定する値をパラメータに代入することで目的のホスト命令列を得る。しかし、ISS開発者が変換テンプレートを作成するためには、ターゲットの命令セットだけでなくホストの命令セットも習得する必要がある。

ISS生成フレームワークでは、ISS開発者がホスト命令を直接操作することなくバイナリトランスレータを記述可能としている<sup>(5)</sup>。ISS生成フレームワークは、インタプリタ用に記述した命令動作記述をホストネイティブコンパイラでオブジェクトファイルにコンパイルし、当該オブ

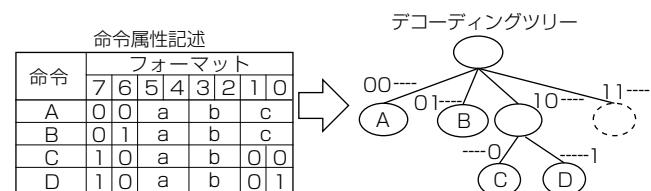


図4. 命令デコード処理のデコーディングツリー

従来の命令動作記述	ISS生成フレームワークでの命令動作記述
<pre>void dadd(int i) {     int rs = (i &gt;&gt; 21) &amp; 31;    处理①     int rt = (i &gt;&gt; 16) &amp; 31;   处理①     int rd = (i &gt;&gt; 11) &amp; 31;  处理②     GPR[rd]=GPR[rs]+GPR[rt];     PC = PC + 4;            处理② }</pre>	<pre>DEFINST(DADD) {     GPR[rd]=GPR[rs]+GPR[rt]; }</pre>

図5. 命令動作記述の比較

変換テンプレート（ホスト命令列）	
■: パラメータ部分	<pre> ba 00 00 00 00      mov \$0x0,%edx b8 00 00 00 00      mov \$0x0,%eax 48 8b 04 c5 00 00 00  mov 0x0(%rax,8),%rax 48 03 04 d5 00 00 00  add 0x0(%rdx,8),%rax ba 00 00 00 00      mov \$0x0,%edx 48 89 04 d5 00 00 00  mov %rax,0x0(%rdx,8) </pre>
変換元のターゲット命令	
<pre>dadd t3, t1, t2 : GPR[2] = GPR[0] + GPR[1]</pre>	
汎用レジスタ @0x80000000	<pre> ba 00 00 00 00      mov \$0x8,%edx b8 00 00 00 00      mov \$0x9,%eax 48 8b 04 c5 00 00 00  mov 0x40000000(%rax,8),%rax 48 03 04 d5 00 00 00  add 0x40000000(%rdx,8),%rax ba 00 00 00 00      mov \$0xa,%edx 48 89 04 d5 00 00 00  mov %rax,0x0(%rdx,8) </pre>
変換先ホスト命令列	

図6. バイナリ変換の例

<pre>DEFINST(DADD) {     GPR[rd]=GPR[rs]+GPR[rt]; }</pre>
↓
コンパイルと変換テンプレート抽出
<pre> ba 00 00 00 00      mov \$0x0,%edx b8 00 00 00 00      mov \$0x0,%eax 48 8b 04 c5 00 00 00  mov 0x0(%rax,8),%rax 48 03 04 d5 00 00 00  add 0x0(%rdx,8),%rax ba 00 00 00 00      mov \$0x0,%edx 48 89 04 d5 00 00 00  mov %rax,0x0(%rdx,8) </pre>

図7. 変換テンプレートの抽出

ジェクトファイルから変換テンプレートを抽出することで、バイナリトランスレータを自動生成する(図7)。ARM, SH<sup>(注2)</sup>, MIPS64<sup>(注3)</sup>をターゲットにベンチマークプログラムDhrystoneとCHStoneを用いた実験評価では、ベースとなるインタプリタに対して1.4倍から13.4倍の範囲で高速化することを確認した。

(注2) SHは、ルネサスエレクトロニクス(株)の登録商標である。  
(注3) MIPS64は、Imagination Technologies LLCの登録商標である。

#### 4. 適用事例

##### 4.1 内製プロセッサの検証

FA(Factory Automation)用コントローラでは、処理性能が重要なポイントである。そこで、当社では、FA用の演算に特化した専用プロセッサを開発し、製品に搭載している。プロセッサの品質は製品全体の品質に大きな影響を与えるため、プロセッサの検証は特に重要である。このため、プロセッサの検証では大量の検証パターンをプロセッサに入力し、正しい出力が得られるか否かの確認が行われる。検証パターン数は膨大であるため、プロセッサの出力が正か否かを人手で判定することは困難である。そこで、ISSを期待値モデルとして利用することで、効率的に検証を進めることができ可能となる。図8はISSを期待値モデルとして用いたプロセッサ検証を示している。検証パターンを実プロセッサとISSに入力し、実プロセッサの出力とISSが出力した期待値を一致比較することで、検証ケースの合否を判定する。

プロセッサ検証時にISSを期待値モデルとして利用するためには、実プロセッサの開発と並行してISSを開発する必要がある。そこで、ISS生成フレームワークを適用することでISSを短期開発し、プロセッサ検証時にISSを利用可能としている。

##### 4.2 人工衛星運用の事前検証

人工衛星は、運用ミス等で一時的に可用時間が失われるだけでも大きな損失となることから、送信するコマンドシーケンスや搭載ソフトウェアバイナリコードのパッチ修正などを、衛星シミュレータで事前に検証することが一般的である。当社でも、人工衛星とともに衛星シミュレータ(図9)を提供しているが、衛星シミュレータには衛星搭載ソフトウェアが組み込まれており、ISSは衛星シミュレータを実現するためのコア技術の1つとなっている。

この衛星シミュレータでは、用途が事前検証であることから高速シミュレーションの要求は強い。また、衛星によっては15年以上の長期にわたって運用を行うことから、衛星シミュレータのプラットフォーム(OSやプロセッサ)変更に対応した長期サポート・ポーティングも要求される。一方、衛星搭載プロセッサやそのISSはニッチであることもあり、これらの要求に応えるISSを外部ベンダーに期待することは難しい。

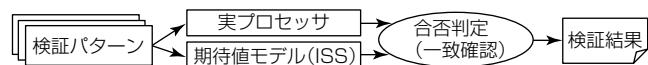


図8. ISSを期待値モデルとしたプロセッサ検証

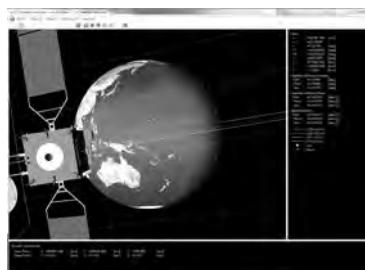


図9. 人工衛星シミュレータの3D確認画面

ISS生成フレームワークはこれら の要求に応えるものであり、バイナリ変換方式のISS生成によって衛星シミュレータの

シミュレーション速度はリアルタイム比10倍以上を実現、また、自社でISSを生成することによって外部ベンダーに依存しない長期サポート・ポーティングを可能としている。

#### 5. むすび

ISSは、プロセッサの設計・検証、組み込みソフトウェア開発、ソフトウェア再利用などに不可欠なツールである。組み込みシステム用途ではプロセッサの種類が多種多様であるため、ISSの効率的な開発が課題である。

本稿では、ISSを効率的に構築するためのISS生成フレームワークとその適用事例について述べた。ISS生成フレームワークは、ターゲットプロセッサごとに用意した記述からISSの主要構成要素である命令デコーダ、インタプリタ、バイナリトランスレータを自動生成可能である。

#### 参考文献

- (1) Bellard, F.: QEMU, A Fast and Portable Dynamic Translator, 2005 USENIX Annual Technical Conference, 41~46 (2005)
- (2) Okuda, K., et al.: Decision tree generation for decoding irregular instructions, 2016 Design, Automation & Test in Europe Conference & Exhibition(DATE), 1592~1597 (2016)
- (3) Fournel, N., et al.: Automated generation of efficient instruction decoders for instruction set simulators, 2013 IEEE/ACM International Conference on Computer-Aided Design(ICCAD), 739~746 (2013)
- (4) 奥田勝己, ほか: 命令セットシミュレータ生成フレームワークの設計と実装, 情報処理学会論文誌, 57, No. 8, 1718~1736 (2016)
- (5) Okuda, K., et al.: Automated Generation of Dynamic Binary Translators for Instruction Set Simulation, 2017 22st Asia and South Pacific Design Automation Conference(ASP-DAC), to appear (2017)

# 幅広い分野の製品を支えるEMC技術

EMC Technologies for Supporting Wide Range of Products

Yuichi Sasaki, Hideyuki Ohashi, Naoto Oka, Chiharu Miyazaki

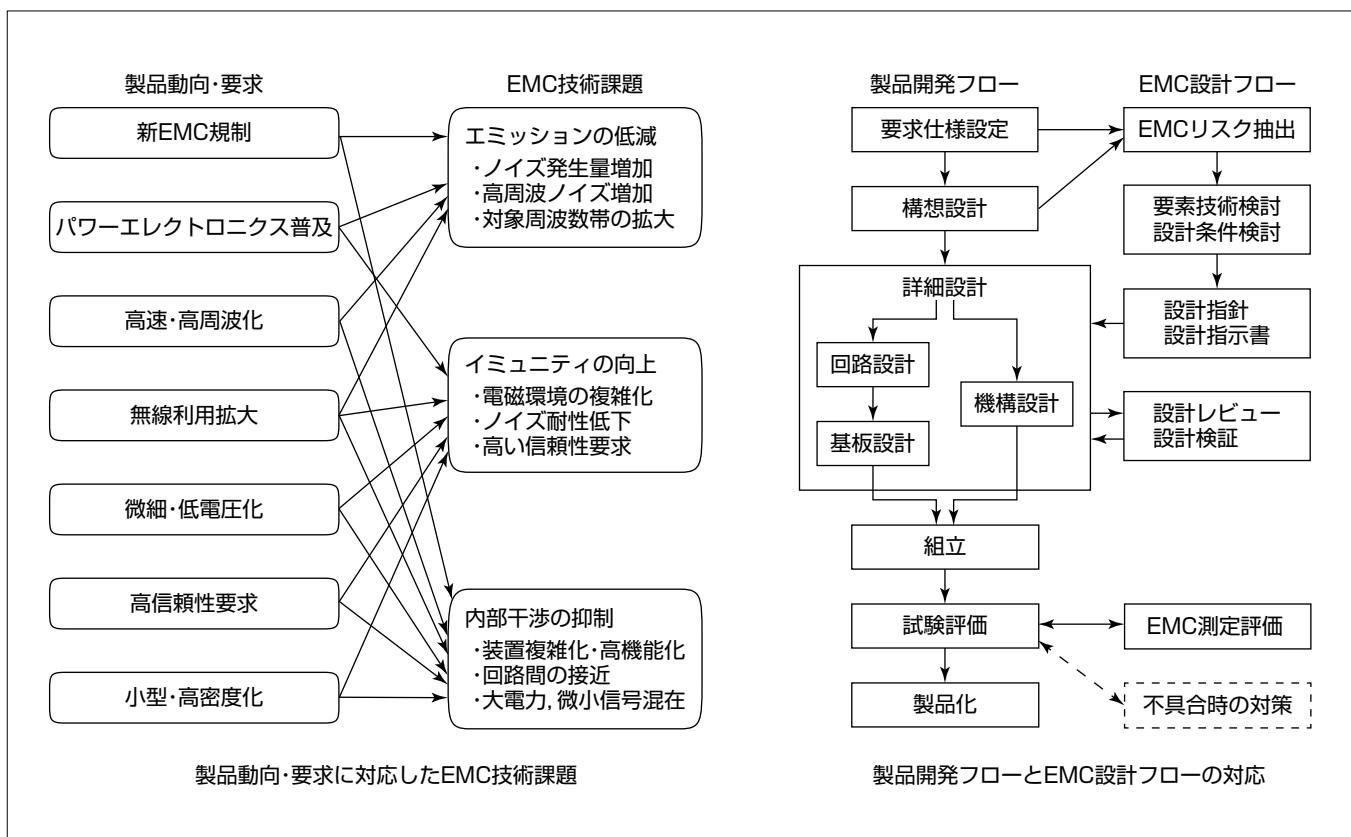
佐々木雄一\* 宮崎千春\*\*  
大橋英征\*  
岡 尚人\*\*

## 要旨

EMC(Electromagnetic Compatibility: 電磁両立性)は、他機器に影響を与える不要電磁ノイズの漏えい(エミッション)を抑え、かつ、外来電磁ノイズで誤動作を起さない(イミュニティ)、電気・電子機器に必要な基本性能である。さらに、機器内部の電磁干渉による誤動作防止もEMCに含まれる。三菱電機では、家電から宇宙機器まで、構成や大きさ、電力レベル、使用環境、適用EMC規格などが大きく異なる幅広い分野の製品を扱っており、それぞれに対してEMC性能を確保した製品開発を行っている。一方、製品の高性能化や小型・高密度実装化に伴う、発生ノイズのレベル増加や高周波化、外来ノイズ耐性の低下、機器内電磁干渉の増加など、EMC問題の顕在化はとどま

らない。今後、IoT(Internet of Things)の普及によって、様々なセンサや機器がネットワークで接続されたシステムが構築される状況では、より複雑な電磁環境下で、高い信頼性が必要となり、EMCに関する課題は更に増加すると予想される。

このように複雑化するEMC課題に対し、EMC品質を確保する製品開発を効率的に進めるためには、製品開発の初期段階からEMC品質を作り込むEMC設計が重要である。新たな課題に対応するEMC要素技術の開発を継続的に行うとともに、これに基づいたEMC設計フローの構築、設計結果の検証や製品のEMC品質評価を行うためのEMC測定評価技術の開発も推進し、当社の成長戦略を支えている。



## 製品動向や要求に対応したEMC技術課題とEMC設計フロー

製品の動向やEMC規制などの要求によって、EMCに対する技術課題が顕在化している(左)。これに対して、EMC品質を確保しつつ製品開発の効率化や製品コストの低減を図るために、開発初期段階からEMC品質を作り込むEMC設計フローの構築、課題に対応するEMC要素技術及び設計の検証や品質評価を行うためのEMC測定評価技術の開発を行っている(右)。

## 1. まえがき

EMC技術は、製品の設計段階から試験評価まで、開発製造の全工程で適用される。

本稿では、設計段階でのEMC技術を主な対象とし、実機適用のための要素技術及び製品へのEMC技術適用例について述べる。

## 2. 製品開発とEMC技術

EMC品質を確保する製品開発を効率的に進めるためには、開発の初期段階からEMC品質を作り込むEMC設計が重要である。この章では、まず、EMC設計技術として、EMCの基本要素及び製品開発の各工程でのEMC検討内容について述べる。次に、EMC設計に向けた要素技術及びEMC測定評価技術について述べる。

### 2.1 EMC設計技術

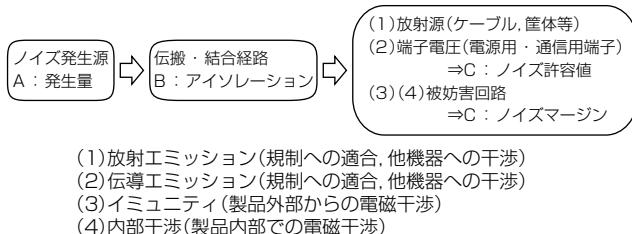
製品で発生するEMC問題は、図1に示す3つの基本要素(ノイズ発生源、伝搬・結合経路、放射源等)で表される。

