第 13 回横幹連合コンファレンス
2022.12.17-18　早稲田大学国際会議場

GPDAC--Goal, Process, Data, Actor, Control による知の統合

○山本　修一郎（名古屋国際工科専門職大学）

Knowledge Integration by GPDAC: Goal, Process, Data, Actor, and Control

* Shuichiro Yamamoto (IPUT in Nagoya)

Abstract－ There is an effort to collect successful cases of DX initiatives. However, it has not progressed to develop a DX theory that generalizes the results of problem solving by DX. In this paper, we propose the GPDAC (Goal, Process, Data, Actor, Control) as a framework for creating new DX knowledge by recombining knowledge from different academic fields.

Index terms－ Knowledge Integration, GPDAC, OPM, Systemigram, ArchiMate

1. はじめに

横幹知で推進する DX 調査研究会 [1][2]では，横幹連合として，異分野知識が共生する研究会活動を通じて，社会へのデジタル技術の進展が知にもたらす影響と，知の共創の姿を探求している．

本稿では，本調査研究会で取り組む研究課題について，山本による GPDAC(Goal, Process, Data, Actor, Control)について解説する．

以下では，2 節で DX と問題解決，3 節で知の共生について述べる．4 節でデータ駆動工程設計法に基づいて知の共創構造を明らかにする．5 節で GPDAC を提案し，Systemigram, OPM, ArchiMate との関係を具体例で明らかにする．6 節で考察を述べ，7 節でまとめと今後の課題を明らかにする．

2. DXと問題解決

我が国の DX が，諸外国と比べて遅れている理由として，DX 人材不足が指摘されている [3]．経営，事業，デジタル技術に精通した「やたがらす人材」が DX を先導したという報告がある [4]．

データとデジタル技術を活用することにより，企業の具体的な問題を解決して，競争力のあるデジタル企業に変革することが DX である．しかし，我が国の企業経営者には，「D（デジタル）は分かるが，X（変革）が分からない」という声が多いようである．手段（解決策）によって，「問題状況」を「問題が解決された状況」に変換することが，問題解決の根本構造である．

「デジタル技術により，課題のある社会を，課題が解決された社会」に変換することが社会的 DX である．また，「デジタル技術により，課題のある企業を，課題が解決された企業」に変換することが，企業の DX である．問題＝既存企業の問題，あるべき姿＝デジタル企業，解決策＝変革，手段＝デジタル技術だと考えると，デジタル技術という手段は分かるが，解決策が分からないのは，企業の問題が分からないからということになる．つまり，DX が分からない真の原因は，問題解決の根本構造を，これまでの学校教育で学習していないために，企業の問題が明確にできない点にある．

手段としてのデジタル技術や DX 事例をいくら勉強しても，DX が成功することはない．企業や社会の問題を認識する能力がないと，DX 人材だけでは DX は進まない．必要なのは，DX 人材育成ではなく，この問題解決構造を理解する人材の育成である．

企業の問題が分からないもう一つの理由は，「悪いことは起こらない」「これまでの延長に未来がある」という正常化バイアスである．したがって，この正常化バイアスの下では現状に問題がないことになるから，変革する理由がないというわけである．これでは，デザイン思考をいくら学んでも，現状に問題はないのだから，新しい価値を生むデジタル製品やサービスのアイデアが生まれることはない．また，業務変革では，本来あるべき姿をどうするかを考えることができず，現状の課題を解決するアイデアしかでないという問題がある．

3. 知の共生

横幹連合の英語名は，Transdisciplinary Federation of Science and Technology である．Transdisciplinary の関連用語として，Multidisciplinary と Interdisciplinary がある．この 3 語は，以下のように区別される [5][6]．

【マルチディシプリナリ】学問分野の境界を横断して新たな知識を創造することなく，異なる分野の学問知識を連携する
【インターディシプリナリ】異なる問題を共通の枠組みで解決することを目的として，異なる学問分野の知識領域や個別学問分野の範囲を超えて専門用語や方法論を統一することによって，知識の新たな連携を創造する
【トランスディシプリナリ】
現実社会の具体的な問題解決を目的として，異なる学問分野における知識の相互作用による知識の再結合に基づいて，新たな知識を創造する

