

問題発見力とモデリング力を涵養する 要求獲得実践手法と人財育成

位野木万里

イノベーションを起こすための「問題発見力」、アイデアをシステムに落とし込む「モデリング力」の強化を目指した、要求工学／サービスデザイン分野のリカレント教育の実践例を紹介し、人と AI が共存する時代の人財育成について議論する。

1. はじめに	159
2. アイデア創出と問題発見をモデリングでアプローチする教育講座	160
3. モデルとモデリング	162
3. 1 モデルの重要性	162
3. 2 改めてモデルとは何か	163
3. 3 改めてモデルの重要性とモデル比較による差異の特定	166
4. モデルの上位モデルであるメタモデル	167
4. 1 工程成果物（ドキュメント）とモデルとメタモデル	167
4. 2 工程毎の成果物モデルとその上位のモデル：Zachman Framework	168
4. 3 メタモデルの拡張	170
5. 人財育成で重要な問題発見力とモデリング力	171
6. おわりに	173
参考文献	173

1. はじめに

顧客の課題を解決し、社会にインパクトを与えるイノベーションの社会実装を目指して、デザイン思考やアート思考による要求獲得が注目されている。生成 AI の登場により、高品質のソフトウェア製品やサービスを素早く提供できる可能性は高まったものの、多様化するユーザと共感できる課題や解決策は何なのか？何を創ればイノベーションが起こせるのかという「問題の発見」の方法は明らかではない。

ところで、アメリカの実業家 James Webb Young によれば、An idea is nothing more nor less than a new combination of old elements.（アイデアとは既存の要素の新しい組合せ以外の何ものでもない）と言っている[Young2003]。そして、現実をよく観察して、そこから浮かび上がる様々な事象やアイデアをどう組み合わせるのか、その関係性を見出すことが重要であるとしている。

問題の発見とアイデアの創出には、既存のソフトウェアやサービスを効率良く組み合わせ、新しいアイデアを具現化する能力、すなわち、再利用力が重要と考えられる。ソフトウェア再利用では、アイデアを含む再利用対象の成果物を体系的に整理し、組織の資産であるコアアセットとして保持し、コアアセットに基づいて新規ソリューションを開発する。組織において再利用による開発の成功には、コアアセットに何を含めるか、どのように維持管理していくかが重要であり、ソースコード、自動化ツールなどのバイナリの成果物よりも、先人の良い設計の考え方であるデザインパターンや、ノウハウを集大成した設計モデルのような形式知化された成果物が重要な資産と考えられている。

著者は、アート思考要求工学、デザイン思考要求工学などの問題発見とソリューション開発につながる要求獲得手法[JISA2023]や、既存のアイデアを組み合わせる新たなアイデアを考案することを含めた再利用率を涵養するためのそしてソフトウェア再利用手法[Inoki2008][Inoki2007]を考案し、例えば、国立情報学研究所が主催するリカレント教育プログラム(<https://www.topse.jp/ja/index.html>)であるトップエスイープログラムの要求工学シリーズ、アーキテクチャシリーズのアート思考要求工学、デザイン思考要求工学、そしてソフトウェア再利用演習等講義で適用してきた。

ここでは、要求工学／サービスデザイン分野のリカレント教育の実践例を紹介し、とくに教育プログラムの中で提供している、モデリングによるアイデアの創出や問題の発見手法について説明し、そのようなモデリング手法において、一段上位のメタモデルを定義すること、そのメタモデルを再利用対象、つまり知識継承の源泉とすることの重要性について問いたい。加えて、そのようなメタモデルを意識したモデリング手法を駆使するモデリング力が、新たな時代に人が身に着ける重要な能力となることを議論する。

2. アイデア創出と問題発見をモデリングでアプローチする教育講座

Neri Oxman によると、イノベーションを起こすには、Science、Engineering、Design、Art の視点での絡み合い（融合）が重要であることが示されている。独創的なアイデアの創出や社会への問題提起を意識し、クリエイティブな成果を生み出すための思考のフレームワークとして、Krebs Cycle of Creativity (KCC) が提案されている [Oxman2016]。

トップエスイープロジェクト要求工学シリーズでは、真の顧客要求の獲得、分析、仕様化、検証のための基本的な知識と実践手法を習得する。トップエスイーの教育プログラムは、基本的には、Science による Engineering モノづくりを支える技術に

フォーカスしたプログラムを提供している。要求工学シリーズでは Engineering をベースにしながら、Design や Art の視点での技術者のレベルアップを図るための科目の工夫をしている。

図 2-1 に要求工学シリーズの 4 つの講座の関係性を示す。本シリーズでは、基礎から応用、技術主導からユーザ主導など多角的な視点から 4 つの講義を提供している。要求工学基礎と要求工学先端は、Engineering 視点で、基礎から応用をカバーする要求獲得技術を提供している。また、デザイン思考要求工学はユーザ主導、ユーザエクスペリエンス、共感共創を主体とした体験主体でユーザ視点でサービスをデザインすることを主眼としたモデリング手法を、アート思考要求工学は、参加する技術者の感性を強化することが主体の講座となっている。