図中の(1)、(2)は、放射・伝導エミッションに対応する。ノイズ源となる回路や部品から発生したノイズが伝搬し、放射の場合はケーブルや筐体(きょうたい)などの金属導体がアンテナとして作用して空間へ放射され、伝導の場合は電源線や通信線にノイズが誘導される。

図中の(3)は、放射・伝導イミュニティに対応する。外部から侵入したノイズが製品内の被妨害回路に伝搬し、誤作動や性能劣化等を発生させる。図中の(4)は、製品内部の電磁干渉であり、ノイズ源、伝搬経路、被妨害回路が機器内に存在する。

製品のEMC設計については、設計初期段階で製品化時に予想されるEMCリスクをこの基本要素に対応させて抽出し、対処方法の検討を行う。

次に、製品開発の各工程でのEMC検討内容について述べる(要旨の図(右))。構想設計では製品仕様、規格、製品構成、回路構成、構造条件からEMCリスクを検討してEMC設計方針を立てる。適用規格、低ノイズ化、送/受信回路への電磁干渉、高分解能ADC(Analog-to-Digital Converter)や高出力パワーエレクトロニクス回路搭載、寸法と質量制限などの課題に対してEMC技術での設計方針を立てる。EMC基本3要素によって、シールドする部分、



EMC抑制設計の実現  
A(発生量) - B(アイソレーション) < C(ノイズ許容値、ノイズマージン)

図1. EMC基本3要素

フィルタを搭載する部分、レイアウトを検討する回路や部品などを明確にする。また、シミュレーションや実験で判断することを決め、過去の不具合の確認や設計変更点などへの対応も行う。構想設計での検討結果をこれら以降の詳細設計(機構・回路・基板設計)で具体化する。この中には、コンデンサ実装など汎用性のある設計ルールで対応できる内容と、制約条件下での配線レイアウトのように方針や考え方をベースに製品の特徴に合わせて実装を具体化していく内容があり、ともに不可欠となっている。

機構設計では、筐体内の基板やモジュールなどの配置、ケーブル配置、ケーブルやコネクタ選定、導電性筐体と基板グラウンドの接続方法などの検討がある。回路設計では、ノイズ発生源となる部品のリスク抽出、ノイズ源と想定される信号線や電源線に対するフィルタ回路の導入や回路定数の検討がある。ノイズに弱い微弱信号線を扱う回路の場合はシールド構造の検討などもある。基板設計では、回路設計でリスク抽出した課題の対応策を実現するため、基板層構成、フィルタ部品配置と配線設計、部品レイアウト、コネクタのグラウンド接続方法、筐体グラウンドと基板グラウンドの接続箇所などの検討がある。これらの取組みを設計初期段階から進め、製品のEMC品質の実現と設計開発の効率化を図る。

### 2.2 EMC設計に向けた要素技術

進化する製品や実装技術に対応するため、適用するEMC要素技術の継続した進化が要求される。ここでは3件の事例について述べる。

差動デジタル信号伝送の基板実装では、コモンモード成分を抑える設計がノイズ放射低減と耐ノイズ性向上のために必要である。しかし、高密度な配線設計が求められるプリント基板の設計では、線長、線幅、線間距離、基材の誘電率、線大地間距離といった基本条件以外の構造条件が要求される。よって、このための設計指針をシミュレーションと実験検証によって作成している。図2に基板グラウンド端部に高速差動信号線を配線する場合の検討例を示す。コモンモード成分の発生量と基板端部までの距離の関係から、この距離の許容量を実装条件として得る<sup>(1)</sup>。

各種コンポーネント間を接続するシールドケーブルでは、

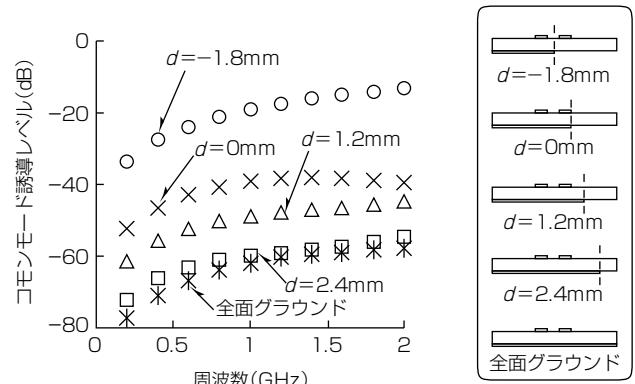
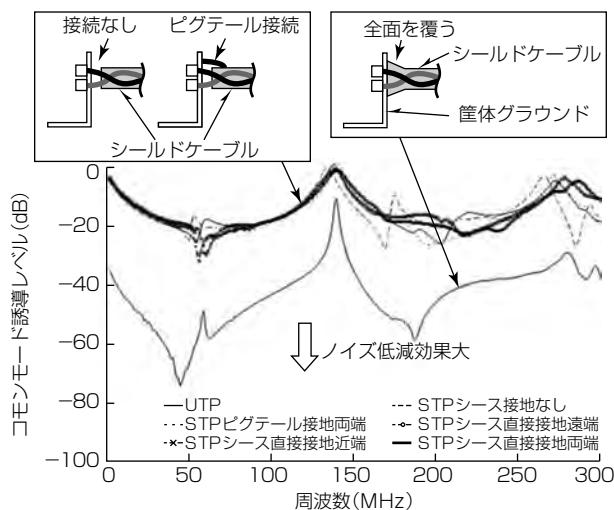


図2. プリント基板のグラウンド端部の差動信号線の配線検討



UTP : Unshielded Twist Pair cable, STP : Shielded Twist Pair cable

図3. シールドケーブルとコネクタの接続方法の比較

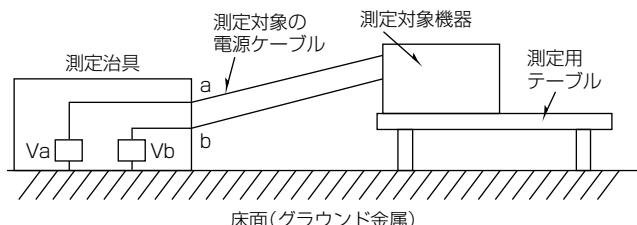


図4. 測定構成

シールド外導体と接続コネクタの金属シェルとの接続方法によって、耐ノイズ性能が変動する。この接続条件と耐ノイズ性能の関係を、実験結果を主に定量評価して実装設計に適用している。図3にこの一例、シールド外導体の全面を覆う形態での接続、単線化しての接続(ピグテール接続)、接続なしでのノイズ低減効果を示す。ピグテール接続や接続なしでは、耐ノイズ性能が大きく劣化する<sup>(2)</sup>。

製品の電源線からのノイズ漏えいを評価する伝導エミッション測定では、疑似電源回路網を用いて線大地間電圧を雜音端子電圧として測定する。図4に、この測定構成を簡略化して示す。伝導エミッションの抑制には、電源ラインノイズフィルタが有効であり、広く使用されている。このフィルタ回路の設計に当たってはノーマルモードとコモンモードに分けて検討すると回路構成に対応させた設計評価が行いやすい。そこで、伝導エミッション測定での雜音端子電圧をノーマル・コモンモードに分離して評価できるようにしてノイズフィルタの設計効率を高めている。図5は、雜音端子電圧のノーマルモードとコモンモードの回路計算による評価結果である。この結果を測定結果と比較することで回路計算によるノイズフィルタの設計検討を行う<sup>(3)</sup>。

### 2.3 EMC測定評価技術

当社の情報技術総合研究所では、国際規格ISO/IEC 17025に準拠した“EMC認定試験所”を運用している。当社製品のグローバル展開が進む中、EMC国際規格の変化に対応

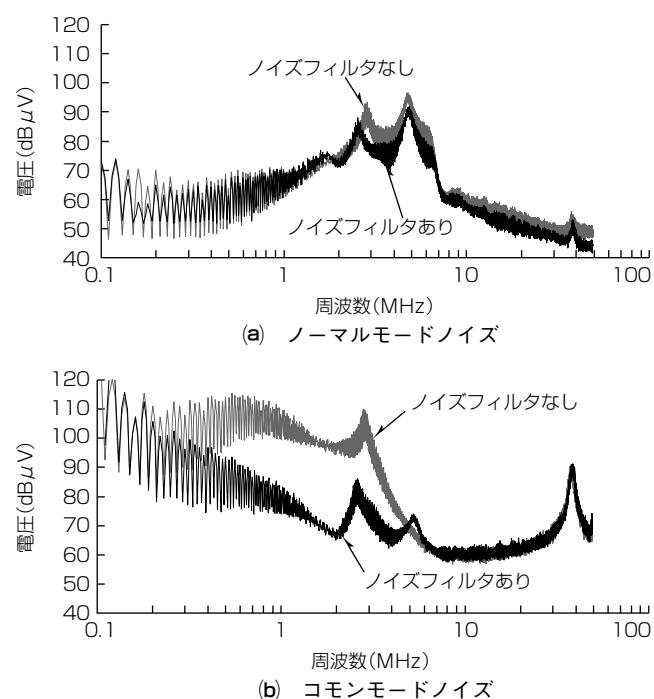


図5. ノイズモード分離の計算結果

するため、EMC規格動向を見ながら最新のEMC測定評価技術を社内で先駆けて構築し、当社製品のEMC品質を確認できる測定評価環境を実現している。

### 3. 製品に適用するEMC技術

ここでは、人工衛星<sup>(4)</sup>、家庭内に設置される無線ルータ内蔵ONU(Optical Network Unit)<sup>(5)</sup>、FA向けグラフィックオペレーションターミナル“GOT2000シリーズ”<sup>(6)</sup>に適用したEMC技術について述べる。

#### 3.1 人工衛星

人工衛星では、数kWクラスの太陽電池パネルの電力を電池に蓄えて機器に供給する電源制御装置、姿勢制御装置、通信用送信器のような大電力を扱う機器、各種センサや観測機器、地上からの微弱な電波を受信する受信機などの高感度・高精度が要求される機器、観測データや通信信号を処理する高速デジタル機器など、様々な機器が構体の内外に搭載され、これらの機器間を相互に接続するワイヤハーネスが衛星構体内に張り巡らされている。図6に衛星構体内的装置構成を示す。各コンポーネント間の電磁干渉を抑制し、各々が正常に動作するよう電磁干渉を抑制する設計を行う。このため、接地条件等を含めた機器間のインターフェース条件やワイヤハーネス、電磁シールドほかを総合的に設計する。

#### 3.2 無線LANルータ内蔵ONU

家庭内に設置される無線LANルータ内蔵ONUは、一枚のプリント基板上に全ての回路を実装し、この基板を樹脂筐体に収納して構成される。この装置は、図7に示すように、CPU、DDR(Double Data Rate)メモリ、Wi-Fi

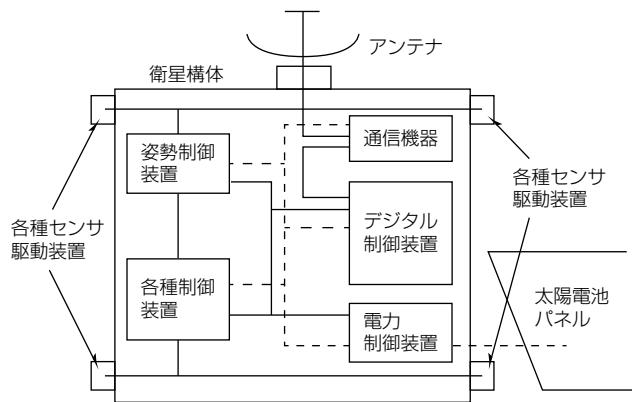


図6. 人工衛星の衛星構体内の装置構成

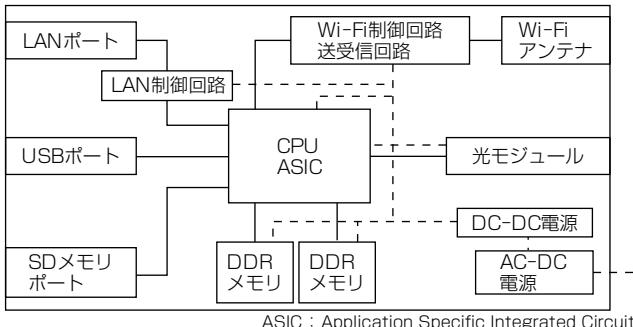


図7. 無線LANルータ内蔵ONUの構成例

(Wireless-Fidelity), 光送受信モジュール, USB, SD (Secure Digital) メモリ等の高速デジタル回路などで構成され, さらに, 電力量の大きいAC-DCやDC-DC電源回路, 微弱な信号を扱うWi-Fiの無線回路も同一基板上に実装される。このような装置では, デジタル信号系回路や電源回路からのノイズが基板から空間に放射される, LANなどのインターフェースケーブルに伝搬して放射される, 空間や配線を介して基板上のWi-Fi回路に干渉して感度劣化を引き起こすといったEMCリスクがある。さらに, 電源線を経由したサージに対する耐性やLANコネクタ部からのESD(静電気放電)に対する耐性なども要求されている。

このようなEMC課題に対処するため, ノイズ伝搬経路に対応したフィルタ適用設計, CPUの放熱板を部分的に基板のグラウンドに接続する部分シールド設計, 高速デジタル回路からのノイズ漏えいを少ない層数で防止するプリント基板のパターンレイアウト設計などを適用した。

### 3.3 GOT2000シリーズ

GOTは, 工場内装置を制御するFA機器である。ここでは“GOT2000シリーズ”に適用したEMC技術について述べる。図8にこの装置構成例を示す。FA機器は, 産業用ロボットなどの大型装置を動かすモータ駆動装置, 放電加工装置など大電力装置が多数存在する工場環境で使用され, 周囲環境のノイズが空間, 電力線, 信号線を通して干渉し, 誤作動を引き起こすリスクがある。また, ESDに対する

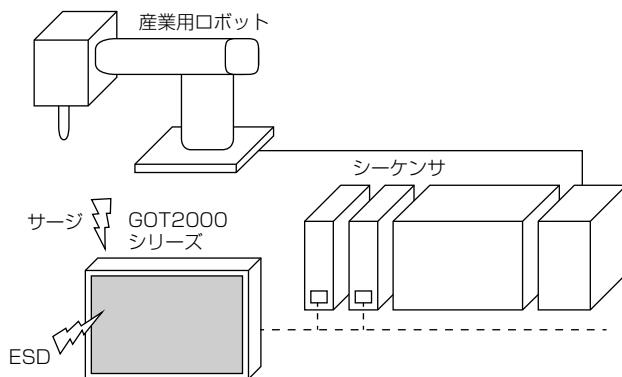


図8. GOT2000シリーズの装置構成例

ノイズ耐性も課題である。加えて, 放射エミッション許容値が設定されているため, 低ノイズ設計が要求される。

数百MHzのクロック信号やデータ伝送, ケーブル伝送, 拡張機能を備え, 基板上の実装密度も高いことから構想設計段階でのEMC設計方針検討と, 機構, 回路, 基板設計時の検討とレビューを行い, 2章で述べた要素技術を適用した。また, 装置内でのグラウンド系の機構設計では, 低ノイズ化とサージ耐性を得られるようにした。

