Vol.2022-IS-160 No.6
2022/6/4

システム開発超上流のモデリングチャレンジ

嶋津恵子[1]　坂本啓[2]　山崎政彦[3]
芝田朋世[4]　濵下宙千[4]　湯浅廣之[5]

概要：本稿は，2022 年度科学研究費補助金基盤研究（B）に採択された「すり合わせ技術を利用したモデルベース全体最適化システム設計手法の開発」(3 年直接経費総額 13,400 千円)の，現状までの調査結果を報告するものである．我々は，情報システム開発における問題解決に至る道筋を描けぬまま，なし崩し的なシステム実装を行い，運用開始後に問題が続発する例が後を絶たない問題に着目した．この解決策として開発の初期段階で描く運用後の状態をモデル化することと，その際“横型すり合わせ”を利用することで解決に至る可能性があると論じた．

キーワード：システムズエンジニアリング，MBSE，アーキテクチャ，設計，すり合わせ

Model-Based Design Methodology for Globally Optimized System Using Concurrent Adjustment: a preliminary survey

KEIKO SHIMAZU[†1]　HIRAKU SAKAMOTO[†2]
MASAHIKO YAMAZAKI[†3]　TOMOYO SHIBATA[†4]
MICHI HAMASHITA[†4]　HIROYUKI YUASA[†5]

Keywords: Systems Engineering, MBSE, Systems Architecture, Systems Design, suriawase

1. はじめに

我々は，システムズエンジニアリング・プロセス標準（ISO/IEC/IEEE 15288）やそれを基盤にしている MBSE (Model Based Systems Engineering)に準拠したシステム開発の事例研究に従事してきた．国産の GNSS (Global Navigation Satellite System : 測位衛星システム)である QZSS (Quasi-Zenith Satellite System : 準天頂衛星システム) を利用した災害急性期警報発令システムの設計にこれらを導入し，またこれらを実習型で指導する授業開発などを行ってきた[1, 2 他]．これらの活動は，日本の産業界に冒頭の標準と MBSE を展開することが狙いであり，志を同じくする国内の研究者や実践者らの活躍で徐々に効果が出始めている．

その一方で，日本の情報システム産業界には，長年に渡り解決できていない問題がある．我々はその問題に対し，システムズエンジニアリング･プロセス標準（ISO/IEC/IEEE 15288）やそれを基盤にしている MBSE を応用することで解決することを狙っている．本稿は，2022 年度科学研究費補助金基盤研究（B）に採択された「すり合わせ技術を利用したモデルベース全体最適化システム設計手法の開発」(3 年直接経費総額 13,400 千円)の，現在までに行った調査結果を報告するものである．

本書は，次の構成を採る．2 章に情報システム開発における普遍的課題を整理し，3 章で問題の本質を論じる．そして，4 章に問題解決のための前提として導入が必要だと我々が考える MBSE において，設計に用いる構造表現用モデル言語の変遷を整理する．さらに 5 章で，異なる複数の観点で作成したモデル群を矛盾なく整合させるためのフレームワークの代表的な例を示し，続く 6 章にこれらの日本の産業化への導入状況を概観する．そして 7 章に本研究の到達に向かうアプローチを解説し，我々の考案する“横型すり合わせ”を紹介する．

2. 情報システム開発における普遍的課題

社会，特に産業界では，問題解決に至る道筋を描けぬまま，なし崩し的なシステム実装を行い，運用開始後に問題が続発する例が後を絶たない．

日経コンピュータの特集記事は，銀行システム開発における設計以降の工程で発生する手戻り作業を集計し，これらの解消で実際の半分のコストで済むことを示した．そして，要因である莫大な手戻り作業の発生原因は，開発すべきシステム像を特定する段階でユーザらの意向を曖昧にし，そのまま詳細設計に移行，実装を行ったことにあるとしている(図 1)[3]．この問題に対応する方法として，1990 年代終盤からアジャイルが台頭し急速に導入された．現在，状況はどの程度改善しただろうか．

1 都立産業技術大学院大学
Advanced Institute of Industrial Technology
2 東京工業大学工学院
Tokyo Institute of Technology
3 日本大学理工学部
Nihon University
4 東京都立大学
Tokyo Metropolitan University
5 横浜国立大学
YOKOHAMA National University