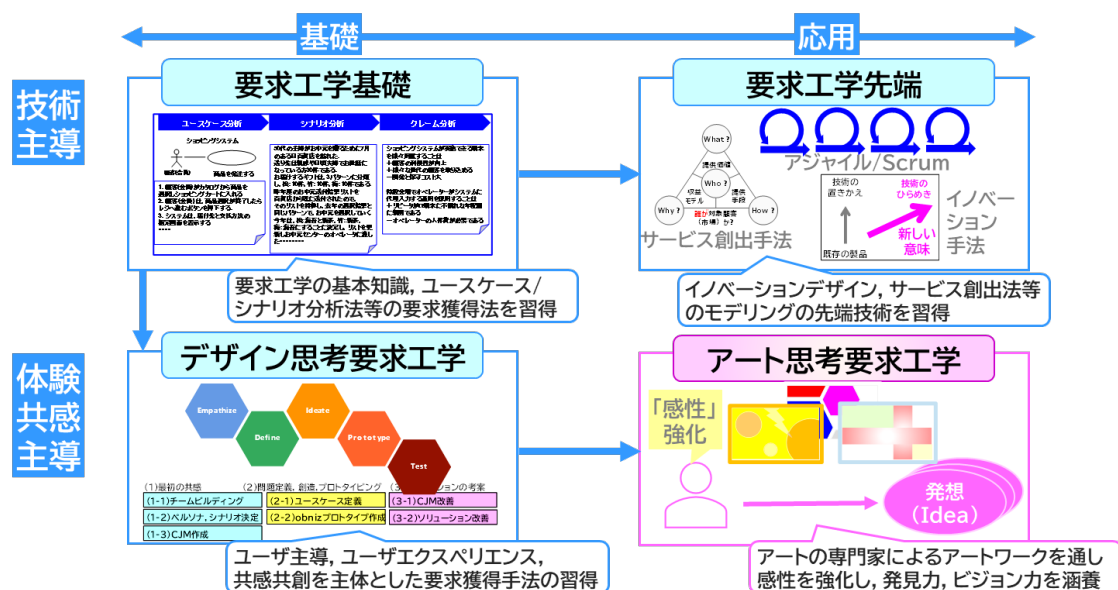


図 2-1 トップエスイー要求工学シリーズの講座

ところで、生成 AI を積極的に活用しながら、さらにエンジニアリング力を高めるためには、前述の再利用率の強化が必要と考えられる。そこで、トップエスイーのアーキテクチャシリーズでは、再利用率を向上させるソフトウェア再利用演習の講座も実施している。

ソフトウェア再利用演習では、ソフトウェア開発において考えられる様々な再利用の形態を体系立てて習得する。再利用の対象には、ソースコードやコンポーネントに加えて、業種業務領域の概念モデル、設計モデル、設計ノウハウに相当するパターンやフレームワーク、プロダクトラインが含まれる。

図 2-2 にソフトウェア再利用の変遷の一部を示す。従来は、開発済みのソフトウェアから汎用的に利用可能と考えられる部分を抽出し、それらを次の製品開発で活用するという再利用が中心であった。その結果、ソフトウェア開発全体に対する再利用部分はわずかで、生産性にそれほど寄与しないという課題があった。

こうした課題の解決のために、望ましい再利用の方法として、プロダクトライン型開発が考案されている。プロダクトライン型開発では、あらかじめ、流通・販売、製造・品質などのドメインを限定し、製品開発の未来を予測し、そうした製品を開発するために必要なだけのソフトウェア資産、つまりコアアセットを先回り開発しておく。そして、コアアセットを再利用し、効率よく製品開発を進めていくことを目指している。

プロダクトライン型の再利用は、製品開発の未来を予測したうえで、コアアセットを作成するので、従来型の再利用に比べてコアアセットの再利用性は高くなると考えられる。しかし、コアアセットは先回り開発をするため、投資が必要です。投資対効果を考慮したうえでコアアセットの開発スコープを定義する必要がある。本講座では、このよう組織のソフトウェア資産を蓄積し開発に再利用できる、プロダクトラインの開発と運用のための知識と実践知識の習得を目指している。

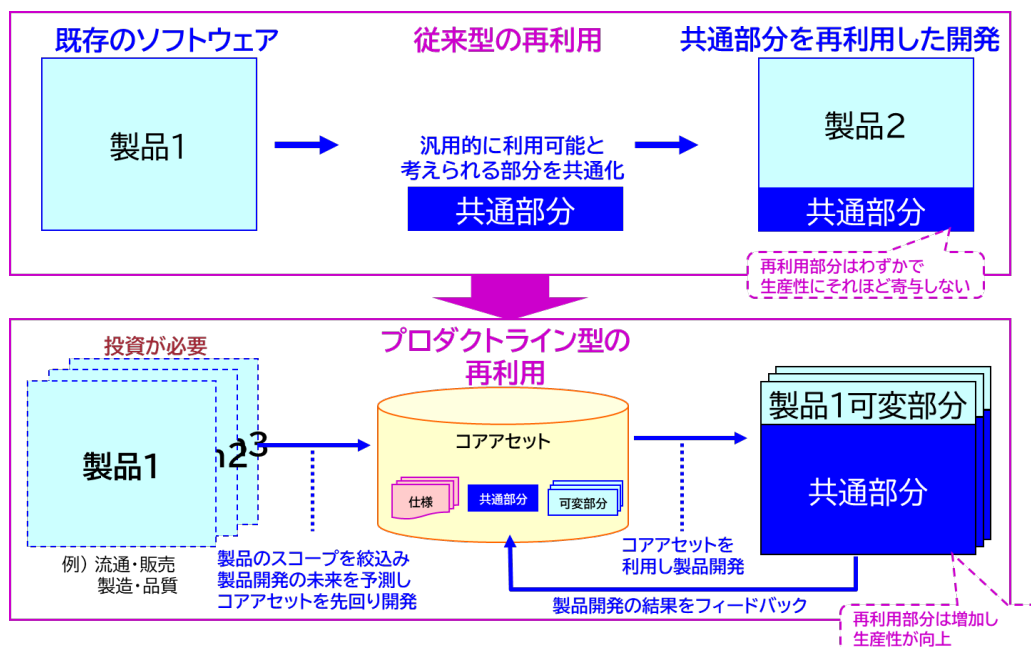


図 2-2 トップエスイー ソフトウェア再利用演習のベースになる考え方

3. モデルとモデリング

要求工学における要求獲得、分析、仕様化、検証のプロセスは、各タスクのアウトプットはモデルが主体になっています。また、再利用によるモノづくりにおいても、再利用対象の重要な要素として仕様や設計モデルを取りあげている。