## 4. む す び

当社の成長戦略を支える幅広い分野の製品のEMC品質に向けた取組みについて, EMC設計技術, EMC測定評価技術などのEMC設計検証技術や製品特有のEMC設計課題例について述べた。

EMC設計でのEMC基本3要素に基づいてEMCリスクを抽出・管理し, これら検討項目を製品設計フローに沿って体系的に取り組み, EMC設計効率化, 開発期間短縮, 製品のEMC品質の向上に努めている。

## 参 考 文 献

- (1) 米田 諭, ほか: グラウンド端が近接する差動伝送線路におけるコモンモード変換量の測定結果, 電子情報通信学会講演論文集, B-4-15 (2012)
- (2) 渡邊陽介, ほか: シールド付きツイストペアケーブルのノイズ耐性評価, エレクトロニクス実装学会誌, 14, No. 4, 267~271 (2011)
- (3) 廣瀬健二, ほか: 電源端子妨害電圧のディファレンシャルモードとコモンモードに関する評価検討, 電子情報通信学会技術研究研究報告, 112, No.372, 93~98 (2013)
- (4) 三菱電機技報2016年2月号 特集「宇宙利用を支える先端技術」
- (5) 三菱電機技報2016年6月号 特集「社会を支える通信技術」
- (6) 三菱電機技報2016年4月号 特集「e-F@ctoryを支える最新のFA機器」

# 空調用冷凍サイクル設計技術の高度化

羽下誠司\* 玉木章吾†  
山下浩司\*\* 竹中直史††  
鳩村 傑\*\*\*

Advancement of Refrigeration Cycle Design for Air Conditioning

Seiji Haga, Koji Yamashita, Takeshi Hatomura, Shogo Tamaki, Naofumi Takenaka

## 要旨

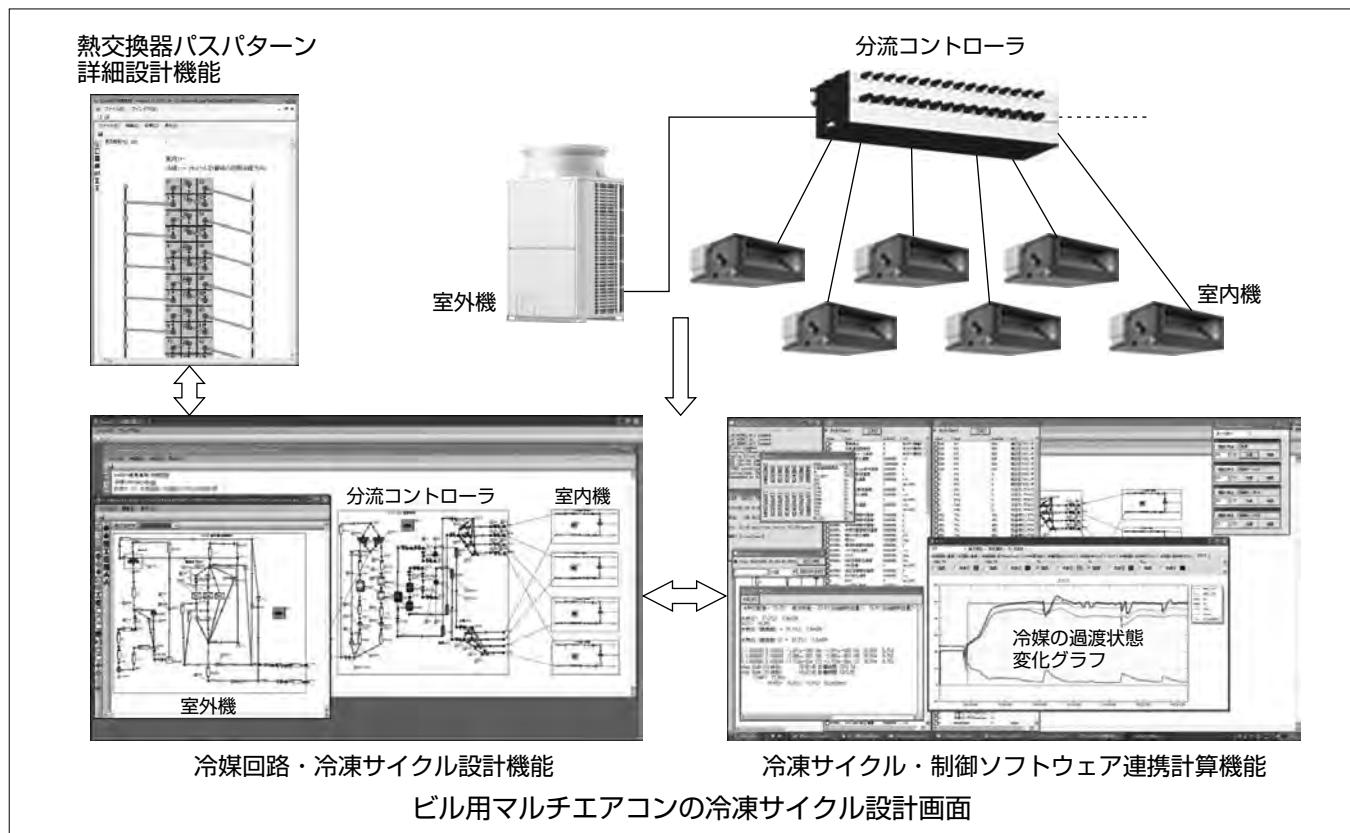
三菱電機では、家庭用エアコン霧ヶ峰を始め、ビル用マルチエアコンなど、冷凍サイクル(ヒートポンプ)を利用した空調機を全世界に向けて製造・販売している。空調機は、今まで日本などの温暖地域を中心に普及していたが、近年の世界的なCO<sub>2</sub>排出量の削減の動向や、新興国の経済成長を背景に、北米、北欧、ロシアなどの極寒地域や、インド、中東などのトロピカル(熱帯)地域へのニーズが高まっている。

従来極寒地域では、ストーブなど化石燃料を使用する暖房機器が普及していたが、最近のAPF(Annual Performance Factor: 通年エネルギー消費効率)が6以上の冷凍サイクル機器に置き換えることで、CO<sub>2</sub>排出量を削減できる。一方、日本などの温暖地域向けに設計した製品を、極寒地域やトロピカル地域へ出荷することはできない

ため、地域に特化した冷凍サイクルの再設計が必要となる。そこで課題となるのが、温度範囲の拡大・開発機種数の増大に対応した冷凍サイクル設計技術の高度化、効率化である。

この課題を解決するため、室外機、分流コントローラ、室内機などの冷媒回路をモジュール化し効率的に冷凍サイクルを設計するツール“dynaREF”を開発した。dynaREFは冷凍サイクルシミュレーションと制御ソフトウェアをパソコン上で連携させ、様々な外気温度、負荷条件での性能や冷媒制御をパソコン上で検討できる。

今後、dynaREFを活用することで試験の代替や設計手戻り削減が期待できる。さらに、ZEB(Zero Energy Building)、ZEH(Zero Energy House)、AI(Artificial Intelligence)など最新制御技術の早期検証ツールとして展開予定である。



## 空調機冷凍サイクル・冷媒制御設計ツール“dynaREF”的画面イメージ

ビル用マルチエアコンは、1つの室外機に分流コントローラを介して複数の室内機が接続されている。dynaREFでは、GUI(Graphical User Interface)ベースのエディタによって、室外機、分流コントローラ、室内機などの冷媒回路を部分的にモジュール化できる。それらのモジュールを再利用し、任意に組み合わせることで、様々な機種の冷媒回路モデルを構築できる。このGUIエディタをベースに、COP(Coefficient Of Performance: エネルギー消費効率)最大化設計、冷媒制御設計などが可能である。

\*設計システム技術センター \*\*空調冷熱システム事業部(工博) \*\*\*冷熱システム製作所  
†住環境研究開発センター ††先端技術総合研究所

## 1. まえがき

当社では、家庭用エアコンの霧ヶ峰を始め、ビル用マルチエアコンなど、冷凍サイクル(ヒートポンプ)を利用した空調機を全世界に向けて製造・販売している。空調機は、今まで日本などの温暖地域を中心に普及していたが、近年の世界的なCO<sub>2</sub>排出量の削減の動向や、新興国の経済成長を背景に、北米、北欧、ロシアなどの極寒地域や、インド、中東などのトロピカル(熱帯)地域へのニーズが高まっている。

従来極寒地域では、ストーブなど化石燃料を使用する暖房機器が普及していたが、最近のAPFが6以上の冷凍サイクル機器に置き換えることで、CO<sub>2</sub>排出量を削減できる。一方、日本などの温暖地域向けに設計した製品を、極寒地域やトロピカル地域へ出荷することはできないため、地域に特化した冷凍サイクルの再設計が必要となる。例えば、高温外気地域では、冷媒をより高温、高圧で、極低温外気地域では、冷媒をより低温、低圧で使用する必要があり、国内仕様の製品よりも冷媒に対する耐圧性を向上させる必要がある。さらに、これらの地域向けの冷凍サイクルは高低圧差が増大するため、圧縮機を能力の高いものに変更したり、圧縮機の吐出温度が高くなりすぎないように、圧縮過程中的冷媒を冷却するインジェクション圧縮機などの高機能部品に変更したりするなどの対応が必要な場合がある。

このように空調機を様々な地域に特化するとなると、開発機種数を増大させ、今まで想定していなかった外気温度帯の設計に対応する必要がある。そのため、冷凍サイクル設計技術の高度化、効率化が課題となる。この課題を解決するため、室外機、分流コントローラ、室内機などの冷媒回路をモジュール化して効率的に冷凍サイクルを設計する機能や、冷凍サイクルシミュレーションと制御ソフトウェアをパソコン上で連携させ、様々な外気温度・負荷条件での性能や冷媒制御をパソコン上で検討する機能を実現した空調機冷凍サイクル・冷媒制御設計ツールdynaREFを開発した。

dynaREFの冷凍サイクル計算アルゴリズムは、当社で開発したものを使用しており、冷凍サイクルの静特性、動特性をシミュレーションすることができる<sup>(1)</sup>。また、今回開発した冷媒回路のモジュール化・再利用の機能によって、機種数が増大しても、同じモジュールを再利用したり、モジュール単位で組合せを選択したりすることで、一から回路モデルを作成することなく、効率的に冷媒回路の構築と冷凍サイクル設計計算を実施できるようになった(2章)。さらに、冷凍サイクルと制御ソフトウェアをパソコン上で連携させることで、実機試験前に制御ソフトウェアの検証・最適化ができるようになった(3章)。

## 2. 冷媒回路モデルのモジュール化と再利用性向上

### 2.1 GUIベースの冷媒回路エディタ

冷媒回路設計は、今まで研究部門又は機能設計部門の限られた専任者が担当していた。しかし、開発機種数が増大すると、限られた専任者の数では設計リソースが不足するため、一般の機能設計者でも容易に冷媒回路設計ができるようにし、設計リソース不足を解消する必要がある。そこで、一般の機能設計者でも容易に冷媒回路設計ができる環境として、図1に示すGUIベースの冷媒回路エディタを開発した。冷媒回路エディタによって、使い慣れている表計算ソフトウェアやプレゼンテーションソフトウェアと同様の操作性で冷媒回路の作図と編集が可能となり、一般の機能設計者でも容易かつ直感的に冷媒回路の作図と編集を行い、冷媒回路の改善検討ができるようになった。

さらに、冷媒回路データのモジュール化機能によって、冷媒回路を室外機、分流コントローラ、室内機などの単位でデータベース(又はひな形)を構築することが可能となり、冷媒回路の再利用性を向上させた。図2に、冷媒回路の室外機部分、室内機部分をモジュール化した例を示す。また、熱交換器、圧縮機、LEV(Linear Expansion Valve :

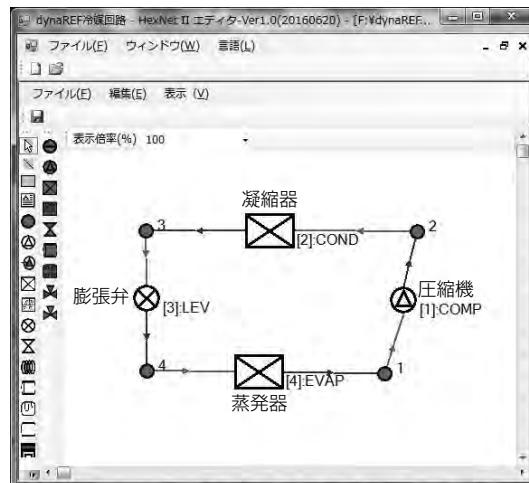


図1. GUIベースの冷媒回路エディタ画面

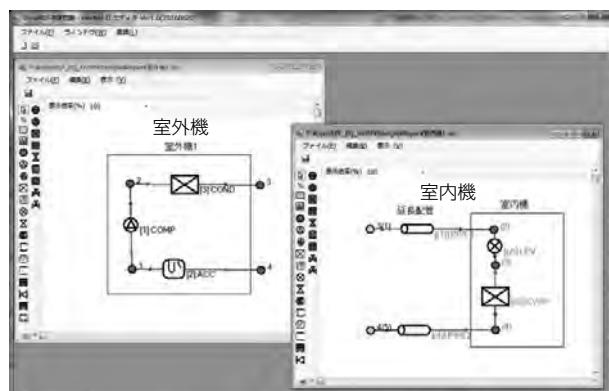


図2. 室外機と室内機の冷媒回路のモジュール化

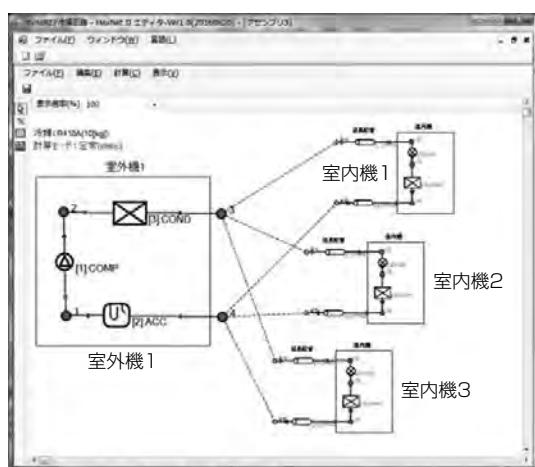


図3. モジュールを組み合わせた冷媒回路モデル例

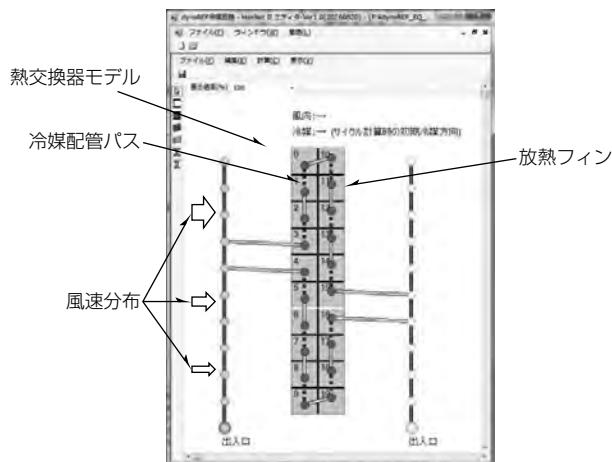


図4. 热交換器の冷媒配管パス設計画面

電子膨張弁)などの性能データも型名ごとにデータベース化しており、冷媒回路図上で、容易に熱交換器、圧縮機、LEVの型名を変更して性能への影響を検討することができる。一方、ビル用マルチエアコンでは、室外機、室内機を任意に組み合わせることが可能となるため、客先の特定の物件に対応した構成を事前に性能評価することが容易に実現可能となった。図3に、1つの室外機モジュールに、複数の室内機モジュールを組み合わせたマルチエアコンの冷媒回路モデル例を示す。