2019 年の記事“動かないコンピュータ”に，国内 380 人のシステム開発者に対するインタビューの集計結果が掲載された(図 2) [4]．現場に導入されたシステムが，設計通りに実装されたにもかかわらず当初の期待に応えていない事例の調査報告である．回答者の半数以上がこの事例ありと回答し，その原因は開発すべきシステム像を特定する段階で，曖昧性や矛盾を残したまま詳細設計に移行したことにあるとしている．

この状況だけをみると，アジャイルは効果を発揮しなかったような印象を与えるが，それは誤解であると申請者らは考える．ウォーターフォール型開発の「前工程の出力を次工程の入力情報として厳格に進める方式」を見直し，アジャイルは仕様として把握できた部分から実装・テストすることを繰り返すことで，工程間の移行のリードタイムを最小に抑え，エンドユーザからの新たな要望にも即座に応えることを可能にした．

一方，把握できた部分の仕様がシステム全体のそれに及ぶことはほとんどなく，操作画面部分や検索エンジン部分など，部品内に影響は限定される．つまりアジャイルは部分最適化に高い効果を発揮する手法であるが，システム全体としての品質を保証する手法ではない．

ところが顧客らは，システム化対象領域を「どういうシステムを開発すれば，現在の問題を解決できるのか」として特定する作業に長期間要することを嫌い，アジャイルの利用を歓迎する[5]．結果として，部分最適手法であるアジャイルで製造したシステム部品の繋ぎ合わせが繰り返され，完成間近もしくは現場導入後に全体としての不具合が判明する事態が，頻発していると言える．

部分最適化作業の結合によるシステム設計が，致命的な不具合を誘発した代表的な例に EMIS (Emergency Medical Information System)がある．EMIS は阪神大震災 (1995 年 3 月) を機に，大規模広域災害が発生した際に，都道府県を横断し救命病棟の受け入れ状態を共有し，医療資源を効率的に運用することを目的に開発された．当時の最先端技術が随所に導入された一方で，災害医療や救命行為で本来必要としている作業などへの影響を見逃したまま現場導入した．これにより東日本大震災発災時(2011 年 9 月)に，システムの操作のために救命作業を優先できない事態が多発し，想定通りの活用に至らなかった．部分最適化で作られたシステムは完成後の改修で全体最適化を実現することは困難であり，継ぎ接ぎ作業が繰り返される．

EMIS では，このための予算が毎年計上され，平成 31 年度は 2.8 億円，そして令和 2 年度は 5 億円を超える[6, 7].

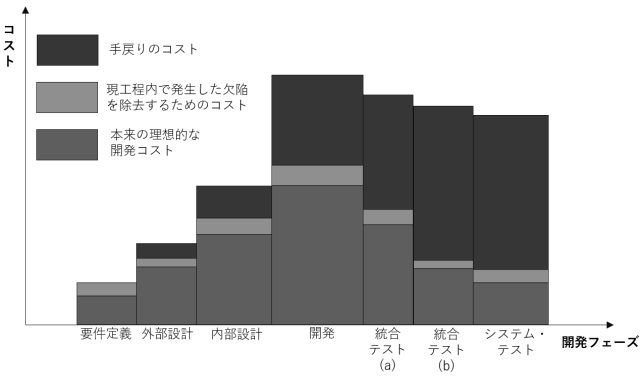

図 1　銀行システムの開発における手戻り発生例（転載）[3]

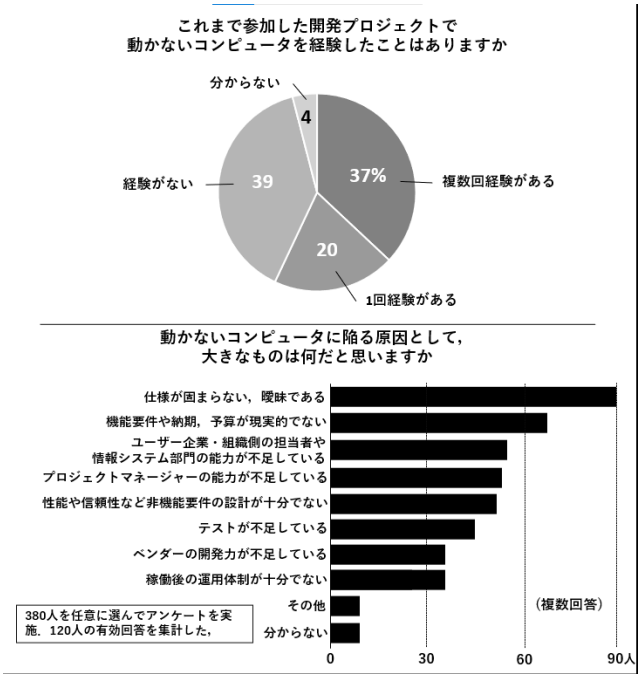

図 2　設計通りに実装したシステムが期待に応えていない（転載）[4]