3. 1 モデルの重要性

問題の発見には、現状の把握が不可欠であるが、現状を漠然ととらえていたのでは、

十分な把握は困難である。現状と望ましい姿を、何等かの視点でモデル化し、2つのモデルを比較して浮かびあがる差異を抽出することで、問題の特定がしやすくなると考えられる。

新たなアイデアを再利用によって創出する場合でも、既存のアイデアとなるものを、一定の視点によってモデル化し、接続部分を標準化することによって、組み合わせが容易になり、新たなモデルが見えやすくなると考えられる。

3. 2 改めてモデルとは何か

ここで、改めてモデルとは何かを整理してみる。

3. 2. 1 国語辞書による定義

精選版 日本国語大辞典 によれば、「モデル」とは次の①～⑥として定義されている。ソフトウェアの開発の分析や設計モデルは、例えば、①や⑤が相当すると考えられる。しかし、本稿の対象としている、問題を発見するためのモデルに対しては、明確に説明されている状況ではない。

- ① ある事柄の手本や見本となるもの。原型、典型、ひな型。
- ② 絵・彫刻・人物写真の素材となる特定の人物や物。また、職業としてポーズを提供する、若い女性など。
- ③ 文芸作品の素材とした実在の人物や事件。
- ④ 模型。
- ⑤ 商品や試みなどの標準とされるものやその形式。
- ⑥ 「ファッションモデル」の略。

3. 2. 2 システム科学分野からの定義

システム科学分野の文献[Takahashi2022]によれば、次のような定義が示されている。

*A と B という二つの対象があったとする。
B が A の何等かの特徴を写していると考えられるとき、
B を A のモデル、A を B の原型という。*

国語辞典からの定義と比較すれば、上記の「モデル」の定義は明確である。文献[Takahashi2022]では、より厳密化するため、「モデル」の定義に関する初期の文献[Apostel1961]に基づき、定義の詳細化を試みている。

Apostel は、利用する目的に従って抽象化により表現されたモデルは、その目的を達成するために次の必要な機能を持っている、としている。

- Theory formation: 知識の構造化を行って理論を形成
- Simplification: 認識したい対象を単純化して理解を促進
- Reduction: 複雑な現象と特定の関係に還元
- Extension: あるいは、逆に認識した関係をより広い領域へと拡張
- Explanation & Adequation: モデルを用いて認識した現象を説明し、納得させる
- Concretization: 一般的あるいは抽象的な概念をモデルに表現することで具体化
- Globalization: モデル化した領域とは異なる他の領域にモデルを適用
- Action: モデルに基づいて得られたことを実践
- Experimentation: モデルを用いて仮想的に実験を行う

モデルが持つべき機能を整理した上で、Apostel は、モデルを次のように定義している。なお、下記において、原型 A、そのモデル B の関係性がわかるように、文献 [Apostel1961] の原文から、記号 A と B を置き換えて示す。

Any subject using a system B that is neither directly
nor indirectly interacting with a system A,
to obtain information about the system A,
is using B as a model for A.

あるシステム A と直接的であれ間接的であれ共通部分を持たないようなシステム B を利用して、システム A についての情報を得ている主体は、
B を A のモデルとして使っている

モデルには機能があり、ある主体によって何等かの目的の達成のためにその機能が利用され、原型から写像されて定義されることになる。このような、主体、原型、目的、モデルの関係性を、文献 [Takahashi2022] では図式化して説明している。図 3-1 に、文献 [Takahashi2022] での関係性の図より、各要素の記号を文献 [Apostel1961] の記述に合わせて加筆修正したものを示す。

ここに示すように、主体 S は、原型 T をある目的 P を達成するために、モデル M を利用している、と表現することができる。または、モデル M は、主体 S が目的 P を達成するために原型 T の要素を写像した結果とも表現できる。

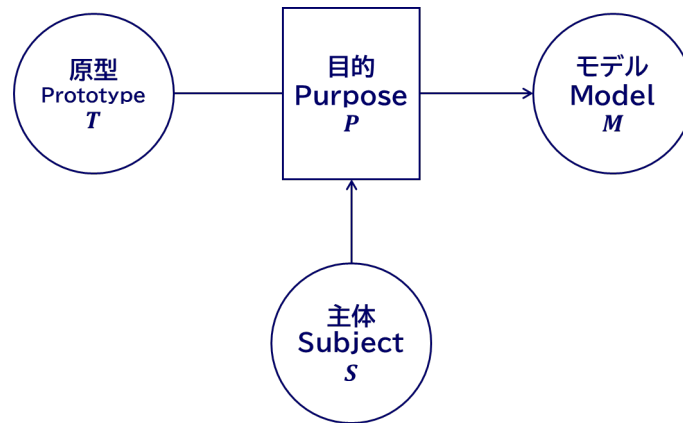


図 3-1 主体、目的、原型、モデル間の関係

3. 2. 3 モデリング

モデリングとは、モデルを作成すること、すなわち、原型からモデルを作成する過程である。良いモデルとそうでないモデルはどのように判断が可能なのか考えてみる。モデルが良いモデリングを経て定義されていれば良いモデルの可能性は高く、良くないモデリングにより作成されれば、良くないモデルになるリスクが高いと考えられる。良いモデリングとは、原型に対するモデルが目的に沿って妥当に写像されていること、つまり、写像が高品質であること、と説明ができる。そのような高品質な写像として、原型とモデルの間の写像の同型性が考えられる。