DX のためのマルチディシプリナリ知識の例は，知識体系である．たとえば，筆者による大学一年生に教育しているデジタルデザインエンジニアリング知識体系 [7]では，①価値層（要求工学，イノベーション理論，デザイン思考，社会システム論等），②ビジネス層（カスタージャーニー，ビジネスシナリオ，プロセス，サービスブループリント等），③システム層（対話デザイン，安全性，セキュリティ，アーキテクチャパターン等），④物理層（CPS）から階層的に構成している．DX のためのインターディシプリナリ知識の例は，

知識表現モデル間の変換である [8)9)10)11)12)]．SysML，Systemigram, ArchiMate,ものこと分析図を対象とする相互変換法の構築が期待される．

Systemigram の起源は，Checkland によるソフトシステム方法論（SSM, Soft System Methodology）[13)]で用いられたシステムモデル図にある．複雑な人間活動を分析するために考案された方法論が SSM である．人間活動システムに含まれる個々の活動は「動詞」で表現できるから，人間活動システムを定義するためには動詞間の「結合性」も表現できる必要がある．Boadman は，システムモデル図が自然言語表現と対応しやすいことに着目して，より明確に自然言語による文章と対応する図式として，Systemigram を提案した [14)]．最近では，複雑なシステムを分析するために，Systemigram が利用されている．たとえば，エンタープライズアーキテクチャや SoS(System Of Systems)の複雑性を分析するために Systemigram が用いられるようになっている [15)]．

ArchiMate は，The Open Group が標準化する EA モデリング言語である [16)17)]．ArchiMate コアフレームワークでは，ビジネスアーキテクチャ(BA)，アプリケーションアーキテクチャ(AA), テクノロジーアーキテクチャ (TA) からなるアーキテクチャ階層と，Active Structure，Passive Structure，Behavior，Motivation 要素からなる Aspect が定義されている．TOGAF(The Open Group Architecture Framework)[18)] の Architecture Development Method（ADM）の全工程の成果物を ArchiMate で記述できる．

「学を超えて社会と連携した研究活動」が「トランスディシプリナリ研究」である [6)]．「トランスディシプリナリ研究」は，複数の学問分野の知を統合して新しい社会価値創出につなげる考え方である．

横幹知で推進する DX 調査研究会は，2022 年 4 月から 2024 年 3 月までの予定で発足した．この研究会の活動では，参加していただいている各学会の先生方と「複数の学問分野の知を統合して，新しい社会価値創出につなげる DX の考え方」を探求している．まさに，この活動こそが「トランスディシプリナリ研究」の実践である．

また，山本は，中部品質管理協会で「2030 年の質価値創造研究会」で企業の参加者とともに，SDGs，DX，QC の知を統合して新しい価値創出を目指す活動 [19)20)]を進めている．この研究もトランスディシプリナリ研究である．この研究会の活動の中で，入出力データの関係からプロセスを導出するデータ駆動工程設計法 [21)]の着想が得られた．

4. 知の共創構造

デジタル技術を活用した問題解決の実践と，その結果だけでなく，そこから生まれる理論としての統合知が新たな実践を生む反復構造の解明に取り組む．たとえば，DX による問題解決には Fig.1 に示す構造がある．

この図には，実践と結果からなる第 1 の反復構造と，結果から理論を導き，それを実践して新たな問題解決を実践する第 2 の反復構造がある．第 1 の反復構造では，結果に基づく改善により，実践手法を最適化する．既存の実践手法では問題解決できない場合，第 2 の反復構造を用いて，新たな問題解決のために，結果から理論にまで遡って，実践手法を創造する．また，複数の実践を集めて分類するパターン化と，複数の理論を組み合わせる統合化がある．

現状では，DX の実践事例を収集する取り組みがある [22)23)]．しかし，問題解決結果を一般化する DX 理論の構築までは進んでいない．今後，DX 理論の構築が期待される．以下では中小製造業における DX 事例とデータ駆動工程設計法との関係について紹介する．