3. 問題の本質はどこにあるのか

前章で述べた問題が発生する原因は，システム設計の前提として用意されるはずのシステム導入後を描いた運用現場の設計が，システム設計に反映されないことにあると考えた．その理由は，システム設計が（モデルの利用等で）構造的に表現されるのに対し，システム導入後を描いた運用現場の設計は，ポンチ絵や文章など非構造で表現されることによるものではないかと考えた．

設計の本質は何かに対し，2017 年に日本設計工学会が開催した第 26 回 設計オープンセミナーで，Autodesk Inc.の主席研究員であるマイケル・ベルギン氏は，設計とは形状を探し出すことだと説明した．これを反映するように，情報システム開発における設計には，その表現方法として，ER（Entity Relationship）や DFD（Data Flow Diagram），そ

して IDEF や, さらにオブジェクト指向型の UML や SysML などが採用されてきた．つまり，これらを表現方法として利用する設計結果は構造として表現される．本書では，この表現結果をモデルと記す[8-12].

ここで物理モデル(Physical model) として分類されるモデルは，実装や製造に直結する設計であり，このモデル通りに製造/製作すると望む対象物が実現できる．日本の情報システム開発の多くの現場ではこれを詳細設計と称し，そして機械システム開発の現場では製造設計と呼ぶことが多い．ところで，この実装や製造に直結する設計書/図の作成の前段階として，望む対象物が具備すべき必要な機能（および性能）を特定し，物理設計と同様に構造として表現しておく．これを機能モデルや論理モデルとして用意する．

つまり物理モデルを作成（物理設計）する方法は，構造表現化した機能モデルを，同じく別の構造表現化した物理モデルに変換する作業となる．この時機能モデルは，最終的に完成するシステムに望む機能を構成要素として漏れなく論理的に配置することが目的であり，物理モデルは，それら機能を実際に動作させるための実装/製造方法を特定の部品（とインタフェース）に配置した結果である．機能モデルは,（現実に存在する部品に依存せず）論理上成立するモデルである．物理モデルは，これら機能を動作させ現存する（もしくは新たに製造する）部品群の選択となるため，両者の構造体の体形は異なることが多い．また，開発対象の複雑性や特性によって，機能モデルと物理モデルの間に配置設計モデルや構造設計モデルなどが必要となることがある．いずれにしても，これらすべてはモデルであり構造として表現される．

以上のことから，機能モデルが（性能も含め）望むシステムを機能として過不足なく網羅され全体最適化が図られていれば，それを正確に変換した結果である物理モデルもその状態になっているはずであり，前章に述べた問題は発生しない．

つまり，前章の問題の原因は，機能モデル作成の段階ですでに内在していたことになる．

では，機能モデルはどのように作成するのであろうか．

システムズエンジニアリング・プロセス標準（ISO/IEC/IEEE15288）を実践的に利用する方法を解説した Systems Engineering Handbook によると，まず，システム導入後の利用者の状態を表現したモデルを作成し，その構造と不整合を起こさないように機能モデルに変換することが肝要とされている[13]．このシステム導入後の利用者の状態を表現したモデルは運用モデル（Operational Model）と称されることが多い．つまり，機能モデルの前提となる運用モデルの段階で，必要となる事柄が過不足なく網羅され全体最適化が図られ，それを機能モデルに変換する作業が機能設計の本質となる．図 3 は，システムズエンジニアリングの産業界と学界の横断的コミュニティである INCOSE(INternational Council On Systems Engineering)の公式インストラクターである John Clark 氏が講習用に利用している資料の抜粋である．左から運用モデルのイメージ，機能モデルのイメージ，そして物理モデルのイメージが並んでおり，これらの整合を取ることが示されている．

実際，日本の情報システム開発現場では，この作業をおこなっていることが少ない．なぜ，日本の情報システム開発の現場では, 望む運用状態をモデルとして構造表現化し，それと整合する機能モデルを設計するという作業を実施しないのであろうか.