同型性について数学的な定義を示しておく次のようになる（文献[Moriya1984]）。

G, G' を群とするとき、 G から G' への写像 f が

$$f(xy) = f(x)f(y), x, y \in G$$

を満足するとき、 f を G から G' への準同型という

G から G' への準同型が単射であるとき、

f を G から G' への同型という

原型 A に対して良いモデル B をモデリングするには、 A から B への写像が準同型（または同型）になるように定義すればよい、といえる。原型 A とモデル B の間に写像が定義でき、それらが準同型（または同型）であれば、原型 A の代わりにモデル B を扱うことの妥当性の根拠になると考えられる。

これから開発しようとする未知の対象システム A を、分析、設計、実装のモデリングによって徐々に具体化していくという過程においては、モデル B は、原型 A を特定の視点から写像した部分的な表現になることが一般的であるので、その写像が単射になることはほとんど稀である。加えて、対象システム A からモデル B への写像自体も定義できるかどうかあいまいである。情報システム開発のモデリング過程

では、良いモデルであるかどうかを厳密に数学的モデルにおいて証明しながら進めるわけではない。現実には、原型に対してモデルが妥当かどうかは、例えば、準同型や同型写像の数学的考え方をバックグラウンドとして、主観的に評価していくことになると思われる。

3. 3 改めてモデルの重要性とモデル比較による差異の特定

改めてモデル化することの重要性と、モデル比較によって差異を特定することで、問題点を明らかにできることを整理する。図 3-2 に、モデル間の比較による差異の特定方法の考え方を示す。

例えば、組織の現状を、*As is T*とし、その組織のあるべき姿を *To be T*とする。組織の現状とあるべき姿は、複雑かつあいまいなため、そのままでは比較が困難で、現状の課題もあるべき姿に到達するまでになすべきこともが明らかになりにくいと考えられる。

それぞれの原型を、*As is M*と *To be M*の構造やふるまい等のモデルに写像する。同じ考え方で写像されたモデル間同士であれば比較が容易であり、モデル間の差異を特定しやすいと考えられる。この場合、現状の *As is M*モデルとあるべき姿である *To be M*モデルの差異は、現状モデルに対する不足事項、すなわち「問題」としてとらえることが可能になる。

このように原型同志の比較は困難でも、各々をモデルに写像すれば、モデル間の比較と差異の抽出は容易になると期待できる。

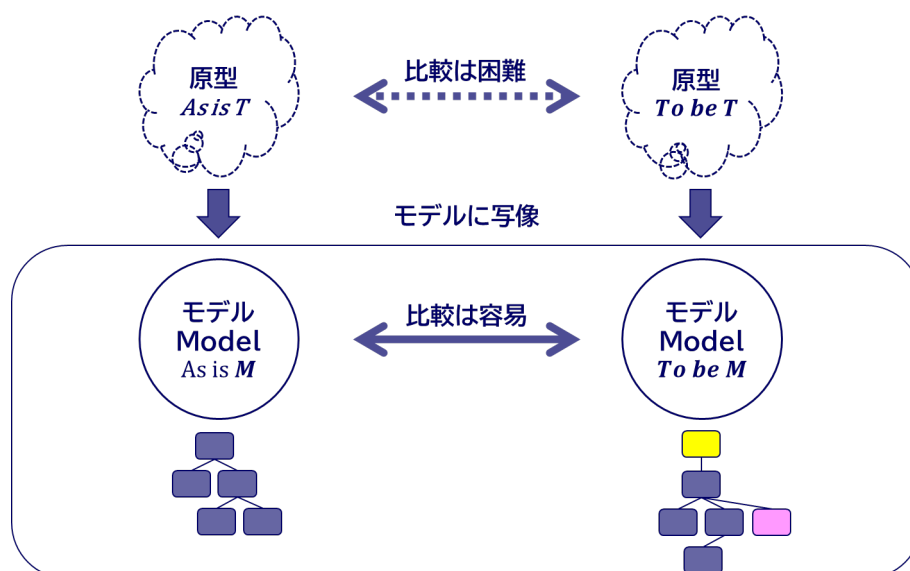


図 3-2 モデル間の比較による差異の特定

4. モデルの上位モデルであるメタモデル

メタモデルとは、モデルを抽象化したという意味であり、対象領域に対するモデル構築において、そのモデルを定義するための枠組みや規則を示す。また、メタモデリングとは、メタモデルの構築や分析をする行為のことです。なお、メタとはギリシャ語で上位のという意味を示している[Gonzalez-Perez2008]。

システム開発では各工程で要求仕様書や基本設計書などのドキュメントを工程成果物として作成することになる。以下、工程成果物と、モデル、メタモデルの関係を組織での知識継承のための再利用資産の視点から掘り下げてみる。

4. 1 工程成果物（ドキュメント）とモデルとメタモデル

要求仕様書や基本設計書などのドキュメントは、その表現や見た目に関する「ビュー」と、中身や内容に相当する「モデル」に分けることができる（図 4-1）。一般にドキュメントは、ビューとモデルが混在した状態になっている。ビューとモデルを混在させたドキュメントを再利用すると、再利用したい部分の特定が困難なことや、影響関係が不明確となるリスクが高まる。つまり、ビューとモデルという異なる性質のもの、すなわち、見た目と中身が渾然一体化した状態では、再利用部分の特定やカスタマイズ結果の影響範囲の特定は簡単ではない。従って、ドキュメントを再利用しやすくするには、見た目と中身の役割を分離して、それぞれを再利用対象として扱うことが重要である。