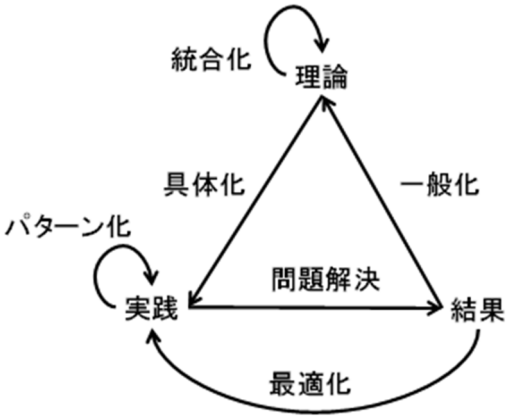


Fig.1 Problem Solving Structure

4.1 データ駆動工程設計法

筆者らは入出力データの依存関係分析に基づく工程設計法を提案した [21)]．設計対象となる業務の契機となる最初の入力データと最終的な出力データの依存関係に基づいてデータフロー図に変換することにより，入力データから出力データを作成するために必要な工程を設計できる．

BPR(Business Process Reengineering) の起源は，Hammer & Champy による Reengineering the Corporation [24)] である．Hammer & Champy は，この書籍の表紙で，「ビジネスがどのように機能するかあなたが知っていることを忘れなさい．ほとんどが間違っているから」と述べている．すなわち，BPR の本質は既存プロセスの改善ではなく，変革にあった．

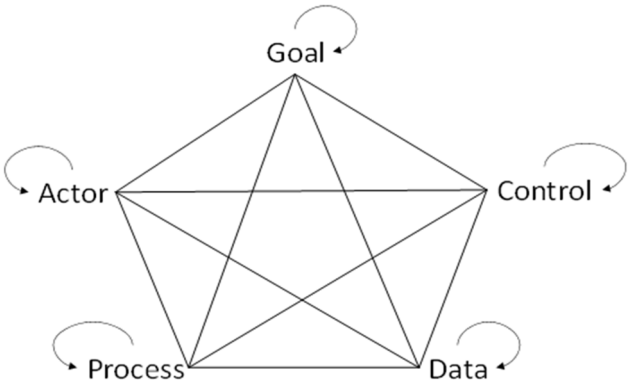
データ駆動工程設計法でも，業務を構成する詳細な手順を分析するのではなく，行の外部にある入力と最終的な出力だけに着目することで，詳細な内部プロセスを忘れることができる．これに対して，従来の業務改善では，既存の業務プロセスを調べてから，業務の無駄や重複を識別してプロセスを改善する．この場合，次のような問題がある．業務プロセスが曖昧・複雑な場合には，業務プロセスの調査に時間がかかるだけでなく，調査した業務プロセスが正しいかどうか分からない．また業務プロセスの曖昧性・冗長性の摘出には経験が必要であり，属人的な摘出結果になる可能性がある．

業務改善では，実践結果に基づく最適化を反復する．Fig.1 の問題解決・最適化サイクルに対応している．これに対して，データ駆動工程設計法は，結果として

の最適な業務プロセスを設計する理論として，Fig.1 の具体化・問題解決・一般化サイクルに対応する．

5. GPDAC

現行企業の具体的な問題解決を目的として，異なる学問分野知識の相互作用による知識の再結合によって，新たな知識を創造する．たとえば，ヒトと，AI, IoT, ロボットなどのデジタル技術との共生課題を考えるための枠組みとして GPDAC (Goal, Process, Data, Actor, Control)を要素とするフレームワーク（Fig.2）を考えることができる．GPDAC は，異分野の知識を再結合する点でトランスディシプリナリ知識の例である．



Fig,2 GPDAC Framework

GPDAC では，顧客や企業と，デジタル技術（AI,IoT,ロボットなど）からなるアクタが共生するためのゴール，プロセス，データ，品質保証についての知識とその関係を明らかにするとともに，異なる知識間の関係を分析できる．異なる理論であっても，DX を扱うのであれば，GPDAC を共通要素として含むと考えられる．したがって，GPDAC を接点として用いれば，異なる理論間を関係づけ組み合わせることにより，新たな理論を創造できる．

また，GPDAC を用いて知の統合の在り方を示す羅針盤を構成できる可能性がある．たとえば，GPDAC を軸とする 5 次元空間を考えることにより，各軸におけるデジタル技術との共生度を測定することにより，DX による問題解決の成熟度を可視化できる．