4. 構造化表現用モデル言語の進歩

前章に一部紹介したモデル言語には，システムを特定の観点で表現した結果を構造化することが目的であるものと，異なる複数の観点で構造化表現した結果を全体として整合を図ることを目的とするものがある. 特に後者に関しては，該当するモデル言語の利用方法をフレームワークとして整備している例もある.

ER モデルは，ソフトウエアシステムが取り扱うデータの構造化を，そして DFD は，同システムで加工や編集され

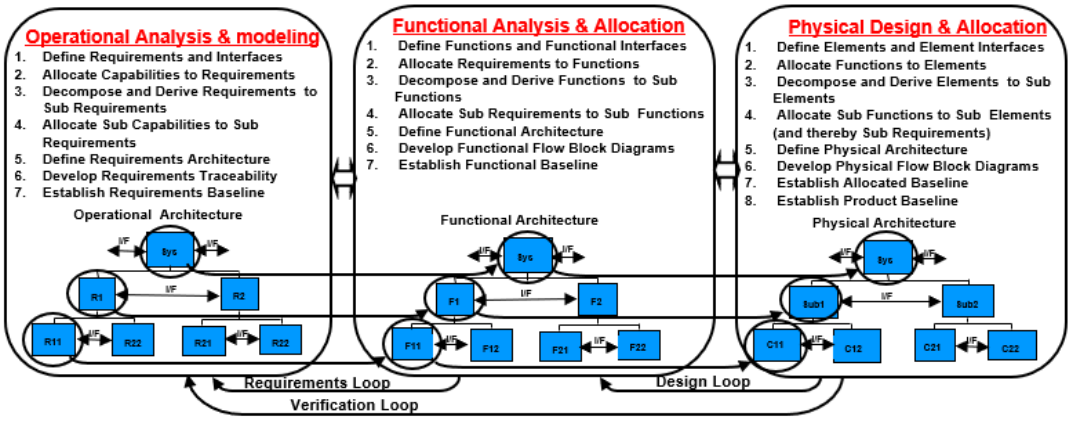

図 3　運用モデルと機能モデルと物理モデルが矛盾なく整合している状態のイメージ図（部分転載）

るデータの変遷を，それぞれ表現することを目的に開発された．

一方, UML はソフトウエアの設計用に開発されたモデリング言語体系であり，特定の異なる複数の観点に立って設計した結果を，整合性と全体最適化を図る方法として提案された．図 4 は，UML 開発者の論文に掲載された図の転載である．左上にある Logical View が現在の機能モデルを作成する際の観点に相当し，右上の Development View と右下の Physical View が現在の実装モデルを作成する観点に相当する．そして，中央の楕円で示された Scenarios（ユースケース・シナリオ）が現在の運用モデルを作成する観点になる．現在最も利用されている版である UML2.0 では，ユースケース図(Use case Diagram)が用意され，これを使って表現されるユースケースは，複数のユースケース・シナリオの集合として設計される．最も重要なことは，異なる観点で設計された結果が, 運用モデルである Scenarios と整合するように調整（本稿では“すり合わせ”と表現）する点にある．異なる観点で作成されたすべてのモデルを，運用モデルと整合させることで，結果的にすべてのモデルを横断して整合が図られる． そして SysML は，ソフトウエア専用のモデリング用言語である UML をシステム設計用に拡大したものであり, 現在 MBSE 用の世界標準になっている．

さらに IDEF は 0 から 14 までの異なる観点でのモデル表現を用意し，システム全体を包括的に表現できるように配慮されている．

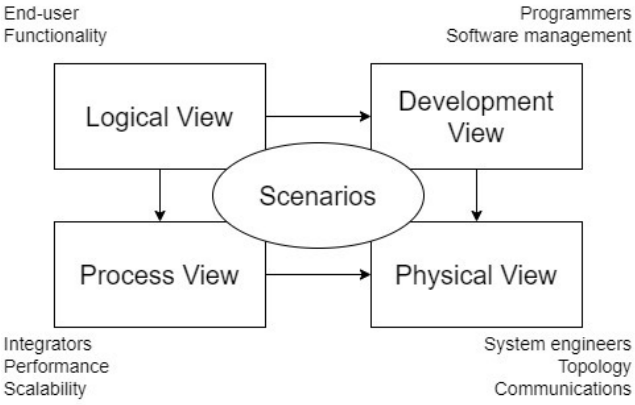

図 4　複数の異なる観点で作成したモデルを運用モデルに矛盾なく整合させる UML の狙いを示した図（転載）[11]