ドキュメントがビューとモデルに分離されていても、ビューやモデルの粒度や定義内容が均一でないと、それぞれの再利用は困難である。ビューとモデルには、要求定義や基本設計のノウハウが凝縮されている。前者は、発注者／企画者にとってわかりやすいドキュメントに関するノウハウ、後者は抽出すべき要求／仕様の内容に関するノウハウである。そこで、ビューとモデルの定義内容を均一化するために、それぞれの定義方法のノウハウを一般化することを考えられる。

ビューとモデルは、特定の発注者向け要求仕様書の構成要素ととらえることができる。同一ドメインに汎用化した際のビューとモデルのノウハウを、それぞれテンプレートとメタモデルとして定義することができる。

テンプレートは、わかりやすいドキュメントの表現方法（記述のひな形）であり、メタモデルは要求／仕様として定義すべき要素とその関係を定義した方法になる（図 4-1）。よって、テンプレートとメタモデルを組織で標準化しておけば、これらに則ったビューとモデルは、均質化されており、再利用箇所の特定やカスタマイズがしやすくなる。

設計の本質は、ビューよりもモデルに現れる。モデルを一般化したものがメタモデ

ルであり、モデル、メタモデルも設計の再利用の対象になる。さらに、メタモデルに基づいて設計を行う際、より良いモデルを作るには、設計の考え方に相当する「パターン」を再利用することが有効である。

パターンにもさまざまなものがある。開発対象にするソフトウェアシステムの、ドメインの種類、費用や納期などの条件、開発メンバーのスキルなどを考慮すると、あらゆるパターンを同時に選択することは困難である。つまり、パターンの取捨選択が必要である。ここで有効になるのがフレームワークである。フレームワークは、対象ドメインや条件にそった、最適なパターンの組み合わせのノウハウを具現化した成果物に相当する。最適なパターンの組合せの知識である「フレームワーク」を再利用することで、高品質なモデルを効率的に構築することが期待できる。

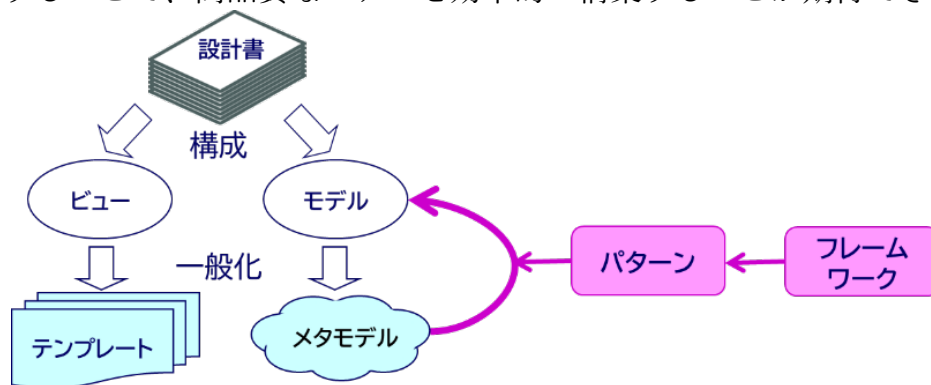


図 4-1 ドキュメントとモデルとメタモデルの関係

4. 2 工程毎の成果物モデルとその上位のモデル：Zachman Framework

ソフトウェア開発の視点では、工程別の分析、設計、実装、テストモデルがある。それぞれの工程成果物のモデルには、役割や抽象度によって、モデルが細分化される。細分化されたモデルを整理分類するため、様々な思考のフレームワークが提供されています。そのようなフレームの一つに例えば、Zachman Framework（以下、ZF と略す）がある。これらの視点をを用いて、設計成果物を整理することが重要で、そのようなモデルをメタモデルとして定義することができる。

ZF は、大規模で複雑な企業システムを理解する目的で、全体と部分の関係を 30 個の視点の集合により表した、思考のための枠組みである。表 4-1 に ZF の全体像を示します[IEEE2000]。ZF は、専門軸とステークホルダの役割軸の 2 つの軸により、対象を分解することで、システムの全体像を把握する。専門軸は、(Data、Function、Network、People、Time、Motivation)の 6 つの分類で、これらは、5W1H (What、How、Where、Who、When、Why) にそれぞれ該当する。抽象軸は、ステークホルダのタイプにより、Planner、Owner、Designer、Builder、Subcontractor の 5 つの役割を定義している。

ソフトウェア製品開発におけるプロダクトを、専門軸と時間軸に沿って 30 個に分割して、各グリッドにソフトウェア開発の専門知識を反映させたモデルを対応させるととらえることができる。このようにすることで、専門知識をより具体的に形式知化し、蓄積・共有、維持管理が可能になると期待できる。

コアアセットを適切に整理、分類、蓄積するための視点、つまりメタモデルをまとめると図 4-2 のようになる。図 4-2①に示すように、コアアセットの要素はシステム開発のライフサイクルで構築または利用される成果物であり、これらは IEEE std 1471-2000[IEEE2000]、ISO/IEC/IEEE 42010:2022[ISO/IEC/IEEE2022]によれば、モデルとして表現される。また、各モデルは視点 (Viewpoint) の集合によって表現されることになる。視点は、様々な視点 (Leaf viewpoint) の組合せにより構成される。Leaf viewpoint に相当するものとしては、上記②開発知識、③開発プロジェクト、④知識の形態の観点に相当する観点と、⑤ZF による抽象軸と専門軸による観点等が考えられる。これらの Leaf viewpoint のクラス群は一例であり、図 4-2 の例をベースに、各組織が組織の資産を構成する際のメタモデルを定義し、同メタモデルに沿った資産の運用管理を行うことが重要である。また、そのようなメタモデルそのものが組織の重要なコア資産となると考えられる。