社会や企業を変革する技術としての DX と，変革された結果とを総合してあつかうことにより，DX についての横幹知・総合知を構築することができると期待している．

以下では，異なる 3 分野の代表的な図式モデルに対して GPDAC との関係を明らかにする．すなわち，システム思考分野の Systemigram，システム工学分野の OPM(Object-Process Methodology)，エンタープライズアーキテクチャ分野の ArchiMate に対して GPDAC を考える．

複雑なシステムを 1 つのモデルで記述するために，オブジェクト指向とプロセス指向のパラダイムをシステム思考に基づいて統一するために，Dori が OPM を提案した [25][26][27]．OPM では，オブジェクトを矩形，プロセスを楕円で表現して同じ図の中で記述できる．

5.1 モデルの記述比較

まず Li ら [28]による航空機設計に対する OPM の記述に対する Systemigram と ArchiMate を，Fig.3 と Fig.4 に，それぞれ示す．

OPM には，Object と Process がある．航空機設計の OPM には，Object として，Stakeholder Needs Set と,3 個の Assumptions and Constraints Set，3 個の Requirements がある．Process には，Defining と Realizing, Implementing という 3 種類の Process がある．さらに，物理的な Object として，Aircraft, System, item, item component がある．

Fig.3 の Systemigram では，OPM のプロセスを振舞とした．また，OPM の論理的な Object を動機に，物理的 Object を構造に対応付けた．

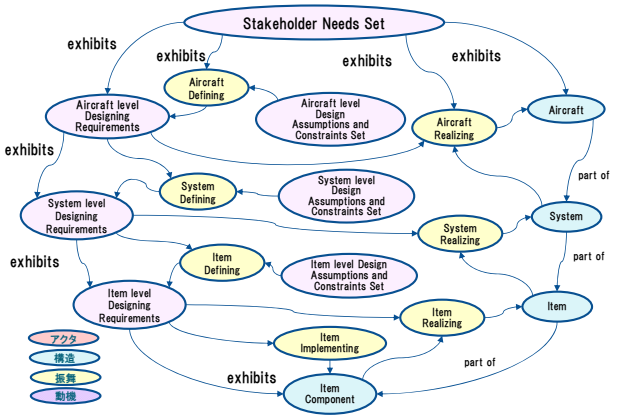


Fig.3 Aircraft Design by Systemigram

Fig.4 の ArchiMate では，Systemigram と同様に，OPM のプロセスをビジネスプロセスとした．また OPM の論理的な Object を動機要素に，物理的 Object を Device に対応付けた．ここで，ArchiMate では，要素の種類によって，使用できる関係が定義されているため，要素種別に応じて，異なる関係を使う必要があった点を注意しておく．また，ArchiMate では，動機要素として，driver, requirement, constraint の 3 種類を Object の種類に応じて使い分けている．

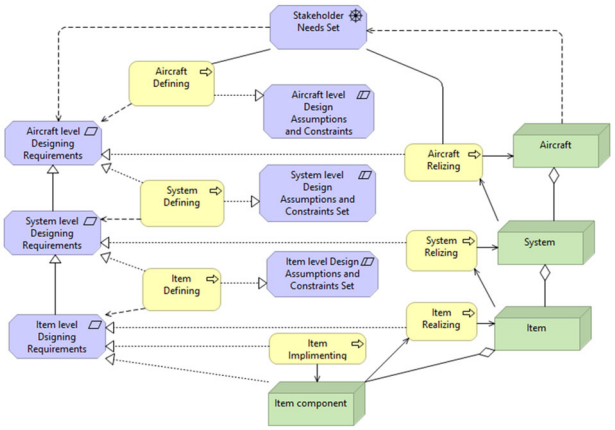


Fig.4 Aircraft Design by ArchiMate

次に，ビジネス系の例に対して，OPM，

Systemigram，ArchiMateの表現を示す．以下では，3手法を比較するために，ソフトウェア設計の共通問題[29]に対して3手法を適用した結果を説明する．この共通問題では，酒販売会社の倉庫で酒銘柄を積載するコンテナを管理しておき，注文に応じてコンテナ内の酒を発送する業務をシステム化することを考える．