さらに、熱交換器の冷媒配管パス設計と冷凍サイクルの連携シミュレーションも可能であり、熱交換器の冷媒配管パスや熱交換器前面の風速分布の違いが冷凍サイクルに与える影響も検討することができる(図4)。

## 2.2 COP自動最大化計算機能

冷凍サイクル設計の目的の1つは、COPを最大化する省エネルギー設計である。従来は多数の設計変数を手動で変化させ、パラメータサーベイによってCOPが最大となる設計変数の組合せを導出していた。この方法は、人によって変化させた設計変数、範囲、刻みが異なると、得られる設計解も異なる問題があった。

そこで、dynaREFでは、COPを自動最大化する機能を

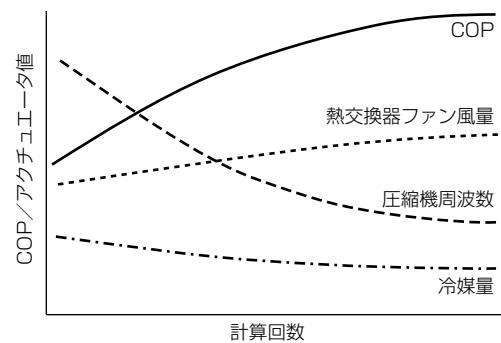


図5. COP自動最大化計算結果のイメージ図

開発した。冷媒回路のアクチュエータ(圧縮機周波数、熱交換器ファン風量、LEVパルス(キャピラリチューブ長さ))、冷媒量を設計変数として、COPに対する感度を計算してCOPが最大となるアクチュエータの数値の組合せを自動的に最適化することが可能となった。図5にCOP自動最大化計算結果のグラフのイメージ図を示す。このグラフから、COPを増加させるには、どのアクチュエータの値を増減させればよいか把握できる。

## 3. 冷凍サイクル動特性シミュレーション機能と制御ソフトウェアの連携

### 3.1 背景

環境試験室による性能実験及び制御検証は、環境試験室の能力、サイズ(高低差)の制約によって、客先での設置状況・負荷を十分に模擬できず、出荷前の検証や客先での不具合再現が困難な場合がある。また、動作限界を見極めるために、実際よりも過酷な外気温度条件の検証も必要である。しかし、環境試験室によっては、そのような温度帯に対応していないものもある。

そこで、実機と同じ制御ソフトウェア(各機器の故障保護機能を含む)とdynaREFの冷凍サイクル動特性シミュレーション機能をパソコン上で連携させ、実機と同じ冷媒状態の過渡的な挙動をパソコン上で再現するシミュレーション環境を構築した。今までではパソコンの性能が低く、このようなシミュレーション環境を構築しても、実機による実験の方が早く結果を得ることができた。しかし、最近のパソコンの高性能化やCPUの並列化を背景に、パソコン上のシミュレーションの方が、実機による実験より早く結果を得ることができるようになってきた。これによって、様々な外気温度・負荷条件や、様々な室外機、室内機などのユニットを組み合わせて、最適な制御定数の設計を実機試作前にパソコン上でバーチャルに検討できるようになった。

### 3.2 冷凍サイクル動特性シミュレーション機能と制御ソフトウェアの連携の仕組み

冷凍サイクルの動特性シミュレーションは、1章で述べたように当社で開発したものをベースとして、冷房・暖房・停止モード切替え時の冷媒流れの停止・逆流に対応し

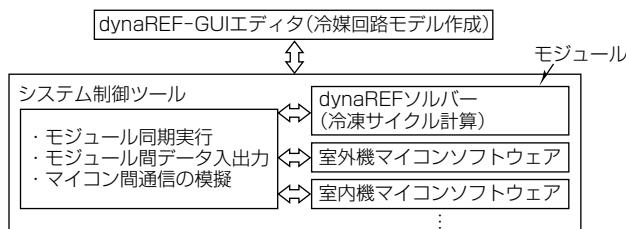


図6. dynaREFとシステム制御ツールの関係

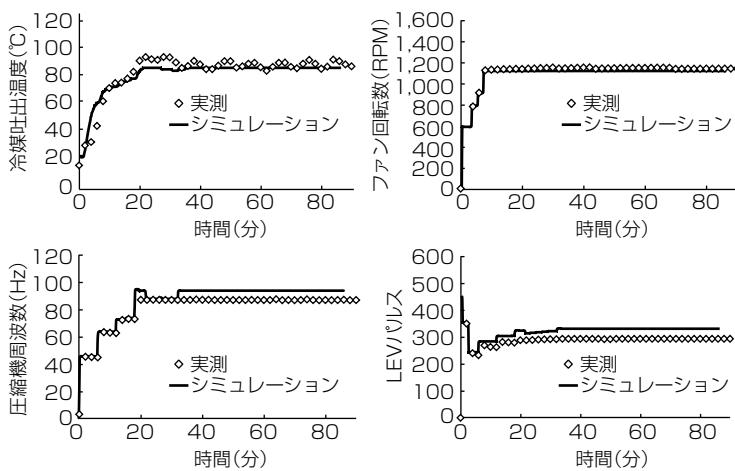


図7. 冷凍サイクル動特性シミュレーションと実測比較

た機能強化や計算速度の高速化を図るとともに、制御ソフトウェアと連携させるためのインターフェースを開発した。制御ソフトウェアでは、ROM化するソースコードそのものをパソコン上で設計検証できるように、制御基板をエミュレートする環境を構築した。さらに、冷凍サイクルの動特性シミュレーション機能と制御ソフトウェアの動作の同期及び室外機と室内機間のネットワーク通信を模擬する上位の“システム制御ツール”を開発した(図6)。この仕組みによって、制御マイコンソフトウェアを変更して、冷凍サイクルの動特性への影響を容易に確認することができるようになった。シミュレーション上で最適化した制御ソフトウェアは、ソースコードに手を加えなくても、そのままROM化することができるため、バグの作り込みを抑制することができるようになった。

また、システム制御ツールは、dynaREFのGUIエディタからシームレスにバックグラウンドで実行できるため、一般的の機能設計者は、複雑な仕組みを理解しなくても、この機能を利用できる。

### 3.3 実測とシミュレーション結果の比較

圧縮機起動後の冷媒吐出温度、圧縮機周波数、ファン回転数、LEVパルスなどのアクチュエータの実測結果と解析結果の一例を図7に示す。

試験では、境界条件である外気温度(試験室温度)が周期的に変動しているのに対し、シミュレーションでは外気温度は理想的に一定としているため、冷媒吐出温度の過渡変化に若干の差異がみられるものの、冷凍能力に大きく影響

する冷媒吐出温度の立ち上がりや一定となる時間(時定数)、消費電力に大きく影響する圧縮機周波数とファン回転数の過渡変化がほぼ一致しており、COPの過渡変化を5%程度の精度で再現することを確認した。有効数字が二桁程度求められるCOPの計算では十分な精度である。

一方、シミュレーションの計算時間は、使用するパソコンのスペックにもよるが、例えばCPUとしてIntel社のXeon<sup>(注1)</sup> E5-2690 2.6GHz(2CPU, 12コア)を使用した場合、小規模の構成(室外機1台+室内機4台)では実際の試験時間の1/10、中規模の構成(室外機1台+分流コントローラ1台+室内機10台)であれば1/2、大規模の構成(室外機2台+分流コントローラ1台+室内機30台)であれば、同等の時間で計算できる。パソコンの性能が向上すれば、計算時間は更に短縮できる。今後は、これらのようないくつかの単一系統(1つの閉じた冷媒回路で構成される系統)の製品が複数連携するような超大規模モデルに対応し、ビル丸ごとの空調の計算ができるよう開発を継続している。

(注1) Xeonは、Intel Corp. の登録商標である。

## 4. むすび

今後の空調機の市場拡大が予想される極寒地域及びトロピカル地域の機種開発の高度化・効率化に対応した空調機冷凍サイクル・冷媒制御設計ツールdynaREFの開発について述べた。このツールは、国内設計者だけではなく、海外拠点の設計者へも展開して既に地域に特化した製品設計に適用を始めている。

今後、更に冷凍サイクルシミュレーションの高精度化・高速化を図り、試験検証項目をシミュレーションへ置き換えることによる開発工数削減も検討していく。また、ビル空調機全体のシミュレーションを可能とし、ZEB・ZEHの検証やIoT(Internet of Things)、AIによる新制御方式の早期検証ツールとしても活用していく予定である。世界的な低GWP(Global Warming Potential: 地球温暖化係数)化が加速しており、次世代冷媒の早期検証や水回路とのハイブリッド空調(Air To Water)の設計・検証ツールとしても展開予定である。

今後も更なる空調設計技術の高度化を継続し、地域に特化した省エネルギー性、快適性、品質が高い空調製品を迅速に市場に提供していく。

## 参考文献

- (1) 畠崎史武, ほか: 非共沸混合冷媒を用いた蒸気圧縮式冷凍サイクルの動特性計算モデル第1報: 非共沸混合冷媒に対応した汎用計算モデル, 日本冷凍空調学会論文集, 18, No.3, 321~330 (2001)

# 製造ばらつきを考慮した機構解析技術

幸本宏治\* 渡辺和昌\*\*  
舛田真一\*\* 三木伸介\*\*\*  
中田光昭\*\*

Method for Analyzing Multibody Motion Considering Production Tolerance

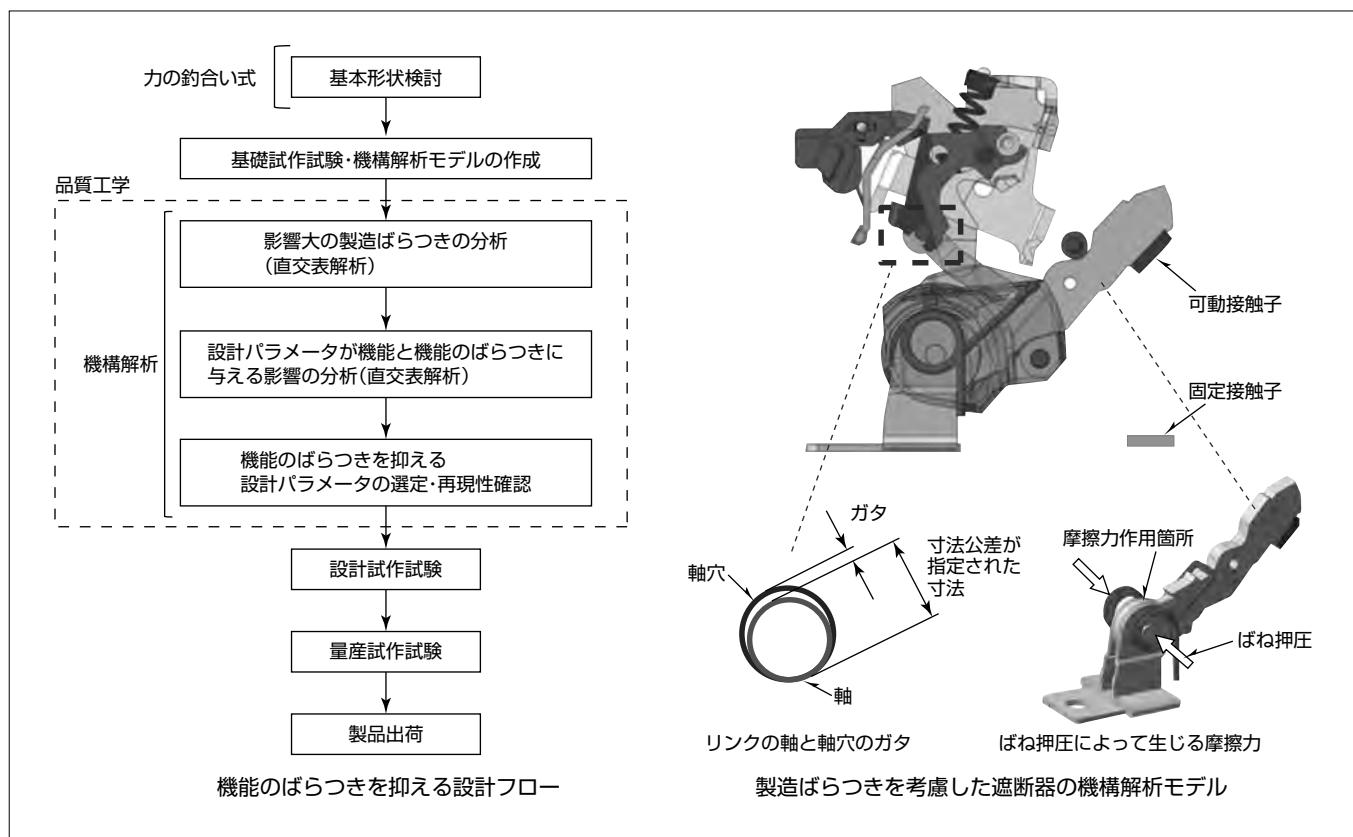
Koji Sachimoto, Shinichi Masuda, Mitsuaki Nakata, Kazumasa Watanabe, Shinsuke Miki

## 要旨

複数の部品がリンクやばねで接続された駆動機構を持つ量産製品の設計では、寸法公差、組立て公差等の製造ばらつきが製品の機能に大きく影響を与えるため、設計自由度の高い設計初期段階からそれらの影響を把握することが重要である。しかし、設計の初期段階では量産用の設備がないため少数の試作品しか製作することができず、製造ばらつきが製品の機能に与える影響を正確に把握することは難しい。設計の後工程になって製造ばらつきによる製品機能のばらつきの問題が顕在化すると、対策に多大な時間と費用を要する。そのため、駆動機構を持つ量産製品では、設計初期段階で製造ばらつきによる機能のばらつきを評価し、機能のばらつきを抑える設計技術の開発が望まれている。

そこで、産業用分電盤等で使用される遮断器を対象として、近年普及が進んでいる機構解析ソフトウェアを活用し、設計初期段階で品質工学を適用することによって機能のばらつきを抑える機構解析技術を構築した。