表 4-1 ザックマンフレームワーク

専門軸 役割軸	Data (What)	Function (How)	Network (Where)	People (Who)	Time (When)	Motivation (Why)
Scope (Contextual) Planner	List of things important to business	List of processes the business performs	List of locations in which the business operates	List of organizations important to the business	List of events/cycles significant to the business	List of business goals/strategies
Business Model (Conceptual) Owner	Semantic model	Business process model	Business logistics system	Workflow model	Master schedule	Business plan
System Model (Logical) Designer	Logical data model	Application architecture	Distributed systems architecture	Human interface architecture	Processing structure	Business rule model
Technology Model (Physical) Builder	Physical data model	System design	Technology architecture	Presentation architecture	Control structure	Rule design
Detailed Representations (Out-of-context) Subcontractor	Data definition	Software programs	Network architecture	Security architecture	Timing definition	Rule specification

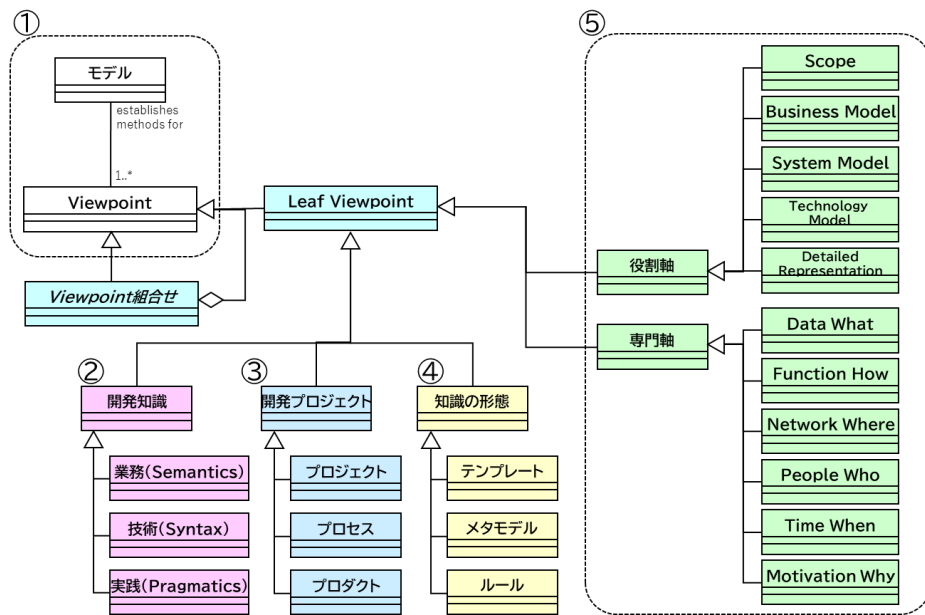


図 4-2 コアアセットの整理、分類、蓄積ためのメタモデル

4. 3 メタモデルの拡張

エンタープライズ系のシステム開発では、現行業務分析にもとづき、システム化のゴールを設定し、新システムの機能とそれら機能を横断する非機能の要求を定義する。メタモデルにそのような視点を具体化するために、Zachman Framework[Zackman]の考え方の枠組みと、エリクソン・ペンカーのゴール、ビジネスプロセス、リソース、ルール等からなるビジネス環境の基本メタモデルを取り入れ[Ericsson2000]。そして、実事例の分析と関係者インタビューを通してメタモデルを詳細化することが提案されている[Kitagawa2010]。ここで定義されたメタモデルの例を図 4-3 に示す。

図 4-3 に示す要求仕様のメタモデルは要求を網羅的に定義するための観点に相当する。要求仕様のメタモデルは、プロジェクトの「ゴール」、「問題」、「解決策」と、業務分析要素となる「業務」、「関係者」、「情報」、「拠点」、「イベント」のクラスで構成される。ここでクラスは、定義すべき観点到に相当します。「ゴール」、「問題」、「解決策」の関連は、「ゴール」の達成を妨げるものが「問題」であり、「問題」を解決するものが「解決策」です。各クラスは発注者と合意済／未済など状態変化があるため、状態変化を扱えるように各クラスの属性を定義されている。図 4-3 では表現の単純化のため「構成要素」クラスにこれらの日付を定義し、各クラスは「構成要素」を継承するように記述されている。なお各日付は以下の意味を持たせてある。

【発行日付】 要求を抽出した日付。

【確定日付】 要求を発注者と合意した日付。

【削除日付】 要求をシステム化の対象外とした日付。

このメタモデルを組織の標準として共有すれば、業務分析からシステム導入のための要求定義までの一連のプロセスにおいて、組織が明らかにすべきアクションアイテムが明確になるため、初級の技術者や業務分析に取り組むことに不慣れなステークホルダ間でも、実施すべき事柄の漏れの防止が期待できる。