この共通問題に対するOPM，Systemigram, ArchiMateによる記述例を，Fig.5, Fig.6, Fig.7に示す．

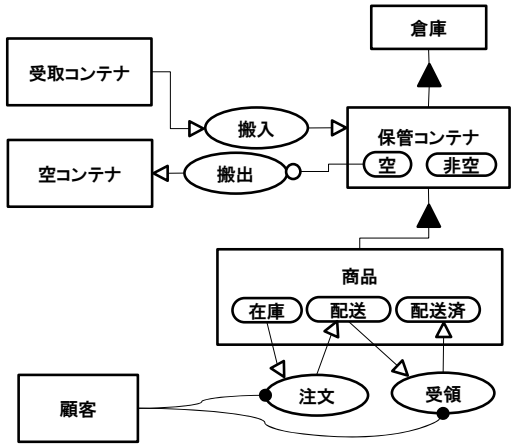


Fig.5 Container Warehouse Management in OPM

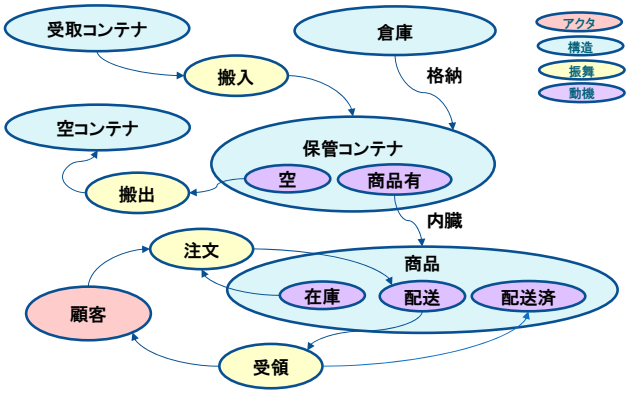


Fig.6 Container Warehouse Management in Systemigram

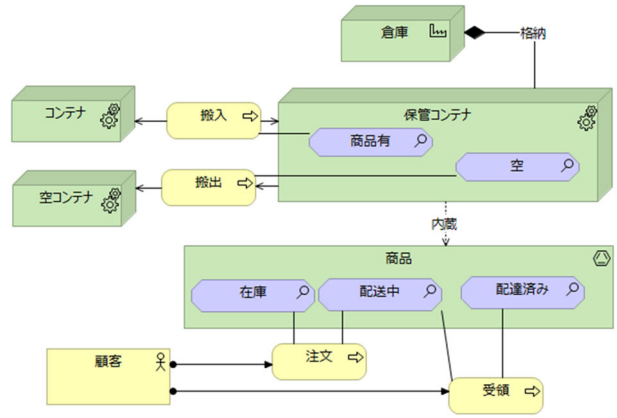


Fig.7 Container Warehouse Management in ArchiMate

Fig.5に示したOPMのObjectには，倉庫，受取コンテナ，保管コンテナ，空コンテナ，商品，顧客がある．保管コンテナの状態として，空と，空でない（非空）がある．また，コンテナに内蔵される商品には，在庫，配送，配送済みという3状態がある．倉庫には，複数のコンテナが保管される．倉庫から空コンテナが搬出される．コンテナが倉庫に搬入される．搬出processと空状態の関係は，保管コンテナが空になることが搬出processの前提条件になっていることを示す．

顧客が商品を注文すると，在庫があれば配送される．顧客によって，配送された商品が受領されると配送済み状態になる．

Fig.5のOPMには，搬入，搬出，注文，受領という4個のProcessがある．Fig.6のSystemigramでは，顧客がアクタ，倉庫，受取コンテナ，保管コンテナ，空コンテナ，商品が構造である．空，商品有，在庫，配送，配送済が状態である．また，振舞は，搬入，搬出，注文，受領である．Fig.7のArchiMateでは，顧客がビジネスアクタである．倉庫がArchiMateの設備である．コンテナ，空コンテナ，保管コンテナはArchiMateの装置である．商品はArchiMateのマテリアルである．

搬入，搬出，注文，受領がArchiMateのビジネスプロセスである．商品有，空，在庫，配送中，配送済みはArchiMateのアセスメント要素である．

5.2 GPDACによるモデルの比較

OPM, Systemigram, ArchiMateでは，1つの図を使用する．これに対して，SysMLでは9種類の異なる図を使用する点が異なる．SysMLとOPMの関係についてDoriが比較していることから，以下では，Systemigram, OPM, ArchiMateについて，比較する．