まず、製造ばらつきを詳細に反映した機構解析モデルを構築した。次に、製造ばらつきを反映した多数の機構解析を短時間で実施するために、解析モデルを自動生成する設計環境を整備し、2週間以上かかる解析作業を1日で実施可能にした。また、量産機種で検証試験を実施し、機能の特性値のばらつきが解析による予測値と同様の傾向を示すことを確かめ、構築した機構解析技術の有効性を確認した。この技術は遮断器の新機種開発に適用されている。



## 製造ばらつきによる機能のばらつきを抑える機構解析技術

機能のばらつきを抑える設計フローは、設計初期段階で機構解析ソフトウェアを活用し、品質工学を適用することによって機能のばらつきを抑える。遮断器の機構解析モデルは、リンクの軸と軸穴のガタや摩擦等の製造ばらつきを詳細に反映した。

## 1. まえがき

複数の部品がリンクやばねで接続された駆動機構を持つ量産製品の設計では、寸法公差、組立て公差等の製造ばらつきが製品の機能に大きく影響を与える。そのため、設計自由度の高い設計初期段階からそれらの影響を把握することが重要である。しかし、設計の初期段階では量産用の設備がないため少數の試作品しか製作することができず、製造ばらつきが製品の機能に与える影響を正確に把握することは難しい。設計の後工程になって製造ばらつきによる製品機能のばらつきの問題が顕在化すると、対策に多大な時間と費用を要するため、駆動機構を持つ量産製品では、設計初期段階で製造ばらつきによる機能のばらつきを評価し、機能のばらつきを抑える設計技術の開発が望まれている。

一方、近年のCAE(Computer Aided Engineering)技術の進歩によって、市販の機構解析ソフトウェアを用いた駆動機構の動作解析技術が普及してきている。

本稿では、産業用分電盤等で使用される遮断器を対象として、機構解析ソフトウェアを活用し、設計初期段階でばらつきを抑える設計技術を構築した事例について述べる。

## 2. 製造ばらつきを考慮した機構解析

### 2.1 遮断器の仕組み

本稿で対象とする遮断器の構成を図1に示す。遮断器は回路を開閉する可動接触子と固定接触子、機械的に接触子の開閉を行う取手と開閉機構部、過電流に応動して接触子を開く過電流引き外し装置を含み、それらがモールドケースで一体に構成される<sup>(1)</sup>。

可動接触子と固定接触子は、取手のON/OFF操作又は過電流引き外し装置の作動によって開閉機構部が動作して開閉される。開閉機構部の動作は、図2に示すトグルリンクと呼ばれるリンク機構によって行われる。図2で取手をONからOFFに操作すれば、ばねの作用線がトグルリンクのデッドポイントラインを超えてリンクが屈曲し、逆に取手をOFFからONに操作すればリンクが伸長する(図2(a))、

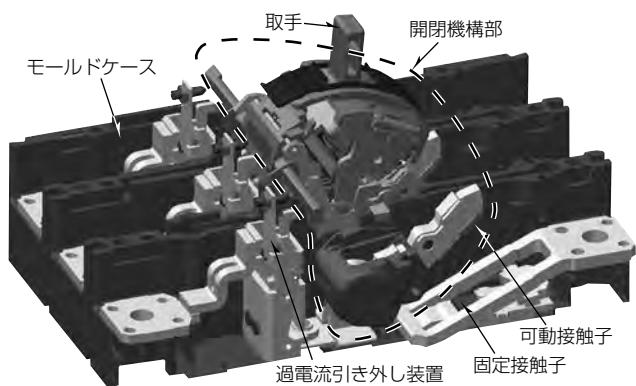


図1. 遮断器の構成

(b))。過電流が流れたときは過電流引き外し装置が作動し、レバーとラッチの係合部がはずれ、ばねの荷重が解放されて接触子が開閉する(図2(c))。

### 2.2 機能のばらつきを抑える設計フロー

遮断器の設計では、各動作における力の釣合い式によって動作の成立性を検討し、基本形状を決定する。このとき、設計パラメータとなるリンクの長さやばね荷重等の寸法公差のばらつきの影響は考慮することができるが、摩擦等の組立て公差まで含めた製造ばらつきの影響を正確に評価することは難しい。設計後工程の量産試作で機能の特性値のばらつきが仕様を満たさない場合は、設計パラメータの調整によって改良が施されるが、製造ばらつきは実験で測定できないことも多く、対策には多大な時間とコストを要する。

そこで、製造ばらつきを詳細に反映した機構解析モデルを構築し、設計初期段階で品質工学<sup>(2)</sup>を適用してばらつきを抑える設計を実施することにした。図3に設計フローを示す。

品質工学では、まず製品の機能に影響を与える製造ばらつきの候補を抽出し、直交表に基づいた解析を実施して影

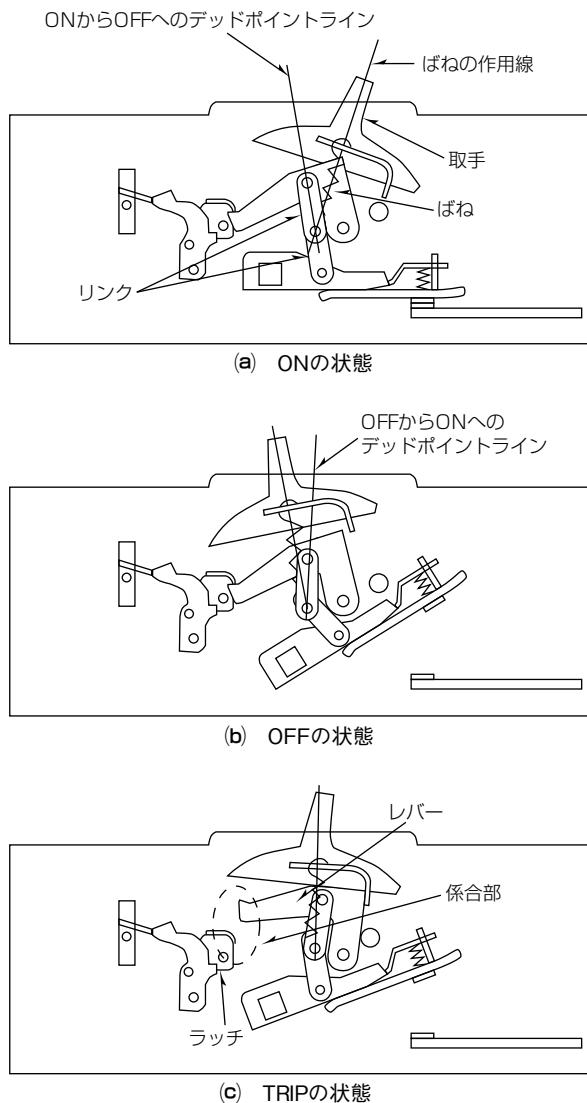


図2. 開閉機構部の動作の仕組み

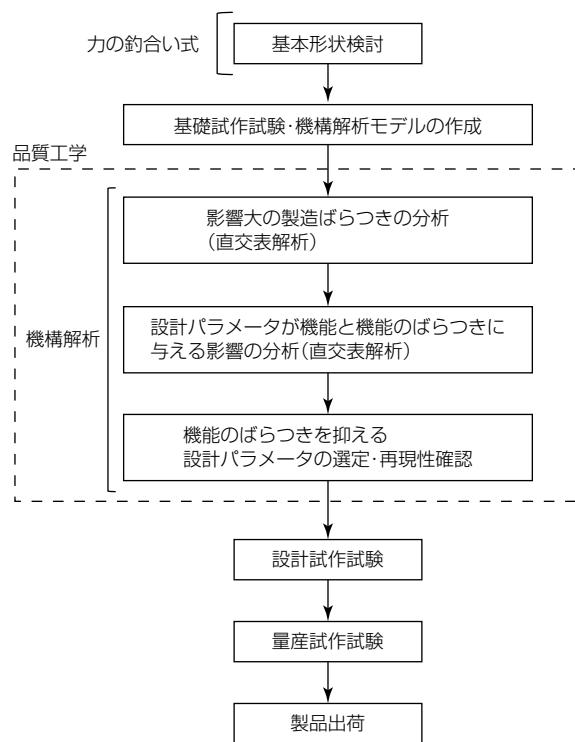


図3. 機能のばらつきを抑える設計フロー

響の大きい製造ばらつきを特定する。次に、影響が大きい製造ばらつきの影響下で、設計パラメータが製品の機能と機能のばらつきに与える影響を直交表に基づいた解析を再度実施して分析する。得られた分析結果から製品の機能のばらつきを抑える設計パラメータの組合せを選定し、確認解析を実施して選定した仕様の再現性を確認する。

図3のフローを実現するには、製造ばらつきを詳細に反映した機械解析モデルの構築と、品質工学でばらつきを反映した多数の解析を実施するための解析モデルの作成工数の削減が課題であった。

### 2.3 機構部の詳細解析モデルの構築

量産機種をベンチマークとして、製造ばらつきを詳細に反映した機械解析モデルを構築した。機械解析では、部品間にリンク拘束、接触拘束、それらに作用する摩擦を定義することで、解析モデルを構築する。まず、これらの構成要素で遮断器の開閉動作を模擬する初期解析モデルを構築したところ、トグル機構の開閉動作は模擬できたが、可動接触子と固定接触子の接触荷重が実測と整合しなかった。

そこで、解析モデルに反映されていない製造ばらつきを網羅的に洗い出し、その中で影響が大きいと推測される次の製造ばらつき(1), (2)を反映したところ、接触荷重が実測と整合するようになった。開閉機構部の解析モデルを図4に、製造ばらつき(1), (2)の模式図を図5に示す。

#### (1) リンクの軸と軸穴のガタ

通常の機械解析ではリンクの軸と軸穴のガタは含まないが、全ての部品のリンクの軸と軸穴のガタに接触拘束を定義してモデル化した(図5(a))。

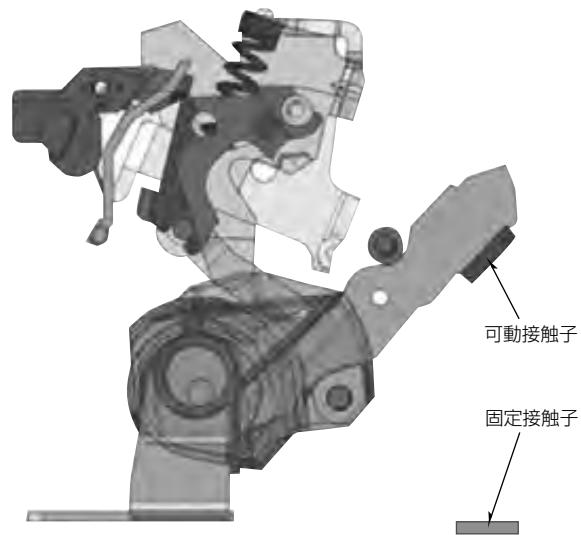
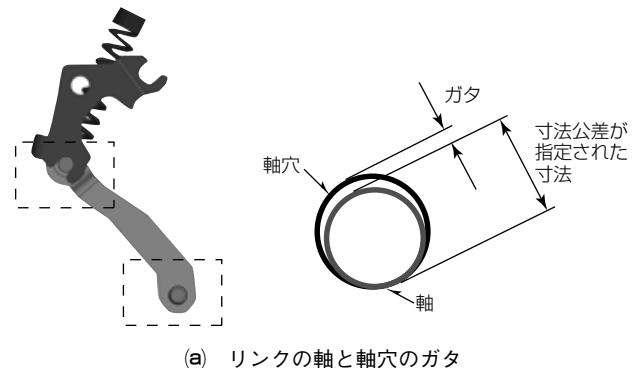
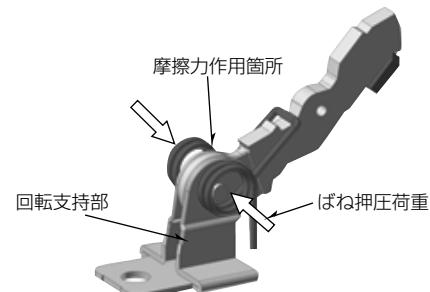


図4. 開閉機構部の解析モデル



(a) リンクの軸と軸穴のガタ



(b) ばね押圧によって生じる摩擦力

図5. 詳細解析モデルのポイント

#### (2) ばね押圧によって生じる摩擦力

可動接触子の回転支持部で、通電を確保するためのばね押圧によって生じる摩擦力をモデル化した(図5(b))。

### 2.4 機械解析モデルの作成工数の削減

機械解析モデルは図6に示すように、三次元CADソフトウェアで形状を作成した後、中間CADファイルを機械解析ソフトウェアにインポートし、機械解析の設定をして作成する。そのため、製造ばらつきを反映し、寸法が異なる機械解析モデルを複数作成するためには、CADソフトウェアで寸法が異なる形状を作成した後に、機械解析ソフトウェアにインポートして解析設定を付け直す作業が必要であった。なお、CADソフトウェアに付属した簡易的な

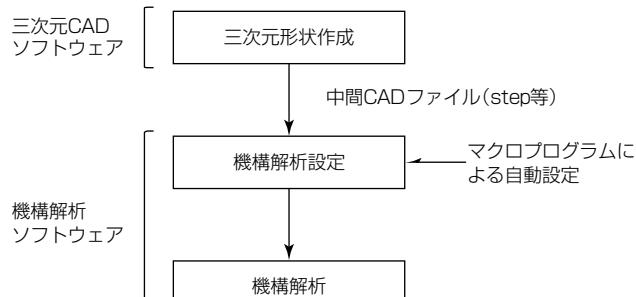


図6. 機構解析モデルの作成手順

機構解析ソフトウェアの中には、CADソフトウェアで形状を変更可能なものもあるが、専用の機構解析ソフトウェアと比較すると接触機能が弱く、ガタの影響を精密に再現することができない。

そこで、機構解析ソフトウェアにインポートしたCAD形状に対して、2.3節で定めた機構解析設定を自動で設定するマクロプログラムを整備した。マクロプログラムでは、部品間のリンクや接触面全てにIDを設定し、プログラムで同じIDを参照することで、寸法が異なる複数のCADモデルに同一の機構解析設定を自動作成できるようにした。

解析設定を自動作成する環境を整備することで、解析モデルの作成時間を大幅に短縮した。例として、ばらつき設計検討に80個の解析モデルが必要な場合、品質工学の適用に2週間以上かかったが、1日で検討できるようになった。

### 3. 実機検証

構築した機構解析技術を用いて、ベンチマークとした量産機種のON動作に関わる特性値の改善を検討した。検討対象の特性値は、可動接触子と固定接触子の間にスペーサをはさんだときにON動作が成立しなくなるスペーサの厚みとした。はさめるスペーサの厚みが大きいほど、製造ばらつきによってON動作の動きが鈍くなる現象に対してマージンがある。

まず、製造ばらつき因子を15個抽出し、それらがスペーサ厚みに与える影響を直交表に基づいた解析によって分析した。その結果、図7に示す“開閉機構部の浮き”と可動接触子の“回転ばねの荷重”の2つの影響が大きいことが判明した。

次に、設計パラメータとして、可動接触子の“回転ばねの荷重”“リンクAの長さ”，可動接触子の“ばね押圧荷重”を選定し、主な2つの製造ばらつきの影響下で、設計パラメータがスペーサ厚みに与える影響を分析した。その結果、設計パラメータの組合せとして2パターンの仕様1,2で、製造ばらつきによってON動作の動きが鈍くなる現象に対