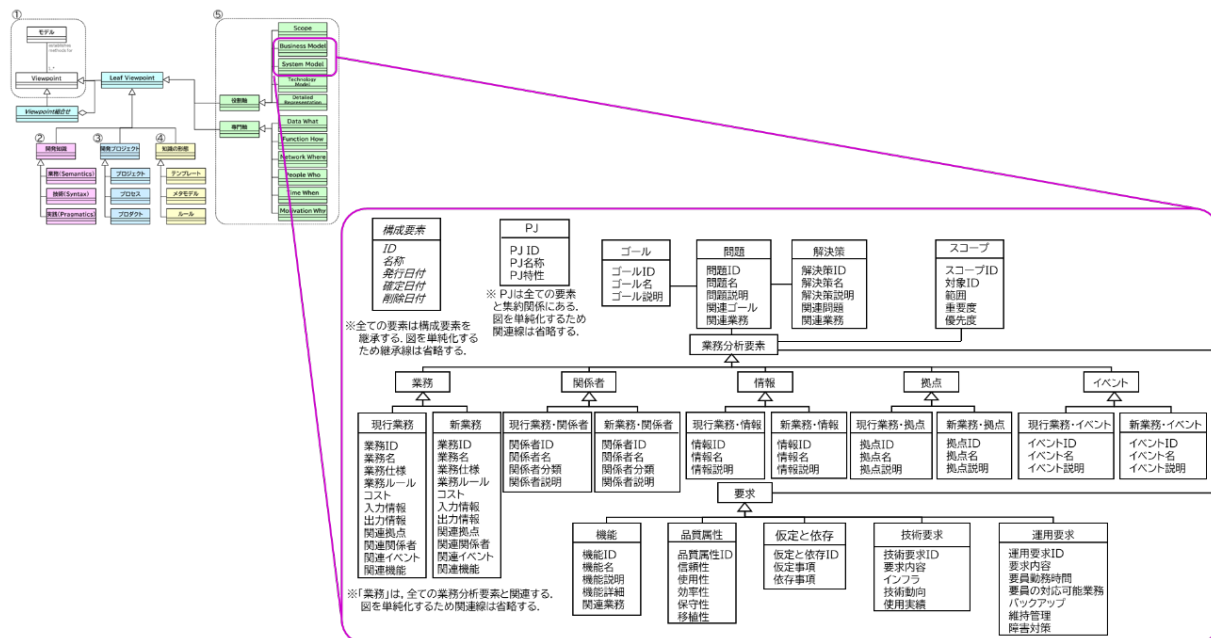


図 4-3 メタモデルの拡張例

5. 人材育成で重要な問題発見力とモデリング力

As is と To be をモデル化し、それらを比較し、これらの差異として、現状の問題と特定することで、問題発見がしやすくなることを述べた。問題発見力を高めるには、対象をモデル化するモデリング力が重要になる。モデリングには、データモデリング、オブジェクト指向モデリング、ビジネスモデリングなど様々ある。モデル化の対象には、既知のモデルの表現形態やモデリングのコツが明らかなものもあれば、どのような表現形態や手続きでモデル化すればよいのか未知のものも含まれる。

モデリング力を高めるには、前述した上位の視点でのメタモデル化の体験が重要であると考えられる。上位の視点で対象を捉え一般化・抽象化することができれば、新たなモデリングの方法の考案が可能であると期待できる。

複雑に変動する対象を何等かの視点で一般化・抽象化して表現し、As is と To be をモデル化し、それらの間の変化をとらえようとするには、単純にモデル化の計算ロジック適用では不十分で、そこには、人間が積極的に何かを発見しようとするこ、あるいは、何らかに気づく感性の強化が重要と考えられる。加えて、人間の試行錯誤や感性を強化する体験を経て獲得した発見的知識を備えていることが、発見

力を高めるための、人財育成のキーポイントになると思われる。

メタモデルは視点の集合です。様々な事象を観察し整理するための重要な視点に気づかなければ、メタモデルとして形式知化することは困難です。メタモデリング力の強化には、様々な気づきを促す感性の強化、他者の立場に立ち共感する体験、モデリングならびに俯瞰したメタモデル化の試行錯誤が肝要であり、得られた発見的知識を、自分の組織に展開し、実装しやり抜く情熱のようなマインドセットが不可欠であると考えられる。

以上をまとめ、KCC の観点で再利用率/問題発見力を強化するための新たな学びの考え方を図 5-1 に示す。現状、トップエスイーの要求工学コースで実践している、Science、Engineering、Design、Art の視点からの学びに加えて、再利用率を強化するモデリング演習の講座は、人財育成に有効であると期待できる。今後のそれぞれの強化のポイントを以下に整理する。

- Science の視点：良いモデルの判断を数学的理論に基づいて実施できるように、改めて、データサイエンスや数学を学び直すこと。とくに、未知の対象に対して新たなモデル化へと写像するには、体系化された数学的体系を理解し適用すること。
- Engineering の視点：従来同様にオブジェクト指向に基づき、構造、振る舞い、ユースケースのモデリング力に加えて、変動性を織り込んだドメインモデリングを通して、A s is と To be とその差異を抽出する試行錯誤を体験すること。
- Design の視点：多様化するユーザの視点に立ち、ユーザ中心のサービスやUI のデザイン力をワークショップ形式での体験を通して高めること。
- Art の視点：未知の対象に対して、新たな問題を発見するための気づきを促す感性を強化すること。気づきの言語化が困難である場合や、対象を言語化することでその本質が失われぬように、言語以外による多様な表現能力を身に着けること。
- メタな視点：S E D A の観点での学びや体験の反復によって各々が考える新たなアイデアをメタな視点で捉えて、収斂と発散を繰り返すこと、あいまいな状況を許容しながらやりぬく情熱を含むマインドセットを涵養し、またその反復のサイクル加速を促すこと。