OPMとSystemigramには，自然言語表現と対になる図式表現が対応している．自然言語からArchiMateを作成する方法が提案されている[30][31]．

OPMでは，SysMLと同等のシステムの詳細な状態変化を表現できる[25][26][27]．しかしArchiMateはノードと関係がエンタープライズアーキテクチャの表現に特化していることから，システムの具体的な状態変化や方程式を記述することはできない．これに対して，Systemigramは抽象的な記述であるから，ノードと関係に対する制約がない．このため，具体的な状態変化や方程式をSystemigramで表現できる．

以下では，各図式のノードに着目して比較する．

OPMのノードはObjectとProcessである．

ノードに対応するArchiMateのAspectは，Motivation(動機), Active Structure(アクタやコンポネントなど)，Passive Structure(データやデバイスなど), Behavior(プロセス，機能，サービス)がある．

ノードに対応するGPDACの要素には，Goal, Process, Data, Actor, Controlがある．GoalとControlはArchiMateのMotivationに対応する．

したがって，GPDACによって，Systemigram，OPM, ArchiMateのノードを意味づけることができると思われる．たとえば，GPDAC解釈を定義できれば，GPDAC解釈によって，Systemigram，OPM，ArchiMateを統一的に相互比較できる．

たとえば，G,P,D,A,Cをそれぞれ Goal, Process, Data, Controlの集合，x, yをG∪P∪D∪A∪Cの要素とする．このとき，対(x, y)の集合を定義できる．systemigram, OPM, ArchiMateで記述された図をSとすると，Sのノ

ード対(n, m)に対して，GPDACの要素対(x, y)を対応付けることができる．このとき，S に対する GPDAC 解釈を{(x, y)| S のノード対(n, m)が(x, y)に対応}で定義できる．

上述したノードについての対応関係を図示すると，Fig.8 の通りである．Fig.8 では GPDAC が細粒度のノード種別を持つことが分かる．

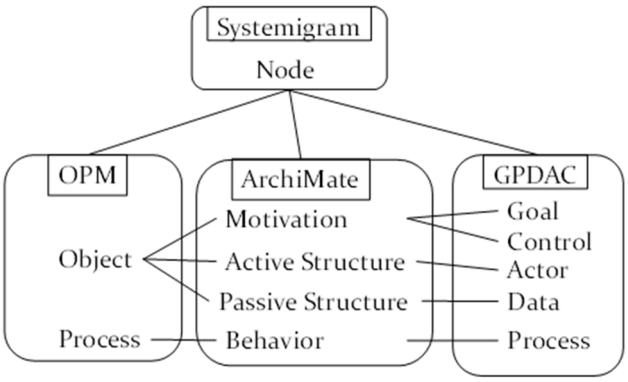


Fig.8 Model Comparison

6. 考察

6.1 新規性

これまで, Systemigram と ArchiMate については，筆者らが比較してきた．本稿では，OPM と GPDAC を含めて Systemigram, ArchiMate と比較することにより，これらの共通点と差異点を明らかにした点に，新規性がある．

6.2 有効性

Systemigram, ArchiMate, OPM, GPDAC の関係を明らかにしたことにより，Systemigram, ArchiMate, OPM で記述された DX 推進のあり方を GPDAC で相互比較できる可能性があることが明らかになった．たとえば，前述した GPDAC 解釈で Systemigram, ArchiMate, OPM による DX 図式の同一性を比較できる．なお，GPDAC には図式表現がないので，これら 3 種の図式を表現として利用できる．

6.3 限界

本稿で取り上げた図式以外にも複雑なシステムを表現する多くの図式がある．たとえば，ものこと分析図[32]は，生産工程を見える化するための図式である．もの・こと分析では，材料や製品など対象を「もの」とし，材料から製品を作る一連の活動を「こと」とすることにより，生産工程を分析して，無駄を発見して最適化することができる．ものこと分析図を Systemigram で表現できる[33]．