5. 全体最適化用フレームワーク

異なる観点で設計した結果（モデル群）同士の整合をとり，それにより全体最適化を実現するためのフレームワークも用意されている．フレームワークを利用することで，採用するモデル言語体系に用意された表現形式に拘束されることなく，全体最適化を図ることができる．

システムズエンジニアリング・プロセス標準（ISO/IEC/IEEE 15288）によるとシステムの基盤となるシステム・アーキテクチャは，異なる複数の観点で設計した結果（モデル群）の集合であり，かつこれらが矛盾せず整合している[14]．この異なる観点で作成したモデル群の整合をとりアーキテクチャ設計を行おうとすると，UML，SysML，そして IDEF などのモデル表現体系に用意されている表現形式を観点ごとに採用し，人手によって整合を図ることになる．一方，それぞれのモデル言語が用意する表現体系に依存せず，様々なモデル言語を利用してこれを行う方法が，フレームワークの利用である．

1987 年に発表されたザックマン・フレームワーク(Zachman Framework)は，情報システム開発を対象としたものであり[15]，1990 年に検討がはじまり現在も改良と更新がおこなわれている DoDAF は，広くシステム全般を視野に置いたものである．これらの他にも DoDAF の派生形や相互に影響を与えあったものに IEEE 1220，TOGAF，FEAF (Federal Enterprise Architecture Framework)などがある[16, 17].

特に，DoDAF は，モデリングをする際の異なる複数の”観点”群を 2 つ（View と Viewpoint）に分割し，さらにそれらを詳細化していることに特徴がある. View は開発するシステムそのものをどの観点で設計するかであり，Viewpoint は開発するシステムを取り巻く環境のうちのいずれかの観点である．

6. 日本での試みと世界標準の導入状況

前章までに示した状況に対し日本独自の試みや，全体最適化用のフレームワークの導入の実績がいくつかあるが，現在までのところ有効に機能しているとは言い難い．

具体的には，システム設計の全体最適化に注目した日本独自の動きとして 1DCAE を挙げることができる. これは，日本のモノづくりの世界からの遅れが，顧客の多様な要求に迅速に対応できていないことにあるとし，設計の冒頭の工程に注目し, 機能設計とそれ以降の工程で, QFD (Quality, Function, Deployment)や FMEA (Failure Mode and Effects Analysis)などシステムズエンジニアリング・プロセス標準（ISO/IEC/IEEE 15288）の手法に繋げようとしている(図 5)[18].

一方，1DCAE は，設計の冒頭の工程で情報を構造化する手法の具体的検討は行っていない．図 5 を参照すると，設計の工程を概念から製造までの 5 つに分け，それぞれで発想，分析，評価，そして可視化の設計行為を行う．これらの中で，分析と可視化で構造化手法を採用する．1DCAE では，概念設計の工程で CVCA (Customer Value Chain Analysis)のような新しい手法を採用しているが，システムズエンジニアリング・プロセス標準（ISO/IEC/IEEE 15288）の実践で実績のあるコンテキスト分析に代表される可視化

の領域は網羅していない．つまり， 1DCAE では概念設計の工程で情報の構造化（モデリング）を行わないまま，機能設計に移行していると理解できる．

以上のことから，1DCAE が提案する方法では，依然として望む運用状態をモデルとして構造表現化し全体最適化を，システム開発の当初から行うという作業を後回しにし，問題発生の先送りを是認している．もしくは CAD/CAM による製造プロダクトに手法利用を限定し，発生する問題を一定程度に留めようとしていると考えられる[19].