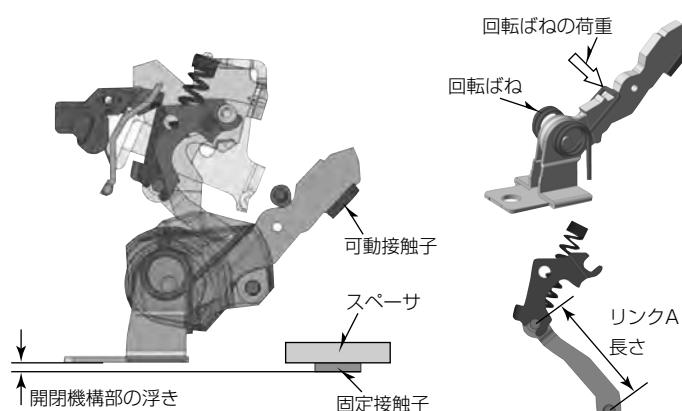


図7. ばらつき因子と制御パラメータ

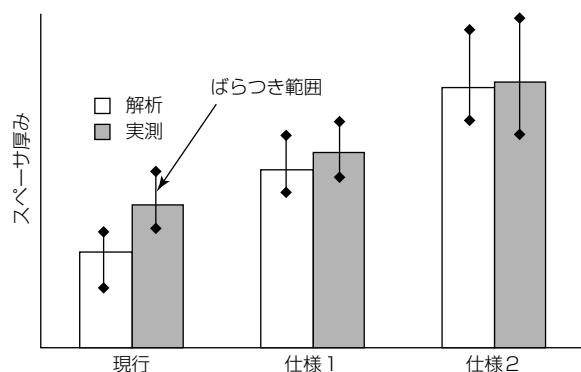


図8. 実機検証結果

してマージンが増える改善仕様であることが分かった。

改善前の仕様と改善後の仕様1,2について、実機検証試験を実施したところ、スペーサ厚みとばらつきが機構解析による予測値と同様の傾向を示した(図8)。そのため、構築した設計技術が有効であることを確かめることができた。

### 4. むすび

遮断器を対象として、機構解析ソフトウェアを活用し、設計初期段階で品質工学を適用することによってばらつきを抑える機構解析技術を構築した。この技術は遮断器の新機種開発に適用されている。

今後は適用機種を増やして製品のQCD(Quality Cost Delivery)改善に貢献するとともに、製造ばらつきのデータの蓄積を行い、製品機能のばらつきの解析精度の向上を行う。

### 参考文献

- (1) 三菱ノーヒューズ遮断器・漏電遮断器 技術資料集, 三菱電機(株) (2013)
- (2) 立林和夫, 入門タグチメソッド, 日科技連出版社 (2004)

# 3D測量データを活用した現地据付工事改善

安達裕輔\* 石川 秀\*\*\*  
安木俊之\*\* 桑名善太郎\*\*\*

*Improvement of Installation Work by Use of 3D Laser Scanner*

*Yusuke Adachi, Toshiyuki Yasuki, Zentaro Kuwana, Shu Ishikawa*

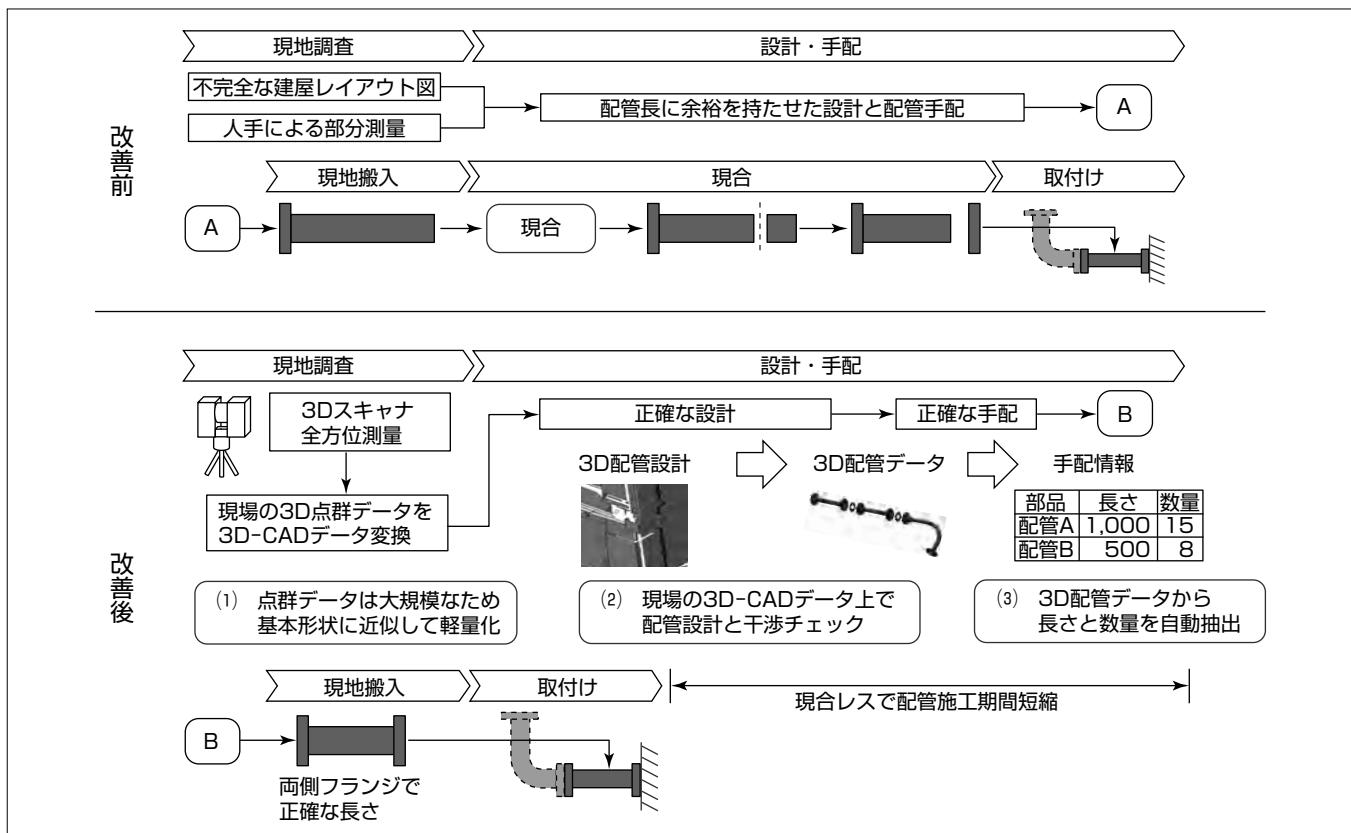
## 要旨

三菱電機製品の現地据付工事に必要な設備配管設計で、顧客から入手する建屋レイアウト図には設備全てが記載されていないケースがあるため、現地調査を行っている。この現地調査では、高所は足場を組む必要があるなど全てを網羅した測量が困難である。このため、現物合わせ(以下“現合”という。)を前提とした配管長に余裕を持たせた設計と配管手配をせざるを得ず、現地据付工事では、長めの配管で搬入し、現合で正確な長さに切断し、後付け溶接して組み付けを行っている。

近年、3Dスキャナの利用が注目されており、3Dスキャナによって正確な全方位測量を行うことが可能である。当社では、次の取組みによって3Dスキャナを配管設計に適

用することで現合を不要とし、施工期間を短縮することができた。

- (1) 3Dスキャナを利用し、高所など測量困難であった場所も含めた正確な全方位測量を実施する。
- (2) 3Dスキャナで測量したデータサイズが膨大な建屋の3D点群データを、面データに自動近似して軽量化することで、3D-CADによる建屋形状の参照を可能にする。
- (3) 3D-CADによる配管設計手法を構築し、建屋形状を参照した正確な設計を実施する。設計データを基にした配管部品の手配情報を自動抽出する仕組みを構築し、正確な手配を行う。正確な長さの部材手配によって現合レスを実現する。



## 3D点群データの配管設計への活用

現地調査では3Dスキャナによる全方位測量を実施し、得られた現場の3D点群データを3D-CADデータに変換する。設計・手配では、3D-CADで現場の3D点群データを基にした正確な設計と正確な手配を行い、両側にフランジを組み付けた正確な長さの部材を現地搬入して取り付ける。これによって現合レスを実現し、施工期間の短縮が可能となる。

## 1. まえがき

プラント設備等の新設・更新では、既存設備との据付検証や機器接続配管設計のために現地調査が必要となる。この現地調査では、高所は足場を組まないと測量できないなど人手で全て測量するのは容易でなく、近年3Dスキャナの利用が注目されている。

今回、当社製品の現地据付工事に必要な設備配管設計で、3D点群データを軽量な3D-CADデータに変換する技術の構築と、据付工事現場の3D-CADデータを活用した配管設計と手配の仕組みを構築した。

本稿では、建屋の3D点群データを活用した現地据付工事改善の取組みについて述べる。

## 2. 設備配管設計と現地据付工事の現状

設備配管設計で、顧客から提供される建屋レイアウト図には、顧客が追加した設備の配管など、一部設備が記載されていない場合がある。このため、配管と設備が干渉しないよう、事前に現地調査が必要となっている。

この現地調査では、建屋レイアウト図に記載されていない箇所を補うために、実際の現場に行き必要寸法の測量を行っている。高所など足場を組む必要があって測量が難しい箇所もあり、全てを網羅した測量は困難である。

このように、情報が不足している建屋レイアウト図や人手による部分測量の情報しかないと、現合を前提とした配管長に余裕を持たせた設計と配管手配をせざるを得ず、現地据付工事では、長めの配管で現地搬入し、現合で正確な長さに切断し、後付け溶接して組み付けを行っている。

## 3. 3Dスキャナによる現地調査

現地調査では図1に示す3Dスキャナを利用して、三脚固定のレーザ測量を行い、正確な全方位測量を実現した。この3Dスキャナでは、ミラー部の縦回転と首振りによる横回転によって縦横360°全方位にレーザを照射して測量を行う。測量データは、三次元座標値を持つ点の集合である3D点群データとして出力される。1回約3分のスキャン時間で4,000万点の3D点群データが出力される。図2の例で示すとおり、測量で得られるデータは3Dスキャナからのレーザ照射が可能な部分であり、柱などの影になっている部分のデータは得られないため影の部分は別の位置からスキャンして補う。このように建屋の複数箇所で測量を実施し、各箇所の3D点群データの位置合わせと合成を行い、建屋全体の3D点群データを作成する。

現地調査では、一部設備の記載がない情報が不足している建屋レイアウト図を補うために、これまで人手による部分測量を行っていたが、この3Dスキャナを利用した測量によって、高所など測量が困難であった箇所も含めて、建

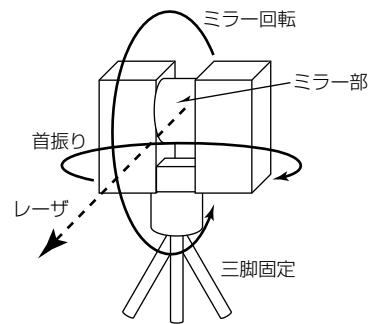
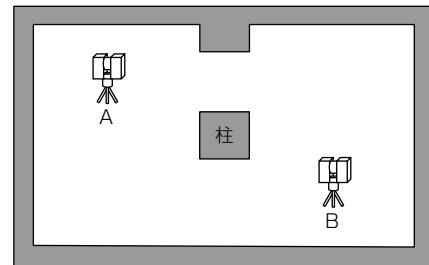
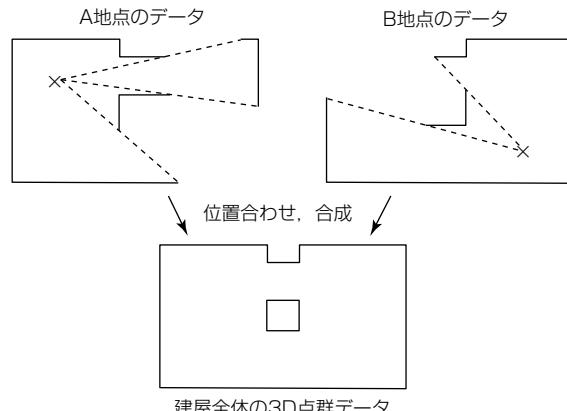


図1. 3Dスキャナ



(a) 建屋全体と測量箇所



建屋全体の3D点群データ

(b) 3D点群データ

図2. 複数箇所の計測例

屋全体の正確な測量を可能にした。また、簡単な操作による自動測量によって現地測量時間の短縮を実現した。

## 4. 3D-CADによる3D点群データの活用

### 4.1 設計における3D点群データ利用の課題

現地調査で得られる建屋の3D点群データを参照して配管設計を行うことで、据付工事現場の形状に従った正確な設計が可能になる。しかし、3D-CADで現地建屋の3D点群データを取り扱うためには、データサイズが膨大でメモリ不足によって3D-CADに取り込めないことが課題となっている。1箇所のスキャンで得られる3D点群データだけでも、3D-CADで読み込むには31GBのメモリ容量が必要である。

### 4.2 形状近似によるデータ軽量化

3D-CADに3D点群データを取り込むためには、データを軽量化する必要がある。軽量化の手法として、多数の点

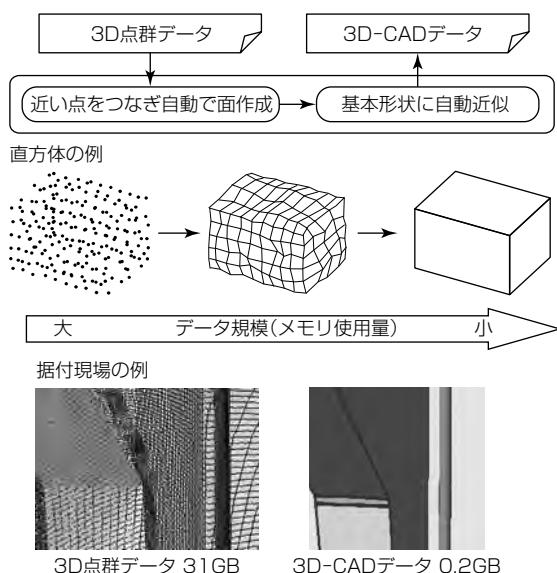


図3. 3D点群データの形状近似

のデータを面のデータに変換する技術<sup>(1)(2)</sup>に着目した。この技術によって、図3のとおり、3D点群データの近い点をつなぎ自動で面作成をして基本形状に自動近似することで、データ容量を少なくして3D-CADへの取り込みが可能となる。実際の建屋の3D点群データでこの技術を評価した結果、メモリ使用量99%減の大幅なデータ軽量化が実現でき、3D-CADに取り込み可能であることを確認した。