これ以外にも，i*framework や TOC(Theory of Constraints)などのゴール図がある[34][35]．

また，複雑な人間活動からなるシステムを分析する手法として，Holland による機能共鳴分析法 FRAM(Functional Resonance Analysis Method)がある[36][37]．FRAM は機能のネットワークを介した社会技術システムの複雑な機能共鳴を分析するために使用されてきた．FRAM の機能は，6 側面(Aspect)を持つ六角形のノードによって定義される．これらの側面は，入力 Input，出力 Output，時間 Time，制御 Control，資源 Resource，および前提条件 Precondition である．機能の出力側面は，他の関数の他の 5 側面に接続できる．FRAM は安全分析に役立つ機能を提供している．たとえば，FRAM は，情報システムのセキュリティの脆弱性分析に適用できる[38]．筆者は，FRAM の行列解釈を定義[39][40]して，企業不正のトライアングル[41][42]を分析できることを明らかにした．

さらに，安全なシステムを実現する上で重要になる知識として，ゴール指向に基づく GSN(Goal Structuring Notation)で記述される保証ケースがある[43]．たとえば，Systemigramに基づく客観的保証ケースレビュー法が提案されている[44]．システムの安全性を表現するためのGPDACの扱いも重要な研究課題である．

7. まとめ

本稿では，横幹知で推進する DX 調査研究会での活動事例として，GPDAC による知の統合に向けた取り組みとして，知の共創構造と GPDAC を紹介した．

知の共創構造では，データ駆動工程設計法が DX の実践事例を説明する理論として捉えることができることを示した．

GPDAC では，OPM による航空機設計例題と，ソフトウェア設計の共通例題に対して Systemigram，OPM, ArchiMate の記述例を示した．この結果，システム思考，システム工学，エンタープライズアーキテクチャで発展してきた 3 種の知識の相互関係を GPDAC で説明できることを明らかにした．

しかし，Systemigram，OPM，ArchiMate の他にも多くの図式表現がある．今後，ゴール指向要求工学分野の知識との関係，FRAM などの社会技術システム分野知識，安全性ケースなどの安全性保証知識との統合についても研究を進めていく予定である．

謝辞

本稿をまとめる上で，貴重なご意見を頂いた，慶応義塾大学 本多敏教授，西村秀和教授，ならびに, OPM に関する文献を教えていただいた舩橋誠壽氏に深謝します．また，横幹知で推進する DX 調査研究会でご議論いただいた研究会の皆様に感謝します．

参考文献

1) 横幹知で推進する DX 調査研究会，https://www.trafst.jp/document/chousa-kenkyuu/
2) 山本修一郎，舩橋誠壽, 西村秀和, 本多敏, 横幹知で推進する DX 調査研究会の紹介， 横幹連合会誌「横幹」16-2, 104/111 (2022)
3) 総務省, 令和 3 年版 情報通信白書 (2021)