また，全体最適化用フレームワークの利用例としては，ザックマンのそれは，日本でも情報システム開発業界に古くから知られているが，導入の実績がほとんどないことが報告されている[20]．そして，DoDAF の影響を受けた陸上自衛隊の「陸自版アーキテクチャ構築要領書」がある[21]．ところが，この要領書が実際のシステム開発に利用されている実績は見当たらない．

フレームワークの利用が広がらない理由を，井手らは①新たな仕様記述法の習得の必要性と，②フレームワーク導入による開発の初期作業の工数増加と，さらに③これら 2 点による予算の増加があげられると報告した[22]．これは本稿 2 章で述べた，“顧客らは，システム化対象領域を「どういうシステムを開発すれば，現在の問題を解決できるのか」として特定する作業に長期間要することを嫌っている”という観察結果と一致する．

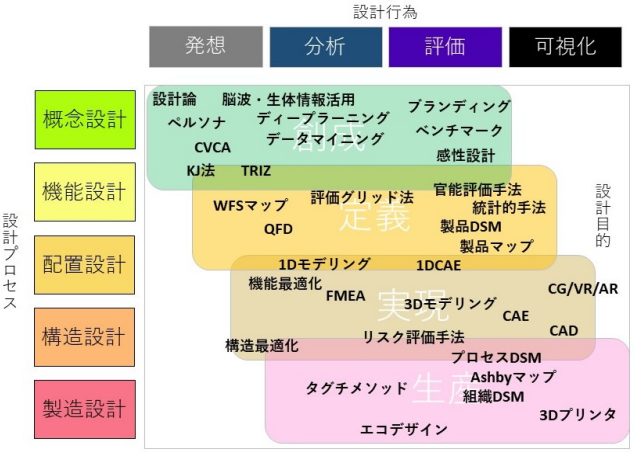

図 5　1DCAE（転載）[18]

7. 解決策へのアプローチ

我々は，日本の産業界で高い実績を上げた“すり合わせ”導入により，複数の観点で設計した結果（モデル群）を矛盾なく整合させ，システム・アーキテクチャと成立させることが可能になると考える．

日本企業は欧米に比べてすり合わせ技術に秀でているという主張は広く知られている[23]．日本の産業界は，1950 年代から高度成長期にかけて製造部門で培った小集団でのすり合わせ作業による部品品質向上(TQC 活動)の成功体験があり，最近のアジャイル導入を自然なものとし，一気にこれを広げたと考えられる．ところが，開発するシステムが大規模複雑化した現在，この方法は期待通りには成功していない[24]．むしろ，部分最適化手法を支えるすり合わせ力を，システムの規模を問わず利用していることで，システム設計の全体最適化にとって致命的な障害を内在化させている．

一方，近年提案された新しいすり合わせ手法が“横型すり合わせ”である[25]．日本が実績を上げてきた技術は縦型すり合わせであり，高度成長期に発展した TQC の導入による製造現場での小集団活動や，ソフトウエア開発現場の数名によるアジャイルプログラミング手法が相当する．いずれもシステム開発における同一のミッションや役割を持った者同士による最適化作業である．これに対し横型すり合わせは，ミッションや役割の異なるステークホルダ間の連携を目指すものであり，インタネットの台頭等により組織間・国家間の距離が変化したことで実現したとされる．我々は，この横型すり合わせを，システム導入後を描いた運用現場の設計に導入する方法を開発したいと考える．これにより，運用を支える複数の観点で設計した結果（モデル群）を矛盾なく整合させることに利用でき，システム・アーキテクチャを効率的に設計することが可能になる．結果，前章に示した 3 つの問題も解消できると考えた．

とはいえ，異なる複数の観点で構造的に設計（モデリング）し，それらを矛盾ない状態に整合させることでシステム・アーキテクチャとなるという，システムズエンジニアリング・プロセス標準（ISO/IEC/IEEE 15288）の基本を実践的に利用できる状態でないと，フレームワークの採用そのものの価値が理解できず，高い効果は望めない．本章で述べた手法の開発を目指すとともに，基本，特に MBSE を多くの実践現場に導入する活動も継続して必要であると考える．

8. おわりに

本稿では，2022 年度科学研究費補助金基盤研究（B）に採択された「すり合わせ技術を利用したモデルベース全体最適化システム設計手法の開発」(3 年直接経費総額 13,400 千円)の，現状までの調査結果を報告した．我々は，情報システム開発における問題解決に至る道筋を描けぬまま，なし崩し的なシステム実装を行い，運用開始後に問題が続発する例が後を絶たない問題に着目した．問題の原因が，システム設計の前提として用意されるはずのシステム導入後を描いた運用現場の設計が，システム設計に反映されないことにあると考え，システム設計が（モデルの利用等により）構造的表現が採用されているのに対し，運用現場の設計がそうでないことに注目した．一方，異なる観点で作成された複数のモデル群が互いに矛盾なく整合させる方法と