#### 4.3 3D点群データを活用した設計手法

4. 2節の技術に基づく3D点群処理ソフトウェアを利用し、3D-CADで据付工事現場の建屋形状を参照して配管設計する手法を図4のとおり構築した。

現地測量では、3Dスキャナを利用して据付工事現場の建屋全体の形状を取得する。取得したデータは3D点群処理ソフトウェアで、複数箇所の3D点群データの位置合わせと合成を行ったあとノイズ除去を行い、建屋全体の3D点群データを完成させる。その後、平面・円柱抽出で建屋の3D点群データを平面と円柱の簡易形状に近似した面データ（以下“建屋近似形状”という。）を自動生成する。この建屋近似形状を3D-CAD上に取り込み、形状を参照して配管設計を行う。建屋近似形状は建屋の8割ほどの形状が生成されており問題なく設計を行うことができるが、最終確認として、設計した3D配管データを3D点群処理ソフトウェアに取り込み、建屋の完全形状である3D点群データとの取り合い確認や干渉チェックを行うことによって、建屋近似形状で生成できていない箇所のチェック漏れを防止する。

また、配管設計時には建屋近似形状に加えて、2D-CADの建屋レイアウト図面を合成する。こうすることで、建屋レイアウト図面に図示された据付機器等によって建屋近似形状における設備の解釈を容易化する。設計結果は、設計した3D配管モデルと建屋レイアウト図面を投影して

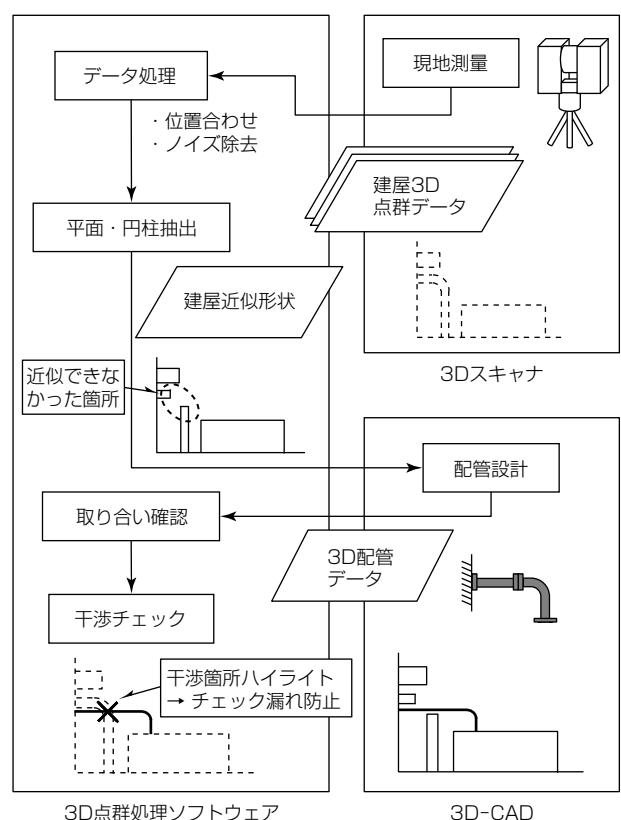


図4. 3D点群データを活用した配管設計の流れ

図面出力し、配管経路図面として据付現場で活用する。この手法によって、建屋の3D点群データを活用した正確な設計を実現する。

## 5. 3D-CADによる配管設計

## 5. 1 3D-CADによる配管設計手法

3D-CADによる配管設計手法を、図5のとおり構築した。経路全体を先に設計し、その後接合箇所を指定する手法としている。

最初に配管経路全体を一本の線で入力したあと、分割箇所を指定する。分割箇所の指定はマウスピックや寸法指定で行う。接合部品はあらかじめライブラリとして整備しておき、その中から配置する部品を選択する。配管設計時は、接合箇所の指示と接合部品の選択を行うことで、接合箇所の素管の分断、接合部品の挿入、接合部品との取り合い部の素管の長さ調整を3D-CADが自動で行う。

この設計手法によって、3D-CADによる配管設計の経験が浅い設計者でも理解しやすく、設計やモデリングにかかる手間を抑えた仕組みを実現している。

## 5.2 配管部品手配情報の自動抽出

設計・モデリングした3D配管データから部品名、寸法、数量を自動抽出して手配情報を自動生成する仕組みを、3D-CADの機能を活用して図6のとおり構築した。

3D-CADで設計した3D配管データをストレート配管、  
ベンド配管、フランジ、ガスケットの種類ごとに分類する。

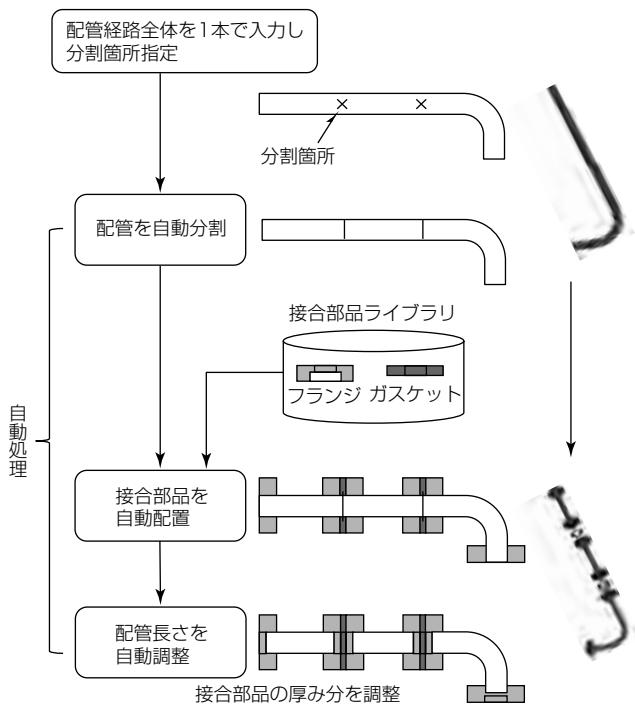


図5. 3D-CADにおける配管の自動分割

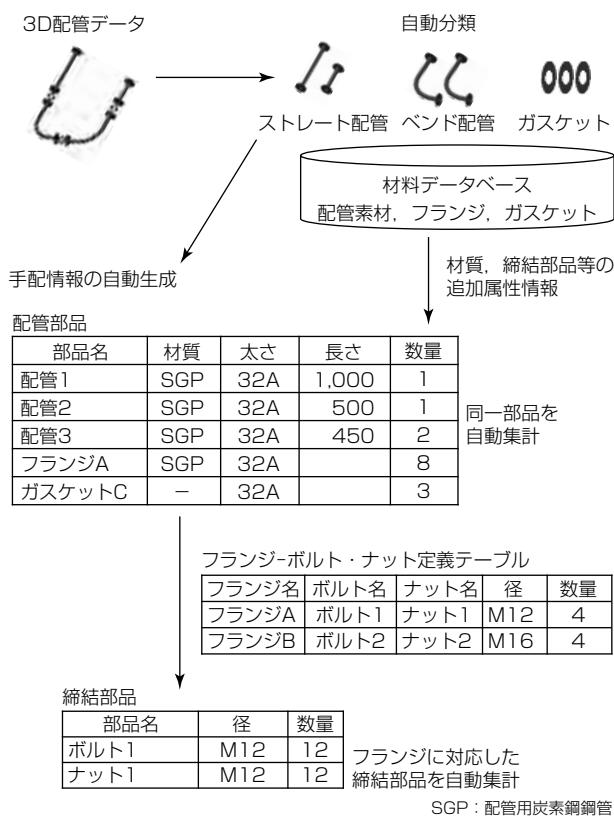


図6. 配管部品手配情報の自動抽出

配管部品は材料データベースを基に配管素材などの追加属性情報を加えて、同一部品を自動集計する仕組みとしている。また、接合箇所に必要なボルトとナットの締結部品は、利用している法兰ジの種類と関連付けを行うことで自動集計する仕組みとしている。この仕組みによって、配管部品の正確な手配とともに、手配情報作成にかかる作業時間の短縮を実現している。

正確な設計と正確な手配によって、現地据付工事では、両側に法兰ジを組み付けた正確な長さの部材を搬入し取り付けることが可能になり、現合レスを実現して施工期間を短縮している。

## 6. む す び

3Dスキャナを活用して現地設備を含む建屋の形状を取得し、取得した3D点群データを利用した3D-CADによる設計手法を確立したこと、据付工事現場の形状に合わせた正確な設計と正確な手配を行うことができ、現合レスによる施工期間短縮を実現した。

3D点群データの活用範囲は幅広く、既存機器と3D-CADで設計した新設機器の置換検証、新設機器の搬入経路及び動的干渉確認、搬入経路アニメーションの作成を行うことが可能である。また、仮想空間における疑似体験が可能なVR(Virtual Reality)技術との連携も進んでおり、現地の3D点群データを基にしたVRを用いた組立性・操作性検証といった様々な場面における活用が想定される。

今後、現実にある形状を3D-CADデータとして取得して活用できる場面やニーズは増えていくと考えられる。これらの活用技術を更に向上させ、あらゆる分野と融合した技術開発と新たなソリューションの提案によって社会貢献に努めていく。

## 参 考 文 献

- (1) 松沼千央, ほか: 画像特徴量を用いた大規模点群データからの円柱面と矩形面の検出, 日本機械学会論文集(C編), 76, No.772, 540~545 (2010)
- (2) 増田 宏: 大規模計測点群のための形状処理技術, 精密工学会 サイバーフィールド構築研究分科会 第2回分科会 (2010)  
[http://www.nakl.t.u-tokyo.ac.jp/~masuda/papers/cyberfield\\_2010\\_6\\_4.pdf](http://www.nakl.t.u-tokyo.ac.jp/~masuda/papers/cyberfield_2010_6_4.pdf)

# グローバル事業拡大に対応した梱包設計技術

潮 敬之\* 田中泰弘\*\*\*  
森 健晴\* 山田泰寛†  
瀧村文裕\*

Packaging Design for Global Business Expansion

Takayuki Ushio, Takeharu Mori, Fumihiro Takimura, Yasuhiro Tanaka, Yasuhiro Yamada

## 要旨

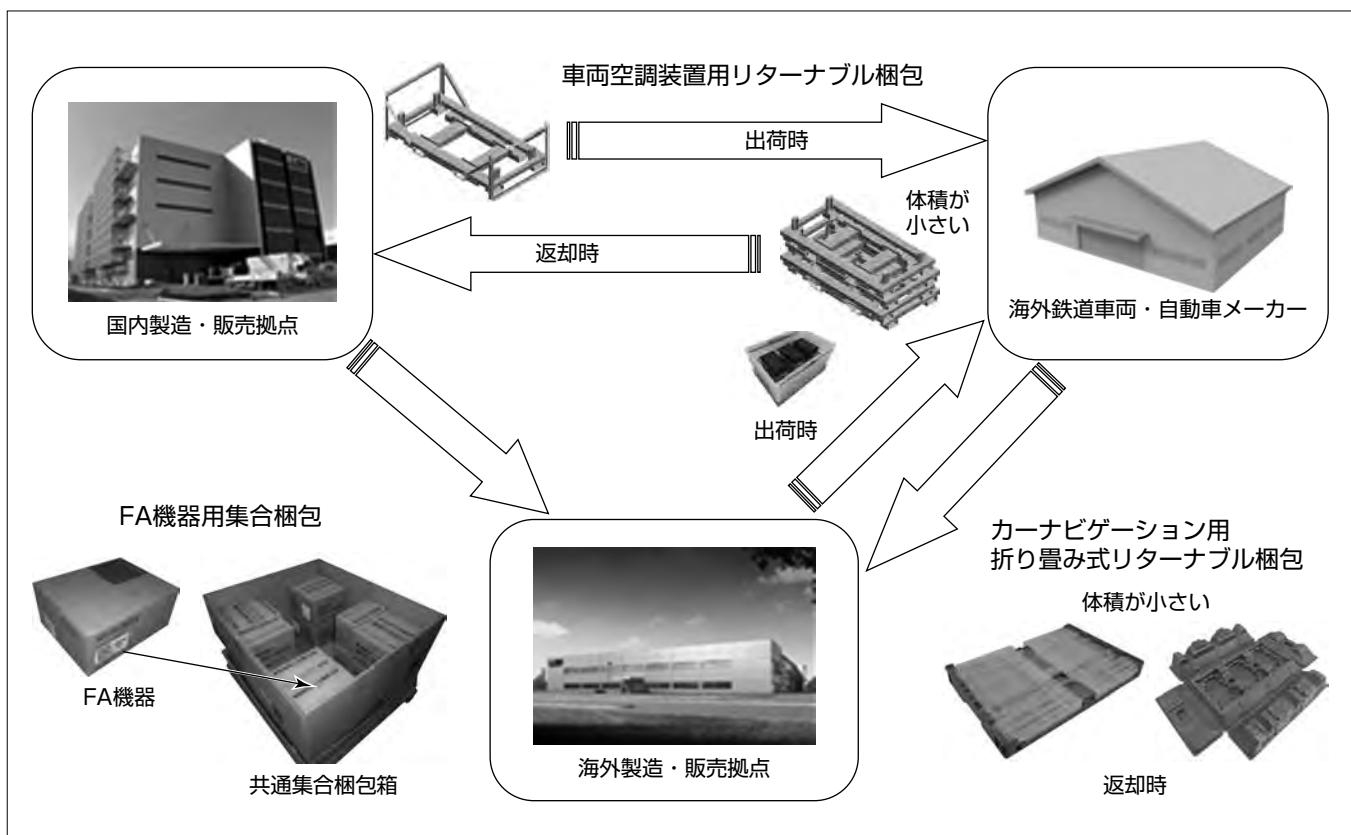
三菱電機グループは2020年度までに連結売上高5兆円以上、営業利益率8%以上を経営目標に設定している。また、成長率の著しい新興市場で事業横断的な地域戦略を成長戦略として強化しており、2020年度には全売上高の50%近くを海外売上げが占める計画である。

当社グループの成長戦略を支えるための物流面の課題は、グローバル市場での物流ルートの長距離化と複雑化、物流リードタイムの長期化による物流費の増加であり、これらをいかに抑制するかが鍵となる。これらの課題を解決するために、梱包(こんぽう)設計技術の開発や海外拠点の教育施策を推進している。

日本から海外販売・製造拠点に輸送するFA機器用の集合

梱包では、出荷4現場の集合梱包箱の仕様を共通化し、梱包費を削減した。日本から海外鉄道車両メーカーに供給する空調装置では、輸送架台のリターナブル化を進め、架台費用と輸送費の抑制を実現した。海外の製造拠点から他国の自動車メーカーに供給するカーナビゲーション用の梱包では、従来のリターナブル通い箱から折り畳み可能な樹脂トレイを開発し、梱包返送コストの削減を達成した。

今後、グローバル市場での競争力を高めるために、現地の梱包設計者教育を推進している。従来、日本国内の拠点で培ってきた経験と理論を体系化した梱包設計ガイドラインを作成し、海外製造拠点でのキーマン育成を推進している。



## グローバル事業拡大に対応した梱包設計技術の事例

グローバル市場における物流ルートの長距離と複雑化、及び物流リードタイムの長期化による物流費の増加に対して、FA機器用の集合梱包、車両空調機器用リターナブル梱包、カーナビゲーション用の折り畳み式リターナブル梱包等の開発によって、物流費を削減した。