4) IPA, 「DX 先進企業へのヒアリング調査 概要報告書」, https://www.ipa.go.jp/files/000093364.pdf
5) Paul Stock and Rob Burton, Defining Terms for Integrated (Multi-Inter-Trans-Disciplinary) Sustainability Research, Sustainability 3, 1090/1113, (2011) doi:10.3390/su3081090
6) 鈴木久敏，”Transdisciplinarity”を巡って，Oukan 8, 1, 3/4 (2014)
7) 山本修一郎, デジタルデザインエンジニアリング知識体系の試み, PM 学会　2021 年度秋季研究発表大会予稿集, 397/402 (2021)
8) Robert Cloutier, Brian Sauser, et.al, Transitioning Systems Thinking to Model-Based Systems Engineering: Systemigram to SysML models, IEEE Trans. on Systems and Cybernetics, 45,.4, 662/674 (2015)
9) 山本修一郎, Systemigram によるものこと分析の試み, 信学会　KBSE 研究会 2021-43, 12/17 (2022)
10) Xia, C., Zhi, Q., Zhou, Z., and Yamamoto, S., An Approach to Transform Enterprise Architecture Models from Systemigrams, 10th IEEE International Conference on Software Engineering and Service Science (ICSESS), 571/574, (2019)
11) 山本修一郎, ArchiMate による Systemigram 表現法の考察, AI 学会　KSN 研究会, KSN-030, 07- (2022)
12) 山本修一郎，システム思考によるデジタルガバナンス・コードの分析，KBSE2022-8, 47/52 (2022)
13) Checkland, P., Systems Thinking, Systems Practice, John Wiley & Sons Ltd. (1990)
14) Boardman, J., Wholes and Parts-A Systems Approach, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. VOL. 25, NO. 7, 1150/1161 (1995)
15) Boardman, J and B Sauser, Systems Thinking: Coping with 21St Century Problems. Boca Raton, FL: Taylor & Francis, CRC Press (2008)
16) 山本修一郎,現代エンタープライズ･アーキテクチャ概論 - ArchiMate 入門, デザインエッグ社 (2016)
17) The Open Group, ArchiMate® 3.1. Specification. C197 (2019)
18) The Open Group, TOGAF v9.2, C182 (2018)
19) 2030 年への企業の羅針盤づくり～DSDG フレームワーク～,質価値創造研究会, 中部品質管理協会 (2022)
20) 山本修一郎, SDGs に向けたデジタル知の統合, AI 学会, 知識流通ネットワーク研究会, KSN-029, 01 (2021)
21) 山本修一郎，細見純子，データ駆動工程設計法の提案，信学会，KBSE2022-24, 79/84 (2022)
22) 山本修一郎, DX の基礎知識，具体的なデジタル変革事例と方法論，近代科学社 Digital (2020)
23) IPA, 中小規模製造業者の製造分野におけるデジタルトランスフォーメーション（DX）推進のためのガイド，https://www.ipa.go.jp/ikc/reports/mfg-dx.html
24) Hammer, M. and Champy, J., Reengineering the Corporation- A Manifesto for Business Revolution, Harper Business (1993)
25) Dori, D., Object-Process Methodology—A Holistic Systems Paradigm, Springer (2002)
26) Dori, D., Model-Based Systems Engineering with OPM and SysML, Springer (2016)
27) E.クロウリー, B.キャメロン, D.セルヴァ, システム・アーキテクチャ--複雑システムの構想から実現まで, 稗方和夫訳，丸善出版 (2020)
28) Linwen Li, Natali Levi Soskin, Ahmad Jbara, Moti Karpel, and Dov Dori, Model-Based Systems Engineering for Aircraft Design with Dynamic Landing Constraints Using Object-Process Methodology. IEEE Access, 61494/61511 (2019)
29) 山崎利治, 共通問題によるプログラム設計技法解説, 情報処理，vol.25, No.9, 934 (1984)
30) 山本 修一郎, 藤川なつ子，活動分析に基づくビジネスプロセスの可視化手法，第 77 回日本情報経営学会全国大会, 97/100 (2018)
31) Yamamoto, S., Zhi, Q., Zhou, Z., Aspect Analysis towards ArchiMate Diagrams, KES 2019, Procedia Computer Science, Volume 159, 973/980 (2019)
32) 中村善太郎，シンプルな仕事の構想法～もの・こと分析で成功する～，日刊工業新聞社 (2003)
33) 山本修一郎, Systemigram によるものこと分析の試み, 信学会　KBSE 研究会 2021-43, 12/17 (2022)
34) 山本修一郎，ゴール指向によるシステム要求管理, ソフト･リサーチ･センター (2007)
35) Yamamoto, S., Kaiya, H., Cox, K., and Bleistein, S., Goal Oriented Requirements Engineering -- trends and issues, E-89D, No.11, 2701/2711, IEICE (2006)
36) Hollnagel, E.: FRAM - the Functional Resonance Analysis Method: Modelling Complex Socio-Technical Systems. Boca Raton: CRC Press (2012)
37) Hollnagel, E: A Brief Introduction to FRAM, https://www.functionalresonance.com/, last accessed 2022/2/10.
38) 金子朋子，セーフティ＆セキュリティ入門，AI, IoT 時代のシステム安全，日科技連 (2021)
39) Yamamoto, S., Comparative Study on Functional Resonance Matrices, Knowledge-Based Software Engineering: 2022: Proceedings of the 14th International Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2022) (2022)
40) 山本修一郎, 機能アスペクト関係行列によるサービス不正分析，KBSE 研究会 11 月(2022)
41) Cressy, D. R. Other People's Money: Study in the Social Psychology of Embezzlement, The Free Press (1953)
42) Albrecht, W. S., Fraud in Government Entities: The Perpetrators and the Types of Fraud, Government Finance Review, Vol.7, No.6, 27/30 (1991)
43) Kelly, T., A Six-Step Method for the Development of Goal Structures, York Software Engineering (1997)
44) 山本修一郎，森崎修司，渥美紀寿，実践的保証ケース作成方式, SEC journal Vol.13 No.1, 16/23 (2017)