してのフレームワークが市場には整備されているにも関わらず，それらが効果的に利用されていないことから，運用現場の設計にもモデル表現を導入するだけでは解決できないと推察した．そこで我々は，“横型すり合わせ”手法を開発することで本稿で特定した問題解決に貢献できると期待する．

謝辞 本研究は，2022 年度科学研究費補助金基盤研究（B)「すり合わせ技術を利用したモデルベース全体最適化システム設計手法の開発」によるものです．

参考文献

[1] 坂本啓，山崎政彦．「衛星キットを活用したシステムズエンジニアリング講義の事例報告」．第 62 回宇宙科学技術連合講演会，2018−10−24，No. 1K05.
[2] 嶋津恵子，五十嵐由衣，青羽真利，大島真言，平岡恵里，及川航平，湯浅廣之，芝田朋世．QZSS（準天頂衛星）を利用した早期警戒放送システムのシステム・アーキテクチャ機能モデルと物理モデル．情報処理学会第 209 回ソフトウェア工学研究発表会，2021.
[3] 日経コンピュータ．日経 BP 社, 1998-09-28, p.118-134.
[4] 日経コンピュータ．日経 BP 社, 2019-07-25, p.92-93.
[5] DX レポート 2.1．経済産業省, 2021-08-31.
[6] 厚生労働省医政局．資料 1 - 1．第 65 回社会保障審議会医療部会, 2019-01-17.
[7] 厚生労働省医政局．令和 3 年度予算・税制改正について, 2021-4-5.
[8] Peter Pin-Shan Chen．The entity-relationship model—toward a unified view of data．ACM Transactions on Database Systems, March 1976, vol. 1, no. 1, p.9-36.
[9] W. Stevens,G. Myers,L. Constantine. Structured Design．IBM Systems Journal, 1974, 13 (2), p.115-139.
[10] https://web.archive.org/web/20070126094915/http://www.stsc.hill.af.mil/crosstalk/1995/06/IDEF.asp,（参照 2022-04-20）.
[11] PhilippeKruchten. Architectural Blueprints—The “4+1” View Model of Software Architecture. IEEE Software 12 (6), November 1995, p.42-50.
[12] https://www.omg.org/hot-topics/syse.htm,（参照 2022-04-20）.
[13] Systems Engineering Handbook, A guide for system life cycle processes and activities．fourth edition, 2.2.3 General System Methodologies, Wiley, 2015.
[14] Systems and software engineering — System life cycle processes．ISO/IEC/IEEE 15288. 6.4.4.1, 2015.
[15] J.A.Zachman. A framework for information systems architecture．IBM Systems Journal, 1987, vol. 26, no. 3, p. 276-292.
[16] The TOGAF® Standard．Version 9.2 11th Edition, Van Haren Publishing.
[17] https://obamawhitehouse.archives.gov/omb/e-gov/FEA, (参照 2022-04-20).
[18] https://1dcae.jp/about/,(参照 2022-04-20）.
[19] https://www.1dcae.org/,(参照 2022-04-20）.
[20] 森本祥一，渥美幸雄．エンタープライズ・アーキテクチャにおける成果物の検証技法．専修大学情報科学研究所所報，2011-02-20, 75, 23-32.
[21] 陸上自衛隊研究本部．陸自版アーキテクチャ構築要領書．2010-03-10，ver3.0J.
[22] 井手達夫，平本健二．総合的な安全保障体制の在り方―全体最適化計画手法（アーキテクチャフレームワーク）の観点から―．国際安全保, 2012−3，vol.39, no. 3, p.28-39.
[23] 藤本隆宏．日本の現場力(トヨタ方式、TQC、カイゼン)．中小企業庁, 2007-07-15.
[24] 柴田友厚．「すり合わせ能力」は「弱み」に変わる．日経 Tech, 2013-11-11.
[25] 経営センサー．わが国の強み「すり合わせ」を活かす．東レ経営研究所，2016-6, p.35-41.