## 1. まえがき

当社グループは経営目標として2020年度までに連結売上高5兆円以上、営業利益率8%以上を設定している。現在、海外事業の拡大に伴い、海外売上高は1.5兆円で全売上高の40%を占めている。経営目標の達成に向けて成長率の著しい新興市場、特にアジアを中心とした事業横断的な地域戦略を強化しており、2020年度には全売上高の50%近くを海外売上げが占める計画である。

しかし、昨今の世界経済を俯瞰(ふかん)すれば、2016年7月現在、新興国通貨安や英国のEU離脱に伴うユーロ安など、為替変動リスクが取り巻いている。こうした外的要因に対しても強固な経営基盤を構築することが必要である<sup>(1)(2)</sup>。そうした経営基盤強化の一環として、当社グループの物流費の抑制に取り組んでいる。

本稿では、梱包設計技術によって物流費用を削減した事例を述べるとともに、グローバル市場での更なる競争力強化を目的とした海外拠点での教育施策について述べる。

## 2. 成長戦略における物流面の課題

当社グループの成長戦略を実現するために、物流面の課題を整理する。海外市场の売上げ増加に伴い、売上高に占める日本国内からの輸出比率は増加する。これに加えて地産地消に向けた海外拠点数の増加に伴い、海外生産比率も増加している。

サプライヤー、生産拠点、販売会社、倉庫、顧客がグローバルに点在している状況は、物流ルートの長距離化と複雑化を招くとともに、物流リードタイムの長期化を引き起こす。つまり、当社グループの成長戦略を支える海外売上高比率の増加に伴い、連結物流費は増加するとともに連結売上高の増加率よりも高い比率で増加することが予測される。したがって、連結物流費をいかに抑制するかが、当社グループの成長戦略における物流面の課題となる。

物流費は梱包費、輸送費、保管荷役費に分類することができる。当社グループの物流費の平均的な内訳は、梱包費4割、輸送費4割、保管荷役費2割である。これら各費用の抑制には、梱包設計技術の開発とその技術の海外拠点への展開が重要である。

## 3. 梱包設計の考え方

梱包設計事例を述べる前に、簡単に梱包設計に必要なフローについて述べる。梱包設計は図1に示す5ステップのフローに従うことをお勧めする。はじめに、物流環境を調査する。輸送方法・距離、荷役方法、保管条件等を調査

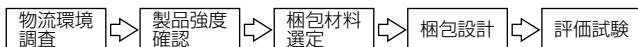


図1. 梱包設計の5ステップ

するだけでなく、輸送振動・衝撃値を計測して梱包強度設計に必要な外力を設定する。次に、内容物である製品の強度(易損性)を確認する。例えば、対象製品の許容加速度や振動外力に対する共振点を把握する。続いて、コストや廃棄性を考慮した梱包材料の選定と梱包設計を行い、規定に基づいた評価試験で梱包設計の妥当性を確認する。このように梱包設計の根拠を明確にして設計を進めることで、手戻りがなく、梱包費、輸送費、保管荷役費の最適化を狙うことが可能となる。

## 4. 梱包設計事例

グローバルに展開する製品のうち、梱包設計によって物流費を抑制した3事例について述べる。

### 4.1 国内から海外販売・製造拠点へ向けての集合梱包

海外市场向けFA機器は、完成品を日本国内4か所の製造・販売拠点から海外へ輸出するケースと、ノックダウン生産用の部品を海外へ輸出して海外製造拠点で製造する2つのケースがある。

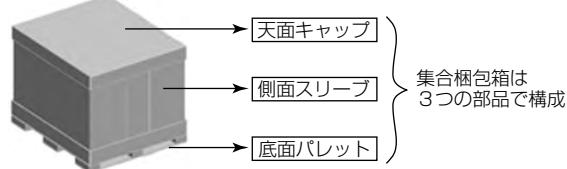
従来、製品・部品をまとめて集合梱包する際の大型段ボール箱は、各々の国内拠点で仕様が異なっており、コストにばらつきがあったため共通集合梱包箱の開発に取り組んだ。

まず日本から海外販売・製造拠点までの物流環境調査を行い、最大積載質量、荷扱いの梱包の要求仕様を再定義した。段積みによって上段の底面パレットから下段の集合梱包箱が受ける圧縮荷重に対して集合梱包箱の圧縮強度を決定し、これを満たす段ボールシートを選定した。4拠点の仕様を統一することによって、梱包費の削減を実現した(図2)。

### 4.2 国内から海外メーカーへ向けてのリターナブル梱包

海外の鉄道車両メーカー向け鉄道車両用空調装置の輸送架台はスチール製の溶接構造であり、従来、車両メーカーで廃棄されていたため無駄があった。そこで、輸送架台を廃棄せずに日本国内の工場と海外車両メーカーとの間で往復させて繰り返し使用するリターナブル輸送化を検討した。

架台の日本返却時に海上コンテナ内の積載台数を増やす



	国内拠点A	国内拠点B	国内拠点C	国内拠点D	共通化
キ天面 キャップ	コスト比率 1.00	1.07	1.01	1.01	1.00
ス側面 スリーブ	2.82	4.73	2.79	3.62	2.79
パ底面 パレット	7.07	7.71	7.33	6.74	6.74

図2. 集合梱包箱の共通化

ため、架台の支柱を取り外し式とした。しかし、輸送架台には空調装置固定用の突起があり段積み時に上段の架台と干渉するため、干渉せずに互い違いに積める構造とした。この対策によって安定的に段積み可能な構造となり、返却時の輸送費の低減及び從来廃棄していた架台費用を抑制し、物流費の削減を実現した(図3)。

#### 4.3 海外製造拠点から海外メーカーへ向けてのリターナブル梱包

海外自動車メーカー向けカーナビゲーション機器は、海外の工場で製造された後、直接海外自動車メーカーの工場所在国へ輸出される。従来は樹脂製通い箱と発泡スチロール製緩衝材を組み合わせて、製造工場と顧客との間でリターナブル輸送を行っていた。

従来の通い箱と緩衝材は、製品出荷時、通い箱回収時とも同じ占有体積であるため、回収時の輸送コストが割高となっていた(図4)。そこで、返却時、海上コンテナ内の通い箱積載数を増やすため、折り畳み可能な通い箱と樹脂トレイの開発に取り組んだ。折り畳み式通い箱は市販品を活用した。樹脂トレイは通い箱に挿入した状態では製品の保持が可能であり、樹脂トレイを通い箱から出すと平面状に展開して重ねることが可能な構造とした(図5)。

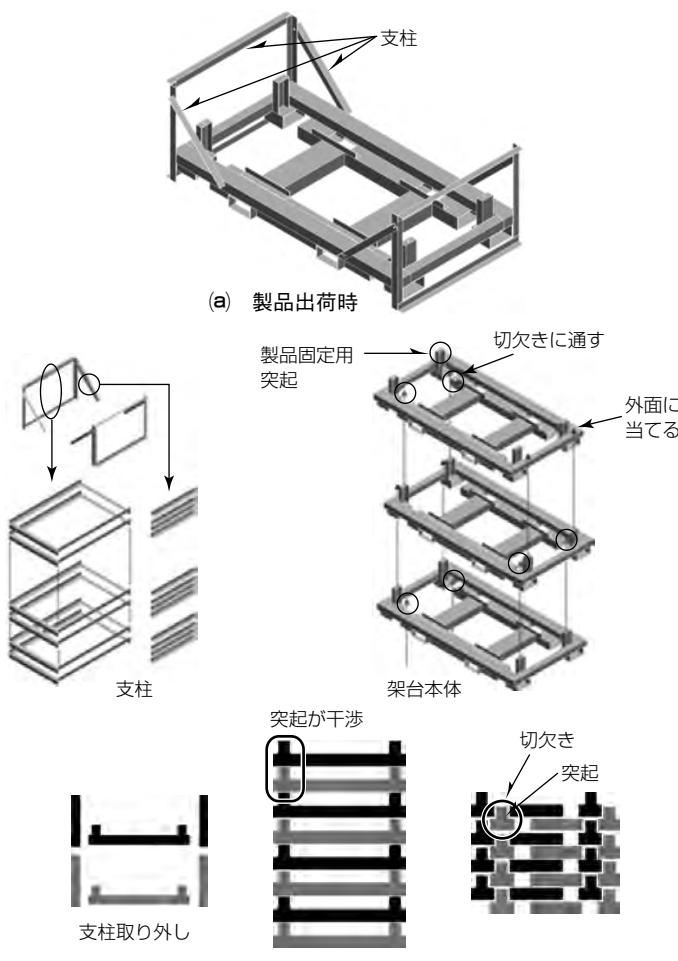


図3. リターナブル梱包

しかし、薄い樹脂トレイで従来の発泡スチロールと同様の落下衝撃に対する緩衝性能を確保する必要があった。そこで、樹脂トレイの製品支持部の周長に着目し、計算と落下衝撃試験による試行をし、発泡スチロール同等の緩衝性能を確認した(図6)。繰り返し使用による梱包費の抑制、輸送効率向上による輸送費の低減によって、物流費の削減を実現した。

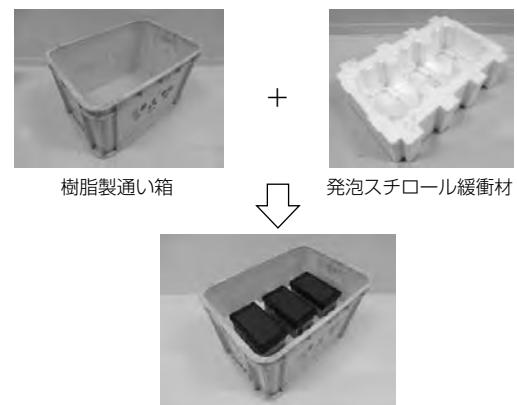


図4. 従来のリターナブル梱包

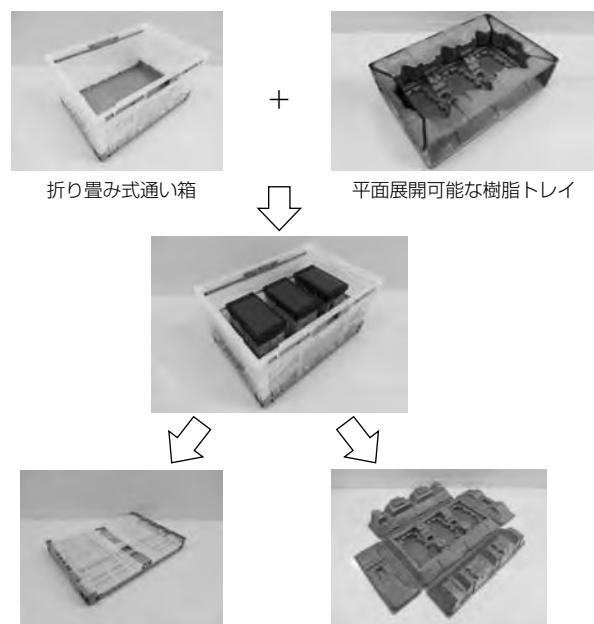


図5. 折り畳み式リターナブル梱包

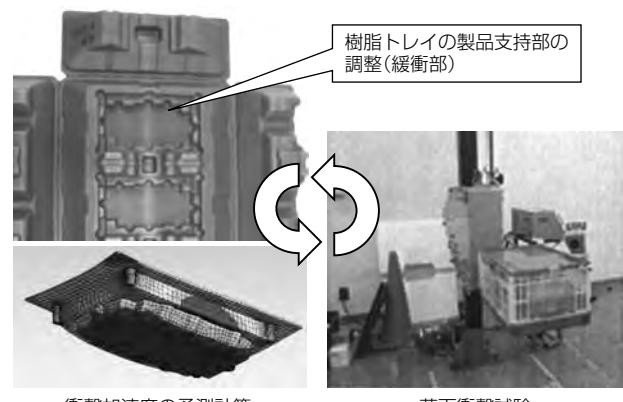


図6. 製品支持部の設計と検証

## 5. 海外拠点での教育施策

設計・生産の現地化に伴い、海外の製造拠点スタッフ自らが物流環境や製品の特性に特化した梱包設計を行い、物流費の最適化を実現する必要があるため、現地の梱包設計スタッフの指導・教育を推進している(図7)。

日本国内の梱包設計者の場合、詳細な設計マニュアルや手順書等が整備されていなくても、ベテラン技術者からの伝承によって過去の経験を活用した設計が可能である<sup>(3)</sup>。一方、海外では日本に比べ離職率が高く現場主体の教育体制を構築することが難しいため、日本で設計した梱包や図面を参考に設計したとしても、設計根拠を理解することができずには設計の手戻りとなる懸念がある。梱包の形はまねることができても、製品固定や衝撃に対する緩衝など本来製品の保護に必要な知識が欠落していると、十分な梱包設計ができない。

そこで、日本国内の製造拠点で、梱包設計における経験・ノウハウと理論式を組み合わせた設計ガイドラインの作成を進めている。木箱梱包の設計を例に挙げると、日本国内の設計では、各種木材寸法の取り方や間隔寸法、板の重ね方にはルールがあり、それらが構造的な裏付けを持っている。こうした長年の経験に基づくノウハウを整理するとともに、理論計算式を体系的に組み合わせてガイドラインに盛り込んだ。作成した梱包設計ガイドラインを基に、

海外製造拠点でグループワーク教育を推進している。まず、現地の梱包設計者とともに海外製造拠点の現場視察と調査を行い、問題点を指摘する。次に、これらの問題がどのような理由で発生したかを指摘するとともに、対策を現地梱包設計者と一緒に導き出す。問題の発生要因が梱包設計にある場合には、梱包設計ガイドラインの内容を講義し、演習としてケーススタディを用いることで梱包設計技術の定着化を図る。

こうした活動を通じて海外製造拠点ごとに梱包設計キーマンを育成することで、海外拠点自らが梱包改善活動を推進するとともに、現地特有の問題にも対応できる体制を構築する。

## 6. むすび

従来の日本国内を中心に考えていた物流からグローバルな物流に移り変わったとしても、梱包設計の観点ではやるべきことに大きな差異はない。

梱包設計改善で得られる物流費の削減効果は、大規模な投資がなくても知識があれば得られることが多い。したがって、当社グループ企業内の改善事例を共有して、梱包設計技術の相互展開を推進している。このような活動を継続し、2020年度までに当社グループ全体の経営目標達成に貢献していく。

## 参考文献

- (1) 松島大輔：空洞化のウソー日本企業の「現地化」戦略、講談社現代新書（2012）
- (2) EYアドバイザリー編：サプライチェーンマネジメントの理論と実践、幻冬舎（2014）
- (3) 北原敬之：日系自動車部品サプライヤーの海外における開発・設計の現地化に関する一考察、早稲田大学日本自動車部品産業研究所紀要、7、15~25（2011）  
[https://dspace.wul.waseda.ac.jp/dspace/bitstream/2065/35785/1/NihonJidoshiaBuhinSangyo\\_7\\_Kitahara.pdf](https://dspace.wul.waseda.ac.jp/dspace/bitstream/2065/35785/1/NihonJidoshiaBuhinSangyo_7_Kitahara.pdf)



図7. 現地スタッフ教育の様子(タイ)