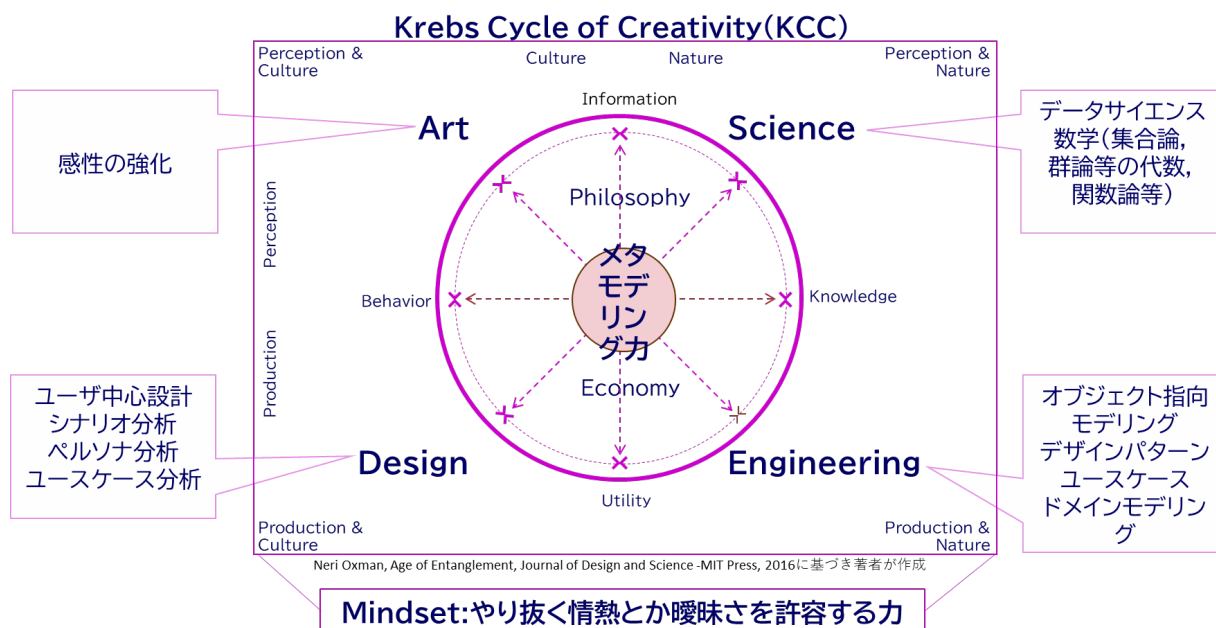


図 5-1 KCC の観点で再利用力/問題発見力を強化するための提案

6. おわりに

イノベーションを起こすための「問題発見力」を高めるには、現実と理想をモデル化し、その差異を特定すること、さらには差異を解決する方法を新サービスとして考案することが有効であることを提案した。対象をどのようなモデルで表現するのかが重要であり、モデリング力の強化には、メタモデリング能力が重要であることを述べた。メタモデリング力を高めるには、様々な気づきを促す感性の強化、他者の立場に立ち共感する体験、モデリングならびに俯瞰したメタモデル化の試行錯誤、得られた発見的知識を、自身の組織に展開して、実装しやり抜く情熱のようなマインドセットが必要であると考えられる。

本稿で述べた、様々な人財育成の強化のポイントを、個人や個別の組織で実践するのは簡単ではない。個人が様々なコミュニティに参加して自己研鑽することや、トップエスイーのようなリカレント教育の場の活用も有効である。著者も、人財育成の場を提供する側として、社会環境、市場、技術の変化に対応しながら、SEDAをメタな視点で捉え、あいまいさを許容し情熱を持って取り組んでいく。

参考文献

[Young2003] Young, James, “*A Technique for Producing Ideas*”, Advertising Age Classics Library, McGraw Hill LLC, 2003.

- [JISA2023] 情報サービス産業協会, 『**Digital Transformation のための要求獲得実践ガイド**』, 近代科学社 Digital, 2023.
- [Inoki2008] 位野木万里, 深澤良彰, 『**プロダクトラインのスコープ定義におけるカバー率と対応度によるコアアセット可視化手法**』, 情報処理学会論文誌, 49/ 2, pp.968-987, 2008.
- [Inoki2007] 位野木万里, 深澤良彰, 『**プロダクトラインの開発と進化：パターン指向コアアセット改善手法**』, 情報処理学会論文誌, 48/ 8, pp.2508-2521, 2007.
- [Oxman2016] Neri Oxman, “*Age of Entanglement, Journal of Design and Science*”, MIT Press, 2016.
- [Takahashi2022] 高橋真吾, 後藤裕介, 大堀耕太郎, 『**社会システムモデリング**』, 共立出版, 2022.
- [Apostel1961] Leo, Apostel, “*Towards the Formal Study of Models in the Non-Formal Sciences. In: The Concept and the Role of the Model in Mathematics and Natural and Social Sciences*”, Synthese Library, vol 3. Springer, 1961.
- [Moriya1984] 守屋美賀雄, 『**代数学教程**』, 帝国書院, 1984.
- [Gonzalez-Perez2008] Cesar Gonzalez-Perez, Brian Henderson-Sellers: “*Metamodelling for Software Engineering*”, 1 edition, Wiley, 2008.
- [Zachman] The Zachman Institute for Framework Advancement, www.zifa.com
- [IEEE2000] IEEE Std 1471-2000: “*IEEE Recommended Practice for Architectural Description of Software-Intensive Systems –Description*”, 2000.
- [ISO/IEC/IEEE2022] ISO/IEC/IEEE 42010:2022, “*Software, systems and enterprise — Architecture description*”, 2022.
- [Eriksson2000] Hans-Erik Eriksson, Magnus Penker, “*Business Modeling with UML: Business Patterns at Work (OMG)*”, John Wiley & Sons, 2000, [翻訳] 『**UML によるビジネスモデリング, ソフトバンククリエイティブ**』, 2002.
- [Kitagawa2010] 北川貴之, 橋本憲幸, 吉田和樹, 位野木万里, 『**要求定義における暗黙知の形式知化手法**』, コンピュータソフトウェア, Vol.27, No.3, pp.93-98, 2010.

